# Table of contents

1 Foreword .......................................................................................................................... 13  
   1.1 Notes on the documentation ....................................................................................... 13  
   1.2 Safety instructions ...................................................................................................... 14  
2 TwinCAT ADS .NET API Documentation ....................................................................... 15  
   2.1 Prerequisites ............................................................................................................... 16  
   2.2 Installation ................................................................................................................ 16  
3 Version History ................................................................................................................ 18  
   3.1 Version 5.0.X ............................................................................................................ 18  
4 Concepts ........................................................................................................................... 19  
   4.1 Access Data via IndexGroup/IndexOffset ..................................................................... 19  
   4.2 Async programming (async, await) ............................................................................ 20  
   4.3 Use of ADS Notifications ............................................................................................ 22  
   4.4 Access Data via Symbolic path .................................................................................. 25  
   4.5 Access Data via Symbol handles .............................................................................. 26  
   4.6 Value marshalling with ANYTYPE concept ................................................................. 27  
   4.7 Access Data via Symbol Loader ................................................................................ 31  
   4.8 Automatic dynamic marshalling of values .................................................................. 32  
   4.9 Reactive Extensions and ADS .................................................................................... 35  
5 HowTo Samples .................................................................................................................. 38  
   5.1 Read/Write primitive values ....................................................................................... 38  
   5.2 Read/Write string types ............................................................................................. 42  
   5.3 Read/Write PlcOpen types (DATE, TIME ...) .............................................................. 45  
   5.4 Event driven read with ADS Notifications .................................................................. 49  
   5.5 Reactive Read/Write with Reactive Extensions .......................................................... 50  
   5.6 Upgrading existing ADS Application code (Version 4.X --> 5.X) .................................. 52  
6 TwinCAT.Ads Namespaces ............................................................................................... 55  
   6.1 TwinCAT Namespace .................................................................................................. 55  
      6.1.1 AdsException Class ............................................................................................... 57  
      6.1.2 ClientNotConnectedException Class ................................................................. 63  
      6.1.3 ConnectionState Enumeration ............................................................................ 67  
      6.1.4 ConnectionStateChangedEventArgs Class .......................................................... 68  
      6.1.5 ConnectionStateChangedEventArgsReason Enumeration .............................. 73  
      6.1.6 IConnection Interface ........................................................................................ 74  
      6.1.7 IConnectionStateProvider Interface .................................................................. 80  
      6.1.8 IConnectionStateChangedEventArgs Class ....................................................... 84  
      6.1.9 IConnectionStateChangedEventArgsReason Enumeration ........................... 84  
      6.1.10 IConnectionFactory Interface ............................................................................ 89  
      6.1.11 IConnectionFactory.TSession, TAddress, TSettings, Interface ....................... 97  
      6.1.12 ISymbolLoaderSettings Interface ..................................................................... 99  
      6.1.13 ISymbolServerProvider Interface ..................................................................... 99  
      6.1.14 Session Class .................................................................................................... 101  
      6.1.15 SessionConnectionStateChangedEventArgs Class ......................................... 116  
      6.1.16 SessionException Class ..................................................................................... 122
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1.17</td>
<td>SessionNotConnectedException Class</td>
<td>129</td>
</tr>
<tr>
<td>6.1.18</td>
<td>SessionProvider.TSession, TAddress, TSettings. Class</td>
<td>134</td>
</tr>
<tr>
<td>6.1.19</td>
<td>SessionProviderCapabilities Enumeration</td>
<td>140</td>
</tr>
<tr>
<td>6.1.20</td>
<td>SymbolLoaderSettings Class</td>
<td>140</td>
</tr>
<tr>
<td>6.1.21</td>
<td>SymbolsLoadMode Enumeration</td>
<td>149</td>
</tr>
<tr>
<td>6.1.22</td>
<td>ValueUpdateMode Enumeration</td>
<td>150</td>
</tr>
<tr>
<td>6.2</td>
<td>TwinCAT.Ads Namespace</td>
<td>151</td>
</tr>
<tr>
<td>6.2.1</td>
<td>AdsClient Class</td>
<td>154</td>
</tr>
<tr>
<td>6.2.2</td>
<td>AdsClientSettings Class</td>
<td>344</td>
</tr>
<tr>
<td>6.2.3</td>
<td>AdsCommandId Enumeration</td>
<td>349</td>
</tr>
<tr>
<td>6.2.4</td>
<td>AdsCommunicationStatistics Class</td>
<td>350</td>
</tr>
<tr>
<td>6.2.5</td>
<td>AdsConnection Class</td>
<td>357</td>
</tr>
<tr>
<td>6.2.6</td>
<td>AdsDataTypeArrayInfo Class</td>
<td>572</td>
</tr>
<tr>
<td>6.2.7</td>
<td>AdsDataTypeId Enumeration</td>
<td>574</td>
</tr>
<tr>
<td>6.2.8</td>
<td>AdsErrorCode Enumeration</td>
<td>575</td>
</tr>
<tr>
<td>6.2.9</td>
<td>AdsErrorCodeExtensions Class</td>
<td>581</td>
</tr>
<tr>
<td>6.2.10</td>
<td>AdsErrorException Class</td>
<td>583</td>
</tr>
<tr>
<td>6.2.11</td>
<td>AdsInvalidNotificationException Class</td>
<td>591</td>
</tr>
<tr>
<td>6.2.12</td>
<td>AdsNotificationErrorEventArgs Class</td>
<td>595</td>
</tr>
<tr>
<td>6.2.13</td>
<td>AdsNotificationEventArgs Class</td>
<td>597</td>
</tr>
<tr>
<td>6.2.14</td>
<td>AdsNotificationExEventArgs Class</td>
<td>601</td>
</tr>
<tr>
<td>6.2.15</td>
<td>AdsSession Class</td>
<td>603</td>
</tr>
<tr>
<td>6.2.16</td>
<td>AdsSessionBase Class</td>
<td>614</td>
</tr>
<tr>
<td>6.2.17</td>
<td>AdsState Enumeration</td>
<td>626</td>
</tr>
<tr>
<td>6.2.18</td>
<td>AdsStateChangedEventArgs Class</td>
<td>627</td>
</tr>
<tr>
<td>6.2.19</td>
<td>AdsStateChangedEventArgs2 Class</td>
<td>629</td>
</tr>
<tr>
<td>6.2.20</td>
<td>AdsSumCommandException Class</td>
<td>632</td>
</tr>
<tr>
<td>6.2.21</td>
<td>AdsSymbolVersionChangedEventArgs Class</td>
<td>637</td>
</tr>
<tr>
<td>6.2.22</td>
<td>AdsTransMode Enumeration</td>
<td>639</td>
</tr>
<tr>
<td>6.2.23</td>
<td>AdsVersion Class</td>
<td>642</td>
</tr>
<tr>
<td>6.2.24</td>
<td>AmsAddress Class</td>
<td>648</td>
</tr>
<tr>
<td>6.2.25</td>
<td>AmsNetId Class</td>
<td>665</td>
</tr>
<tr>
<td>6.2.26</td>
<td>AmsPort Enumeration</td>
<td>693</td>
</tr>
<tr>
<td>6.2.27</td>
<td>AmsRouterNotificationEventArgs Class</td>
<td>695</td>
</tr>
<tr>
<td>6.2.28</td>
<td>AmsRouterState Enumeration</td>
<td>697</td>
</tr>
<tr>
<td>6.2.29</td>
<td>DeviceInfo Class</td>
<td>698</td>
</tr>
<tr>
<td>6.2.30</td>
<td>IAdsAnyAccess Interface</td>
<td>702</td>
</tr>
<tr>
<td>6.2.31</td>
<td>IAdsConnectAddress Interface</td>
<td>733</td>
</tr>
<tr>
<td>6.2.32</td>
<td>IAdsConnection Interface</td>
<td>765</td>
</tr>
<tr>
<td>6.2.33</td>
<td>IAdsDisposableConnection Interface</td>
<td>797</td>
</tr>
<tr>
<td>6.2.34</td>
<td>IAdsHandle Interface</td>
<td>827</td>
</tr>
<tr>
<td>6.2.35</td>
<td>IAdsNotifications Interface</td>
<td>839</td>
</tr>
<tr>
<td>6.2.36</td>
<td>IAdsReadWrite Interface</td>
<td>870</td>
</tr>
<tr>
<td>6.2.37</td>
<td>IAdsReadWrite2 Interface</td>
<td>875</td>
</tr>
<tr>
<td>6.2.38</td>
<td>IAdsReadWriteTimeoutAccess Interface</td>
<td>879</td>
</tr>
<tr>
<td>6.2.39</td>
<td>IAdsRpcInvoke Interface</td>
<td>886</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads.Server Namespace

<table>
<thead>
<tr>
<th>Version</th>
<th>Class</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4.8</td>
<td>ServerConnectionStateChangedEventArgs Class</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.7</td>
<td>ServerConnectionState Enumeration</td>
<td>1056</td>
</tr>
<tr>
<td>6.4.6</td>
<td>Notification Class</td>
<td>972</td>
</tr>
<tr>
<td>6.4.5</td>
<td>NotificationSettings Class</td>
<td>973</td>
</tr>
<tr>
<td>6.4.4</td>
<td>ResultAds Class</td>
<td>979</td>
</tr>
<tr>
<td>6.4.3</td>
<td>ResultAnyValue Class</td>
<td>989</td>
</tr>
<tr>
<td>6.4.2</td>
<td>ResultDeviceInfo Class</td>
<td>1001</td>
</tr>
<tr>
<td>6.4.1</td>
<td>ResultHandle Class</td>
<td>1005</td>
</tr>
<tr>
<td></td>
<td>ResultRead Class</td>
<td>1008</td>
</tr>
<tr>
<td>6.4.0</td>
<td>ResultReadAdsStateClass</td>
<td>1010</td>
</tr>
<tr>
<td>6.4.9</td>
<td>ResultReadDeviceStateClass</td>
<td>1013</td>
</tr>
<tr>
<td>6.4.8</td>
<td>ResultReadWriteClass</td>
<td>1016</td>
</tr>
<tr>
<td>6.4.7</td>
<td>ResultReadWriteBytes Class</td>
<td>1019</td>
</tr>
<tr>
<td>6.4.6</td>
<td>ResultWriteClass</td>
<td>1022</td>
</tr>
<tr>
<td>6.4.5</td>
<td>ResultValue.TValue.Class</td>
<td>1025</td>
</tr>
<tr>
<td>6.4.4</td>
<td>ResultValue.TValue.TClass</td>
<td>1029</td>
</tr>
<tr>
<td>6.4.3</td>
<td>ResultWriteClass</td>
<td>1032</td>
</tr>
<tr>
<td>6.4.2</td>
<td>SessionSettings Class</td>
<td>1035</td>
</tr>
<tr>
<td>6.4.1</td>
<td>StateInfo Structure</td>
<td>1041</td>
</tr>
<tr>
<td>6.4.8</td>
<td>TaskExtensions Class</td>
<td>1049</td>
</tr>
<tr>
<td>6.4.7</td>
<td>TransportProtocols Enumeration</td>
<td>1053</td>
</tr>
<tr>
<td></td>
<td>ValueNotificationEventArgs.T класс</td>
<td>1053</td>
</tr>
</tbody>
</table>

### TwinCAT.Ads.Reactive Namespace

<table>
<thead>
<tr>
<th>Version</th>
<th>Class</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AdsClientExtensions Class</td>
<td>1056</td>
</tr>
<tr>
<td></td>
<td>AnyTypeExtensions Class</td>
<td>1075</td>
</tr>
<tr>
<td></td>
<td>SymbolValueNotification Class</td>
<td>1104</td>
</tr>
<tr>
<td></td>
<td>ValueSymbolExtensions Class</td>
<td>1106</td>
</tr>
</tbody>
</table>

### Table of contents

<table>
<thead>
<tr>
<th>Version</th>
<th>Class</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4.8</td>
<td>ServerConnectionStateChangedEventArgs Class</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.7</td>
<td>ServerConnectionState Enumeration</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.6</td>
<td>Notification Class</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.5</td>
<td>NotificationSettings Class</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.4</td>
<td>ResultAds Class</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.3</td>
<td>ResultAnyValue Class</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.2</td>
<td>ResultDeviceInfo Class</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.1</td>
<td>ResultHandle Class</td>
<td>1121</td>
</tr>
<tr>
<td></td>
<td>ResultRead Class</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.9</td>
<td>ResultReadAdsStateClass</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.8</td>
<td>ResultReadDeviceStateClass</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.7</td>
<td>ResultReadWriteClass</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.6</td>
<td>ResultWriteClass</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.5</td>
<td>ResultValue.TValue.Class</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.4</td>
<td>ResultValue.TValue.TClass</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.3</td>
<td>ResultWriteClass</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.2</td>
<td>SessionSettings Class</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.1</td>
<td>StateInfo Structure</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.8</td>
<td>TaskExtensions Class</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.7</td>
<td>TransportProtocols Enumeration</td>
<td>1121</td>
</tr>
<tr>
<td>6.4.6</td>
<td>ValueNotificationEventArgs.T класс</td>
<td>1121</td>
</tr>
</tbody>
</table>

TC1000 Version: 1.1
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4.9</td>
<td>ServerNotConnectedException Class</td>
<td>1202</td>
</tr>
<tr>
<td>6.5</td>
<td>TwinCAT.Ads.SumCommand Namespace</td>
<td>1205</td>
</tr>
<tr>
<td>6.5.1</td>
<td>ISumCommand Interface</td>
<td>1206</td>
</tr>
<tr>
<td>6.5.2</td>
<td>ResultSumCommand Class</td>
<td>1210</td>
</tr>
<tr>
<td>6.5.3</td>
<td>ResultSumHandles Class</td>
<td>1212</td>
</tr>
<tr>
<td>6.5.4</td>
<td>ResultSumHandles2 Class</td>
<td>1214</td>
</tr>
<tr>
<td>6.5.5</td>
<td>ResultSumReadRaw Class</td>
<td>1218</td>
</tr>
<tr>
<td>6.5.6</td>
<td>ResultSumValues Class</td>
<td>1220</td>
</tr>
<tr>
<td>6.5.7</td>
<td>SumCreateHandles Class</td>
<td>1222</td>
</tr>
<tr>
<td>6.5.8</td>
<td>SumHandleRead Class</td>
<td>1228</td>
</tr>
<tr>
<td>6.5.9</td>
<td>SumHandleWrite Class</td>
<td>1232</td>
</tr>
<tr>
<td>6.5.10</td>
<td>SumReleaseHandles Class</td>
<td>1237</td>
</tr>
<tr>
<td>6.5.11</td>
<td>SumSymbolRead Class</td>
<td>1242</td>
</tr>
<tr>
<td>6.5.12</td>
<td>SumSymbolWrite Class</td>
<td>1248</td>
</tr>
<tr>
<td>6.6</td>
<td>TwinCAT.Ads.TcpRouter Namespace</td>
<td>1254</td>
</tr>
<tr>
<td>6.6.1</td>
<td>AmsTcpIpRouter Class</td>
<td>1254</td>
</tr>
<tr>
<td>6.6.2</td>
<td>IAmsRouter Interface</td>
<td>1273</td>
</tr>
<tr>
<td>6.6.3</td>
<td>Route Class</td>
<td>1281</td>
</tr>
<tr>
<td>6.6.4</td>
<td>RouteCollection Class</td>
<td>1292</td>
</tr>
<tr>
<td>6.6.5</td>
<td>RouterException Class</td>
<td>1306</td>
</tr>
<tr>
<td>6.6.6</td>
<td>RouterNotInitializedException Class</td>
<td>1310</td>
</tr>
<tr>
<td>6.6.7</td>
<td>RouterNotStartedException Class</td>
<td>1313</td>
</tr>
<tr>
<td>6.6.8</td>
<td>RouterStatus Enumeration</td>
<td>1316</td>
</tr>
<tr>
<td>6.6.9</td>
<td>RouterStatusChangedEventArgs Class</td>
<td>1317</td>
</tr>
<tr>
<td>6.6.10</td>
<td>StaticRoutesXmlConfigurationBuilderExtension Class</td>
<td>1320</td>
</tr>
<tr>
<td>6.6.11</td>
<td>StaticRoutesXmlConfigurationProvider Class</td>
<td>1321</td>
</tr>
<tr>
<td>6.6.12</td>
<td>StaticRoutesXmlConfigurationSource Class</td>
<td>1326</td>
</tr>
<tr>
<td>6.7</td>
<td>TwinCAT.Ads.TypeSystem Namespace</td>
<td>1328</td>
</tr>
<tr>
<td>6.7.1</td>
<td>AliasType Class</td>
<td>1329</td>
</tr>
<tr>
<td>6.7.2</td>
<td>ArrayType Class</td>
<td>1335</td>
</tr>
<tr>
<td>6.7.3</td>
<td>BitMappingType Class</td>
<td>1345</td>
</tr>
<tr>
<td>6.7.4</td>
<td>DataType Class</td>
<td>1349</td>
</tr>
<tr>
<td>6.7.5</td>
<td>EnumType.T. Class</td>
<td>1363</td>
</tr>
<tr>
<td>6.7.6</td>
<td>Field Class</td>
<td>1374</td>
</tr>
<tr>
<td>6.7.7</td>
<td>IAdsSymbol Interface</td>
<td>1379</td>
</tr>
<tr>
<td>6.7.8</td>
<td>IAdsSymbolLoader Interface</td>
<td>1383</td>
</tr>
<tr>
<td>6.7.9</td>
<td>IContextMaskProvider Interface</td>
<td>1387</td>
</tr>
<tr>
<td>6.7.10</td>
<td>Instance Class</td>
<td>1388</td>
</tr>
<tr>
<td>6.7.11</td>
<td>Member Class</td>
<td>1406</td>
</tr>
<tr>
<td>6.7.12</td>
<td>PCCHType Class</td>
<td>1411</td>
</tr>
<tr>
<td>6.7.13</td>
<td>PointerType Class</td>
<td>1414</td>
</tr>
<tr>
<td>6.7.14</td>
<td>PrimitiveType Class</td>
<td>1418</td>
</tr>
<tr>
<td>6.7.15</td>
<td>PVoidType Class</td>
<td>1422</td>
</tr>
<tr>
<td>6.7.16</td>
<td>ReferenceType Class</td>
<td>1425</td>
</tr>
<tr>
<td>6.7.17</td>
<td>RpcMethod Class</td>
<td>1433</td>
</tr>
<tr>
<td>6.7.18</td>
<td>RpcMethodParameter Class</td>
<td>1440</td>
</tr>
</tbody>
</table>
6.11 TwinCAT.TypeSystem Namespace .......................................................................................................................... 1622
6.11.1 AnySymbolSpecifier Class ........................................................................................................................................ 1629
6.11.2 AnyTypeSpecifier Class ........................................................................................................................................... 1633
6.11.3 CannotAccessVirtualSymbolException Class ......................................................................................................... 1641
6.11.4 CannotResolveDataTypeException Class .................................................................................................................... 1644
6.11.5 DataTypeCategory Enumeration ..................................................................................................................................... 1649
6.11.6 DataTypeCollection Class ............................................................................................................................................ 1650
6.11.7 DataTypeEventArgs Class .............................................................................................................................................. 1656
6.11.8 DateTimeException Class .............................................................................................................................................. 1658
6.11.9 DateTimeNameEventArgs Class ...................................................................................................................................... 1665
6.11.10 Dimension Class .......................................................................................................................................................... 1668
6.11.11 DimensionCollection Class ........................................................................................................................................ 1671
6.11.12 DynamicAliasInstance Class ........................................................................................................................................ 1687
6.11.13 DynamicArrayInstance Class ....................................................................................................................................... 1698
6.11.14 DynamicOversamplingArrayInstance Class .................................................................................................................. 1712
6.11.15 DynamicPointerInstance Class .................................................................................................................................... 1720
6.11.16 DynamicPointerValue Class ......................................................................................................................................... 1728
6.11.17 DynamicReferenceInstance Class .................................................................................................................................. 1734
6.11.18 DynamicReferenceValue Class ................................................................................................................................... 1747
6.11.19 DynamicRpcStructInstance Class .................................................................................................................................. 1752
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.11.20</td>
<td>DynamicStructInstance Class</td>
<td>1779</td>
</tr>
<tr>
<td>6.11.21</td>
<td>DynamicSymbol Class</td>
<td>1791</td>
</tr>
<tr>
<td>6.11.22</td>
<td>DynamicSymbolsCollection Class</td>
<td>1853</td>
</tr>
<tr>
<td>6.11.23</td>
<td>DynamicUnionInstance Class</td>
<td>1860</td>
</tr>
<tr>
<td>6.11.24</td>
<td>DynamicValue Class</td>
<td>1869</td>
</tr>
<tr>
<td>6.11.25</td>
<td>DynamicVirtualStructInstance Class</td>
<td>1894</td>
</tr>
<tr>
<td>6.11.26</td>
<td>EnumValue.T. Class</td>
<td>1901</td>
</tr>
<tr>
<td>6.11.27</td>
<td>EnumValueCollection Class</td>
<td>1908</td>
</tr>
<tr>
<td>6.11.28</td>
<td>EnumValueCollection.T. Class</td>
<td>1926</td>
</tr>
<tr>
<td>6.11.29</td>
<td>FieldCollection Class</td>
<td>1945</td>
</tr>
<tr>
<td>6.11.30</td>
<td>IAliasInstance Interface</td>
<td>1952</td>
</tr>
<tr>
<td>6.11.31</td>
<td>IAliasType Interface</td>
<td>1954</td>
</tr>
<tr>
<td>6.11.32</td>
<td>IAnyTypeMarshaler Interface</td>
<td>1957</td>
</tr>
<tr>
<td>6.11.33</td>
<td>IArrayInstance Interface</td>
<td>1964</td>
</tr>
<tr>
<td>6.11.34</td>
<td>IArrayType Interface</td>
<td>1971</td>
</tr>
<tr>
<td>6.11.35</td>
<td>IArrayValue Interface</td>
<td>1976</td>
</tr>
<tr>
<td>6.11.36</td>
<td>IAttributedInstance Interface</td>
<td>1980</td>
</tr>
<tr>
<td>6.11.37</td>
<td>IBitSize Interface</td>
<td>1982</td>
</tr>
<tr>
<td>6.11.38</td>
<td>IDatatype Interface</td>
<td>1986</td>
</tr>
<tr>
<td>6.11.39</td>
<td>IDatatypeCollection Interface</td>
<td>1993</td>
</tr>
<tr>
<td>6.11.40</td>
<td>IDatatypeCollection.T. Interface</td>
<td>1995</td>
</tr>
<tr>
<td>6.11.41</td>
<td>IDimension Interface</td>
<td>1998</td>
</tr>
<tr>
<td>6.11.42</td>
<td>IDimensionCollection Interface</td>
<td>2000</td>
</tr>
<tr>
<td>6.11.43</td>
<td>IDynamicSymbol Interface</td>
<td>2004</td>
</tr>
<tr>
<td>6.11.44</td>
<td>IDynamicSymbolLoader Interface</td>
<td>2007</td>
</tr>
<tr>
<td>6.11.45</td>
<td>IDynamicSymbolsCollection Interface</td>
<td>2010</td>
</tr>
<tr>
<td>6.11.46</td>
<td>IDynamicValue Interface</td>
<td>2011</td>
</tr>
<tr>
<td>6.11.47</td>
<td>IEnumType Interface</td>
<td>2014</td>
</tr>
<tr>
<td>6.11.48</td>
<td>IEnumType.T. Interface</td>
<td>2021</td>
</tr>
<tr>
<td>6.11.49</td>
<td>IEnumValue Interface</td>
<td>2028</td>
</tr>
<tr>
<td>6.11.50</td>
<td>IEnumValueCollection Interface</td>
<td>2031</td>
</tr>
<tr>
<td>6.11.51</td>
<td>IEnumValueCollection.TEnumValue, TValue. Interface</td>
<td>2033</td>
</tr>
<tr>
<td>6.11.52</td>
<td>IField Interface</td>
<td>2040</td>
</tr>
<tr>
<td>6.11.53</td>
<td>IFieldCollection Interface</td>
<td>2042</td>
</tr>
<tr>
<td>6.11.54</td>
<td>IGenericTypeMarshaler Interface</td>
<td>2046</td>
</tr>
<tr>
<td>6.11.55</td>
<td>IHierarchicalSymbol Interface</td>
<td>2048</td>
</tr>
<tr>
<td>6.11.56</td>
<td>Instance Interface</td>
<td>2052</td>
</tr>
<tr>
<td>6.11.57</td>
<td>InstanceCollection.T. Interface</td>
<td>2057</td>
</tr>
<tr>
<td>6.11.58</td>
<td>IMember Interface</td>
<td>2065</td>
</tr>
<tr>
<td>6.11.59</td>
<td>IMemberCollection Interface</td>
<td>2068</td>
</tr>
<tr>
<td>6.11.60</td>
<td>INamespaceCollection Interface</td>
<td>2072</td>
</tr>
<tr>
<td>6.11.61</td>
<td>INamespaceCollection.T. Interface</td>
<td>2073</td>
</tr>
<tr>
<td>6.11.62</td>
<td>InstanceCollectionMode Enumeration</td>
<td>2075</td>
</tr>
<tr>
<td>6.11.63</td>
<td>InsufficientAccessRightsException Class</td>
<td>2076</td>
</tr>
<tr>
<td>6.11.64</td>
<td>IOversamplingArrayInstance Interface</td>
<td>2079</td>
</tr>
<tr>
<td>6.11.65</td>
<td>IPointerInstance Interface</td>
<td>2083</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>6.11.66</td>
<td>IPointerType Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.67</td>
<td>IPrimitiveType Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.68</td>
<td>IProcessImageAddress Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.69</td>
<td>IReferenceInstance Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.70</td>
<td>IReferenceType Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.71</td>
<td>IRpcCallableInstance Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.72</td>
<td>IRpcCallableType Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.73</td>
<td>IRpcMethod Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.74</td>
<td>IRpcMethodCollection Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.75</td>
<td>IRpcMethodParameter Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.76</td>
<td>IRpcMethodParameterCollection Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.77</td>
<td>IRpcStructInstance Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.78</td>
<td>IStringInstance Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.79</td>
<td>IStringMarshaler Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.80</td>
<td>IStringType Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.81</td>
<td>IStructInstance Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.82</td>
<td>IStructType Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.83</td>
<td>IStructValue Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.84</td>
<td>ISubRangeType Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.85</td>
<td>ISubRangeType.T. Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.86</td>
<td>ISymbol Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.87</td>
<td>ISymbolCollection Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.88</td>
<td>ISymbolCollection.T. Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.89</td>
<td>ISymbolFactory Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.90</td>
<td>ISymbolFactoryServicesProvider Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.91</td>
<td>ISymbolInfo Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.92</td>
<td>ISymbolLoader Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.93</td>
<td>ISymbolProvider Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.94</td>
<td>ISymbolServer Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.95</td>
<td>ITypeAttribute Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.96</td>
<td>ITypeAttributeCollection Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.97</td>
<td>ITypeMarshaler Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.98</td>
<td>IUnionInstance Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.99</td>
<td>IUnionType Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.100</td>
<td>IValue Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.101</td>
<td>IValueAccessorProvider Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.102</td>
<td>IValueAnySymbol Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.103</td>
<td>IValueRawSymbol Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.104</td>
<td>IValueSymbol Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.105</td>
<td>IVirtualStructInstance Interface</td>
<td></td>
</tr>
<tr>
<td>6.11.106</td>
<td>MarshalException Class</td>
<td></td>
</tr>
<tr>
<td>6.11.107</td>
<td>MemberCollection Class</td>
<td></td>
</tr>
<tr>
<td>6.11.108</td>
<td>MethodParamFlags Enumeration</td>
<td></td>
</tr>
<tr>
<td>6.11.109</td>
<td>PrimitiveTypeFlags Enumeration</td>
<td></td>
</tr>
<tr>
<td>6.11.110</td>
<td>RawValueChangedEventArgs Class</td>
<td></td>
</tr>
<tr>
<td>6.11.111</td>
<td>ReadOnlyDataTypeCollection Class</td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6.11.112</td>
<td>ReadOnlyDimensionCollection Class</td>
<td>2295</td>
</tr>
<tr>
<td>6.11.113</td>
<td>ReadOnlyEnumValueCollection Class</td>
<td>2300</td>
</tr>
<tr>
<td>6.11.114</td>
<td>ReadOnlyEnumValueCollection.T. Class</td>
<td>2309</td>
</tr>
<tr>
<td>6.11.115</td>
<td>ReadOnlyFieldCollection Class</td>
<td>2317</td>
</tr>
<tr>
<td>6.11.116</td>
<td>ReadOnlyMemberCollection Class</td>
<td>2322</td>
</tr>
<tr>
<td>6.11.117</td>
<td>ReadOnlyMethodParameterCollection Class</td>
<td>2327</td>
</tr>
<tr>
<td>6.11.118</td>
<td>ReadOnlyRpcMethodCollection Class</td>
<td>2330</td>
</tr>
<tr>
<td>6.11.119</td>
<td>ReadOnlySymbolCollection Class</td>
<td>2336</td>
</tr>
<tr>
<td>6.11.120</td>
<td>ReadOnlyTypeAttributeCollection Class</td>
<td>2340</td>
</tr>
<tr>
<td>6.11.121</td>
<td>ResultDataTypes Class</td>
<td>2347</td>
</tr>
<tr>
<td>6.11.122</td>
<td>ResultDynamicSymbols Class</td>
<td>2350</td>
</tr>
<tr>
<td>6.11.123</td>
<td>ResultSymbols Class</td>
<td>2353</td>
</tr>
<tr>
<td>6.11.124</td>
<td>ResultSymbols.T. Class</td>
<td>2356</td>
</tr>
<tr>
<td>6.11.125</td>
<td>RpcInvokeException Class</td>
<td>2359</td>
</tr>
<tr>
<td>6.11.126</td>
<td>RpcMethodCollection Class</td>
<td>2364</td>
</tr>
<tr>
<td>6.11.127</td>
<td>RpcMethodNotSupportedException Class</td>
<td>2379</td>
</tr>
<tr>
<td>6.11.128</td>
<td>RpcMethodParameterCollection Class</td>
<td>2384</td>
</tr>
<tr>
<td>6.11.129</td>
<td>StringConvertMode Enumeration</td>
<td>2395</td>
</tr>
<tr>
<td>6.11.130</td>
<td>SymbolAccessRights Enumeration</td>
<td>2396</td>
</tr>
<tr>
<td>6.11.131</td>
<td>SymbolCollection Class</td>
<td>2396</td>
</tr>
<tr>
<td>6.11.132</td>
<td>SymbolException Class</td>
<td>2401</td>
</tr>
<tr>
<td>6.11.133</td>
<td>TypeAttribute Class</td>
<td>2413</td>
</tr>
<tr>
<td>6.11.134</td>
<td>TypeAttributeCollection Class</td>
<td>2418</td>
</tr>
<tr>
<td>6.11.135</td>
<td>ValueChangedBaseEventArgs Class</td>
<td>2435</td>
</tr>
<tr>
<td>6.11.136</td>
<td>ValueChangedEventArgs Class</td>
<td>2439</td>
</tr>
<tr>
<td></td>
<td><strong>6.12 TwinCAT.TypeSystem.Generic Namespace</strong></td>
<td></td>
</tr>
<tr>
<td>6.12.1</td>
<td>DataTypeCollection.T. Class</td>
<td>2441</td>
</tr>
<tr>
<td>6.12.2</td>
<td>INamespace.TType. Interface</td>
<td>2442</td>
</tr>
<tr>
<td>6.12.3</td>
<td>InstanceCollection.T. Class</td>
<td>2458</td>
</tr>
<tr>
<td>6.12.4</td>
<td>ISymbolProvider.TNamespace, TDataType, TSymbol. Interface</td>
<td>2460</td>
</tr>
<tr>
<td>6.12.5</td>
<td>NamespaceCollection.T. Class</td>
<td>2481</td>
</tr>
<tr>
<td>6.12.6</td>
<td>ReadOnlyDataTypeCollection.T. Class</td>
<td>2484</td>
</tr>
<tr>
<td>6.12.7</td>
<td>ReadOnlyInstanceCollection.T. Class</td>
<td>2499</td>
</tr>
<tr>
<td>6.12.8</td>
<td>ReadOnlyNamespaceCollection.T. Class</td>
<td>2505</td>
</tr>
<tr>
<td>6.12.9</td>
<td>ReadOnlySymbolCollection.T. Class</td>
<td>2515</td>
</tr>
<tr>
<td>6.12.10</td>
<td>SymbolCollection.T. Class</td>
<td>2522</td>
</tr>
<tr>
<td>6.12.11</td>
<td>SymbolIterationMask Enumeration</td>
<td>2526</td>
</tr>
<tr>
<td>6.12.12</td>
<td>SymbolIterator.T. Class</td>
<td>2532</td>
</tr>
<tr>
<td></td>
<td><strong>6.13 TwinCAT.ValueAccess Namespace</strong></td>
<td></td>
</tr>
<tr>
<td>6.13.1</td>
<td>CannotAccessValueException Class</td>
<td>2540</td>
</tr>
<tr>
<td>6.13.2</td>
<td>IAccessorRawValue Interface</td>
<td>2541</td>
</tr>
<tr>
<td>6.13.3</td>
<td>IAccessorValueFactory Interface</td>
<td>2544</td>
</tr>
<tr>
<td>6.13.4</td>
<td>ResultAccess Class</td>
<td>2552</td>
</tr>
<tr>
<td>6.13.5</td>
<td>ResultReadDynamicValueAccess Class</td>
<td>2556</td>
</tr>
<tr>
<td>6.13.6</td>
<td>ResultReadRawAccess Class</td>
<td>2562</td>
</tr>
<tr>
<td>6.13.7</td>
<td>ResultReadValueAccess Class</td>
<td>2564</td>
</tr>
<tr>
<td></td>
<td><strong>6.14 TwinCAT.SymbolAccess Namespace</strong></td>
<td></td>
</tr>
<tr>
<td>6.14.1</td>
<td>SymbolException Class</td>
<td>2566</td>
</tr>
<tr>
<td></td>
<td><strong>6.15 TwinCAT.SymbolIteration Namespace</strong></td>
<td></td>
</tr>
<tr>
<td>6.15.1</td>
<td>SymbolIterationMask Enumeration</td>
<td>2582</td>
</tr>
<tr>
<td></td>
<td><strong>6.16 TwinCAT.SymbolProvider Namespace</strong></td>
<td></td>
</tr>
<tr>
<td>6.16.1</td>
<td>ReadonlyProvider.GetSymbolProviderClass</td>
<td>2588</td>
</tr>
<tr>
<td>6.16.2</td>
<td>SymbolProvider.GetSymbolProviderClass</td>
<td>2593</td>
</tr>
<tr>
<td></td>
<td><strong>6.17 TwinCAT.SymbolProvider Namespace</strong></td>
<td></td>
</tr>
<tr>
<td>6.17.1</td>
<td>SymbolProvider.GetSymbolProviderClass</td>
<td>2641</td>
</tr>
<tr>
<td>6.17.2</td>
<td>SymbolProvider.GetSymbolProviderClass</td>
<td>2642</td>
</tr>
<tr>
<td></td>
<td><strong>6.18 TwinCAT.SymbolNamespace Namespace</strong></td>
<td></td>
</tr>
<tr>
<td>6.18.1</td>
<td>SymbolNamespace.GetSymbolNamespaceClass</td>
<td>2643</td>
</tr>
<tr>
<td>6.18.2</td>
<td>SymbolNamespace.GetSymbolNamespaceClass</td>
<td>2644</td>
</tr>
<tr>
<td></td>
<td><strong>6.19 TwinCAT.SymbolNamespace Namespace</strong></td>
<td></td>
</tr>
<tr>
<td>6.19.1</td>
<td>SymbolNamespace.GetSymbolNamespaceClass</td>
<td>2645</td>
</tr>
<tr>
<td>6.19.2</td>
<td>SymbolNamespace.GetSymbolNamespaceClass</td>
<td>2646</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>6.13.8</td>
<td>ResultReadValueAccess.T. Class</td>
<td>2568</td>
</tr>
<tr>
<td>6.13.9</td>
<td>ResultRpcMethodAccess Class</td>
<td>2571</td>
</tr>
<tr>
<td>6.13.10</td>
<td>ResultWriteAccess Class</td>
<td>2575</td>
</tr>
<tr>
<td>6.13.11</td>
<td>SymbolNotificationTypes Enumeration</td>
<td>2579</td>
</tr>
<tr>
<td>6.13.12</td>
<td>ValueCreationModes Enumeration</td>
<td>2580</td>
</tr>
</tbody>
</table>
1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement. No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
with corresponding applications or registrations in various other countries.

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.
The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.
Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.
1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>DANGER</td>
<td>Serious risk of injury! Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.</td>
<td></td>
</tr>
<tr>
<td>WARNING</td>
<td>Risk of injury! Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.</td>
<td></td>
</tr>
<tr>
<td>CAUTION</td>
<td>Personal injuries! Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.</td>
<td></td>
</tr>
<tr>
<td>NOTE</td>
<td>Damage to the environment or devices Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.</td>
<td></td>
</tr>
</tbody>
</table>

Tip or pointer

This symbol indicates information that contributes to better understanding.
2 TwinCAT ADS .NET API Documentation

The TwinCAT .NET API implements support for the TwinCAT Automation Device specification (ADS). It can be used within .NET Framework programming languages and support ADS Client implementations. The ADS API interface permits:

- The Implementation of ADS Clients
- Browsing of (ADS) server side symbolic information.
- Reading and writing ProcessImage information in Raw or in type safe manner.
- Receiving ADS Notifications as events.

Getting Started

For getting started please have a look at the following documents:

- Prerequisites [16]
- Installation [16]
- Concepts [19]
- HowTo Samples [38]

Background information about the TwinCAT ADS protocol can be found here:

- ADS Introduction
- TwinCAT ADS Device concept

for common ADS information.

From the conceptual standpoint within this ADS .NET API reference documentation, the most important starting points for reading are the following Classes/Methods:

Main documentation entry points

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addressing ADS Devices via AmsNetId Address information.</td>
<td>AmsNetId [665]</td>
</tr>
<tr>
<td>Communicate to ADS Devices via the AdsClient class.</td>
<td>AdsClient [154]</td>
</tr>
<tr>
<td>Implement your own (virtual) ADS Server</td>
<td>AdsServer [1121]</td>
</tr>
<tr>
<td>Browse target system symbolic information. Creation of the SymbolLoader via Factory class.</td>
<td>SymbolLoaderFactory.Create(IConnection,ISymbolLoaderSettings) [1524]</td>
</tr>
<tr>
<td>Session and Connection management on top of the ADS communication channel established by the AdsClient.</td>
<td>AdsSession [603], AdsConnection [357]</td>
</tr>
<tr>
<td>Usage of Ads.Rx (Reactive extensions) to write reactive code.</td>
<td>AdsClientExtensions [1056]. AdsClientExtensions [1056]</td>
</tr>
</tbody>
</table>

The ADS reactive extensions are only available for usage, when the corresponding Nuget package is downloaded from Nuget.org. Beckhoff.TwinCAT.Ads.Reactive package on Nuget.
2.1 Prerequisites

Preconditions for installing/Using the TwinCAT .NET ADS Communication API Version 5.X

To develop an application the TwinCAT .NET ADS Communication API the following software infrastructure must be available on the development system.

- An ADS/AMS Router (e.g. TwinCAT >= 4024.10) that is capable to Route the ADS/AMS Frames between systems.
- The .NET SDK to develop the application software.
- ADS Nuget packages referenced within the application project to develop TwinCAT ADS data exchange / communication.

**ADS/AMS Router**

- A TwinCAT installation Version >= 4024.10.
  
  Because the Beckhoff.TwinCAT.Ads Version 5.X uses internal interfaces that are available only from TwinCAT 4024.10 on, an appropriate version must be installed locally.
  
  The package doesn't work with older installations - the actually only alternative is the AdsRouterConsole.

- Or alternativly a running instance of the 'Beckhoff.TwinCAT.Ads.TcpRouter' for customized router implementations or the 'ready-out-of-the-box' router package 'Beckhoff.TwinCAT.Ads.AdsRouterConsole' for use on system without TwinCAT.

**A Software development kit for .NET (SDK)**

At least one of the following SDKs and target frameworks is necessary:

- .NET 5.0 or later
- .NET Core 3.1 or later
- .NET Framework 4.61 or later
- .NET Standard 2.0 compatible SDK or later

**The Beckhoff.TwinCAT.Ads components for the application.**

The following Nuget packages must be added to the application project as package references:

- The 'Beckhoff.TwinCAT.Ads Nuget' package.

2.2 Installation

The now preferred way to install the TwinCAT ADS .NET Communication API is to use the NuGet.org package manager.

**Beckhoff.TwinCAT.Ads package from Nuget.org repository.**

This is the main package implementing the ADS client functionality. This is needed to establish ADS connections to local and remote devices.

Please follow the install instructions on the Nuget.org site.
Beckhoff.TwinCAT.Ads Nuget Package


This package installs Reactive extensions on top of the Beckhoff.TwinCAT.Ads Nuget Package and installs additional extensions to map ADSNotifications to observable events.

Please follow the install instructions on the Nuget.org site.

Beckhoff.TwinCAT.Ads.Reactive Nuget Package
3 Version History

The topics in this section describe the various changes made to the 'TwinCAT.Ads .NET API' over the life of the project.

Version History

- Version 5.0.X

Other Resources

TwinCAT ADS .NET API Documentation

3.1 Version 5.0.X

Feature milestones of the version 5.X.X Series

Milestones of Version 5.0.0.0

- Target support for netstandard2.0, net461, netcoreapp3.1, net50
- Asynchronous programming model (support of the async/await statements, seealso Concept of async operation)
- Integrated support for implementing customized AdsServers (AdsServer)
- Optimized interfaces enhancing scalability and performance (see Span.T. and Memory.T.).

Other Resources

Version History
4 Concepts

Concepts introduction

Concepts discussed

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndexGroup / IndexOffset</td>
<td>Read/Write values by IndexGroup / IndexOffset [19]</td>
</tr>
<tr>
<td>Symbolic access</td>
<td>Read/Write values by symbolic instance path [25]</td>
</tr>
<tr>
<td>Access via symbol handle</td>
<td>Read/Write values by symbol handles [26]</td>
</tr>
<tr>
<td>Asynchronous programming (async, await)</td>
<td>Concept of async operation [20]</td>
</tr>
<tr>
<td>ADS Notifications</td>
<td>Concept of ADS Notifications [22]</td>
</tr>
<tr>
<td>Symbol loader access</td>
<td>Access symbolic information by Symbol loader [31]</td>
</tr>
<tr>
<td>Marshalling values via ANYTYPE concept</td>
<td>ANY Type blittable type marshalling [27]</td>
</tr>
<tr>
<td>Dynamic automatic marshalling via Dynamic Language Runtime</td>
<td>Use of the .NET Framework Dynamic Language Runtime (DLR) [32]</td>
</tr>
<tr>
<td>Access via reactive extensions</td>
<td>Observer value changes by ADS Reactive Extensions [35]</td>
</tr>
</tbody>
</table>

Other Resources

TwinCAT ADS .NET API Documentation [15]

4.1 Access Data via IndexGroup/IndexOffset

Reading/Writing values by Index/Group index offset are the most basic way to access data via ADS. This address combination directly links into the process image of virtual ADS Devices.

As long the process image is static this is unproblematic and a system near access, but if the content is more dynamic and the address changes over time the IndexGroup/IndexOffset can get invalid.

Examples about moving addresses could be:

- Changed Parametrization of IO (and Re-activation)
- The PLC Online change
- New Plc Downloads

In that case other access methods could be advantageous.

Another important point is that the data access is not type safe. The values are read or written to or from byte buffers and the proper marshalling/unmarshalling is the task of the application code.

Asynchronous access

Access ProcessImage Data by IndexGroup/IndexOffset

```csharp
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    UInt32 valueToRead = 0;
    UInt32 valueToWrite = 42;
```
client.Connect(AmsNetId.Local, 851);
byte[] writeData = new byte[sizeof(uint)];

// Write an UINT32 Value
MemoryStream writeStream = new MemoryStream(writeData);
BinaryWriter writer = new BinaryWriter(writeStream);
writer.Write(valueToWrite);
ResultWrite resultWrite = await client.WriteAsync(0x4020, 0x0, writeData.AsMemory(), cancel);

// Read an UINT32 Value
byte[] readData = new byte[sizeof(uint)];
ResultRead resultRead = await client.ReadAsync(0x4020, 0x0, readData.AsMemory(), cancel);

MemoryStream readStream = new MemoryStream(readData);
BinaryReader reader = new BinaryReader(readStream);
valueToRead = reader.ReadUInt32();
}

Synchronous access

Access ProcessImage Data by IndexGroup/IndexOffset

using (AdsClient client = new AdsClient())
{
    UInt32 valueToRead = 0;
    UInt32 valueToWrite = 42;
    client.Connect(AmsNetId.Local, 851);
    // Write an UINT32 Value
    byte[] writeData = new byte[sizeof(uint)];
    MemoryStream writeStream = new MemoryStream(writeData);
    BinaryWriter writer = new BinaryWriter(writeStream);
    writer.Write(valueToWrite);
    client.Write(0x4020, 0x0, writeData);
    // Read an UINT32 Value
    byte[] readData = new byte[sizeof(uint)];
    int readBytes = client.Read(0x4020, 0x0, readData);
    MemoryStream readStream = new MemoryStream(readData);
    readStream.Position = 0;
    BinaryReader reader = new BinaryReader(readStream);
    valueToRead = reader.ReadUInt32();
}

4.2 Async programming (async, await)

Since .Version 4.0, the .NET API supports concurrent operation in form of the compiler support of the async/await statements. This is a special code generation supported flavor of concurrency that uses so called 'futures'. A 'future' (or promise) is a type that represents an operation that will be completed in the future and is represented by the .NET type Task or Task<TResult> type. This ensures that the called asynchronous method is started on call and delivers its result later on, without blocking the calling thread. As consequence the calling thread is able to process other work in the meanwhile. The deep support level in the .NET framework and the underlying code generation makes asynchronous programming almost as easy as synchronous programming.

From version 5.0.0 on, the TwinCAT.Ads API also supports the async programming model.

The main advantages are:
- Remains responsiveness of GUI applications during ADS communication
- Asynchronous programming enables scalability (mainly on the Server side)
- Easy synchronization between threads, nearly as simple as asynchronous programming.
- Functional, stateless programming style seamlessly integrating with other concurrency techniques like parallel programming (.NET Task Parallel Library) or reactive programming (Reactive Extensions [35]).

More about asynchronous programming can be read here: Asynchronous programming
and here: The Task asynchronous programming model in C#

Example

Read/Write AnyType by IndexGroup/IndexOffset (asynchronously)

```csharp
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;
    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);
    ResultWrite resultWrite = await client.WriteAnyAsync(0x4020, 0x0, valueToWrite, cancel);
    bool succeeded = resultWrite.Succeeded;

    ResultValue<uint> resultRead = await client.ReadAnyAsync<uint>(0x4020, 0x0, cancel);
    if (resultRead.Succeeded)
    {
        valueToRead = (uint)resultRead.Value;
    }
}
```

Read/Write AnyType by variable handle (asynchronously)

```csharp
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;
    uint varHandle = 0;
    client.Connect(AmsNetId.Local, 851);

    uint valueToRead = 0;
    uint valueToWrite = 42;

    ResultHandle resultHandle = await client.CreateVariableHandleAsync("MAIN.nCounter", cancel);
    varHandle = resultHandle.Handle;

    if (resultHandle.Succeeded)
    {
        try
        {
            ResultWrite resultWrite = await client.WriteAnyAsync(varHandle, valueToWrite, cancel);
            ResultValue<uint> resultRead = await client.ReadAnyAsync<uint>(varHandle, cancel);

            if (resultRead.Succeeded)
            {
                valueToRead = resultRead.Value;
            }
        }
        finally
        {
            // Unregister VarHandle after Use
            ResultAds result = await client.DeleteVariableHandleAsync(varHandle, cancel);
        }
    }
}
```

Read/Write AnyType by SymbolBrowser (asynchronously)

```csharp
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;
    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    // Load all Symbols + DataTypes
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);

    ResultSymbols resultSymbols = await loader.GetSymbolsAsync(cancel);
    if (resultSymbols.Succeeded)
    {
        Symbol symbol = (Symbol)resultSymbols.Symbols["MAIN.nCounter"];     
```
4.3 Use of ADS Notifications

If values from a PLC or NC are to be displayed continuously on a user interface, then it is very inefficient to use asynchronous read access, since this function must be called cyclically (polled triggered by a timer). Instead of using a pulling (read) model, ADS Notifications are implementing a push model. That means ADS Notifications are triggered by the sender and form a single or series of ADS messages/events. Together with these ADS Notifications, values can be transmitted. A distinction is drawn between whether the TwinCAT server is to transmit the values cyclically, or only when the values change.

In principle (raw mode) a notification is begun with the registration of the notification:[M:Twincat.Ads.AdsClient.AddDeviceNotificationAsync(System.UInt32,System.UInt32,int,Twincat.Ads.NotificationSettings,System.Object, System.Threading.CancellationToken)](asynchronous) or AddDeviceNotification(synchronous). After this, events are automatically fired by TwinCAT. DeleteDeviceNotificationAsync (asynchronous) or DeleteDeviceNotification (synchronous) is used to halt the notification again. Since the number of notifications is limited, you should ensure the notifications no longer required by your program are unregistered/deleted.

There exist several 'modes' for different type of ADS Notification triggers. For a complete list please consult AdsTransMode [639].

All the following examples demonstrate how to receive ADS Notifications. The .NET ADS API supports different information layers which different levels of ADS Notification support. All are using a PLC variable in the PLC and each time the value of the PLC variable changes, an ADS Notification message is sent and the registered callback method is invoked with event arguments that contain all the necessary information (value, time stamp, ...).

Hint: Don't use time intensive executions or ADS commands inside of your callback (not more than approx. 500). Remind to sync your callback in your main thread (typically the UI thread) if necessary, because the ADS Notifications appear on a background thread.

Using ADS Notifications with Symbolic information

C#

private void SymbolValueChanged()
{
    using (AdsClient client = new AdsClient())
    {
        // Connect to target
        client.Connect(AmsNetId.Local, 851);
        Symbol symbol = null;
        try
        {
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
            // DINT Type (UINT32)
            symbol = (Symbol)loader.Symbols["MAIN.nCounter"];
            if (resultRead.Succeeded)
                valueToRead = (uint)resultRead.Value;
        }
    }
}
private void Symbol_ValueChanged(object sender, ValueChangedEventArgs e)
{
    Symbol symbol = (Symbol)e.Symbol;
    // Object Value can be cast to int automatically, because it is an Primitive Value (DINT -- > Int32).
    // The Symbol information is used internally to cast the value to its appropriate .NET Type.
    int iVal = (int)e.Value;
    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread
    /*SynchronizationContext syncContext = SynchronizationContext.Current;
     _context.Post(status => someLabel.Text = iVal.ToString(), null); // Non-blocking post */
}

private void Client_AdsNotification(object sender, AdsNotificationExEventArgs e)
{
    // Or here we know about UDINT type --> can be marshalled as UINT32
    uint nCounter = (uint)e.Value;
    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread
    /*SynchronizationContext syncContext = SynchronizationContext.Current;
     _context.Post(status => someLabel.Text = nCounter.ToString(), null); // Non-blocking post */
}
Asynchronous registering of Notifications

Trigger on changed values by ADS Notifications

```csharp
private async Task RegisterNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;

    using (AdsClient client = new AdsClient())
    {
        // Add the Notification event handler
        client.AdsNotification += Client_AdsNotification2;

        // Connect to target
        client.Connect(AmsNetId.Local, 851);
        uint notificationHandle = 0;

        // Notification to a DINT Type (UINT32)
        // Check for change every 200 ms
        byte[] notificationBuffer = new byte[sizeof(UInt32)];
        int size = sizeof(UInt32);

        ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", size, new NotificationSettings(AdsTransMode.OnChange, 200, 0), null, cancel);

        if (result.Succeeded)
        {
            notificationHandle = result.Handle;
            await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.

            // Unregister the Event / Handle
            ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
        }

        client.AdsNotification -= Client_AdsNotification2;
    }
}
```

private void Client_AdsNotification2(object sender, AdsNotificationEventArgs e)
{
    // Or here we know about UDINT type --> can be marshalled as UINT32
    uint nCounter = BinaryPrimitives.ReadUInt32LittleEndian(e.Data.Span);

    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread
    /*SynchronizationContext syncContext = SynchronizationContext.Current;
     _context.Post(status => someLabel.Text = nCounter.ToString(), null); // Non-blocking post */
}

Synchronous registering of Notifications

Trigger on changed values by ADS Notifications

```csharp
private void RegisterNotifications()
{
    using (AdsClient client = new AdsClient())
    {
        // Add the Notification event handler
        client.AdsNotification += Client_AdsNotification;

        // Connect to target
        client.Connect(AmsNetId.Local, 851);
        uint notificationHandle = 0;

        try
        {
            // Notification to a DINT Type (UINT32)
            // Check for change every 200 ms
            int size = sizeof(UInt32);
            byte[] notificationBuffer = new byte[sizeof(UInt32)];

            notificationHandle = client.AddDeviceNotification("MAIN.nCounter", size, new NotificationSettings(AdsTransMode.OnChange, 200, 0), null);
            Thread.Sleep(5000); // Sleep the main thread to get some (asynchronous Notifications)
        }
        finally
    }
```
private void Client_AdsNotification(object sender, AdsNotificationEventArgs e)
{
    // Or here we know about UDINT type --> can be marshalled as UINT32
    uint nCounter = BinaryPrimitives.ReadUInt32LittleEndian(e.Data.Span);
    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread
    /*SynchronizationContext syncContext = SynchronizationContext.Current;
     _context.Post(status => someLabel.Text = nCounter.ToString(), null); // Non-blocking post */
}

Using reactive ADS Notifications

C#

// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf o[1].CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
        Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToString()));
    });

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 500, 5000); // optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.WhenValueChanged().Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}

4.4 Access Data via Symbolic path

The Read/Write Access by symbol path solves the issue of directly accessing the process image. With specifying the access path to the symbol, the symbol address can be found by a binary search (internally) and reading / writing symbols is independent of the location within the process image.

This access method can only be used, when the ADS device is supporting symbolic information like the TwinCAT PLC.

Because its indirect access, the performance is slightly worse than the direct access via IndexGroup/IndexOffset. However there are internal optimizations to cache handles to the already used symbols to accelerate repeated access.
Asynchronous access

Access symbolic data by instance/symbol path

```csharp
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;
    uint valueToRead = 0;
    uint valueToWrite = 42;
    client.Connect(AmsNetId.Local, 851);
    ResultWrite resultWrite = await client.WriteValueAsync("MAIN.nCounter", valueToWrite, cancel);
    ResultValue<uint> resultRead = await client.ReadValueAsync<uint>("MAIN.nCounter", cancel);
    if (resultRead.Succeeded)
    {
        valueToRead = resultRead.Value;
    }
}
```

Synchronous access

Access symbolic data by instance/symbol path

```csharp
using (AdsClient client = new AdsClient())
{
    uint valueToRead = 0;
    uint valueToWrite = 42;
    client.Connect(AmsNetId.Local, 851);
    client.WriteValue("MAIN.nCounter", valueToWrite);
    valueToRead = (uint)client.ReadValue("MAIN.nCounter", typeof(uint));
}
```

4.5 Access Data via Symbol handles

The Read/Write Access by handle solves the issue of directly accessing the process image like the symbol path access. Because the address is accessed indirectly by the symbol path creating a variable handle, the read/write works also when the data object has changed its position within the process image.

However the cost for this are two extra ADS communication roundtrips by the 'CreateVariableHandle' and 'DeleteVariableHandle' calls compared to the IndexGroup/IndexOffset access methods. It is the responsibility of the application code to optimize these accesses.

Asynchronous access

Access symbolic values by handle

```csharp
CancellationToken cancelT = CancellationToken.None;
using (AdsClient client = new AdsClient())
{
    ushort valueToRead = 0; // System.UInt16
    ushort valueToWrite = 42; // System.UInt16

    // Create the Variable Handle
    ResultHandle resultHandle = await client.CreateVariableHandleAsync("MAIN.testVar", cancel); // Test Var is defined as PLC INT
    if (resultHandle.Succeeded)
    {
        uint varHandle = 0;
        try
        {
            // Write an UINT16 Value
            byte[] writeData = new byte[sizeof(ushort)];
            MemoryStream writeStream = new MemoryStream(writeData);
            BinaryWriter writer = new BinaryWriter(writeStream);
            writer.Write(valueToWrite); // Marshal the Value
            ResultWrite resultWrite = await client.WriteAsync(varHandle, writeData.AsMemory(), cancelT);
```
bool succeeded = resultWrite.Succeeded;

// Read an UINT16 Value
byte[] readData = new byte[sizeof(ushort)];
ResultRead resultRead = await client.ReadAsync(varHandle, readData.AsMemory(), cancel);
if (resultRead.Succeeded)
{
    MemoryStream readStream = new MemoryStream(readData);
    BinaryReader reader = new BinaryReader(readStream);
    valueToRead = reader.ReadUInt16(); // Unmarshal the Value
}
finally
{
    // Unregister VarHandle after Use
    ResultAds result = await client.DeleteVariableHandleAsync(varHandle, cancel);
}
}

Synchronous access

Access symbolic values by handle
using (AdsClient client = new AdsClient())
{
    uint varHandle = 0;
    client.Connect(AmsNetId.Local, 851);
    try
    {
        UInt16 valueToRead = 0;
        UInt16 valueToWrite = 42;

        // Create the Variable Handle
        varHandle = client.CreateVariableHandle("MAIN.testVar"); // Test Var is defined as PLC INT

        // Write an UINT16 Value
        byte[] writeData = new byte[sizeof(ushort)];
        MemoryStream writeStream = new MemoryStream(writeData);
        BinaryWriter writer = new BinaryWriter(writeStream);
        writer.Write(valueToWrite); // Marshal the Value
        client.Write(varHandle, writeData.AsMemory());

        // Read an UINT16 Value
        byte[] readData = new byte[sizeof(ushort)];
        MemoryStream readStream = new MemoryStream(readData);
        client.Read(varHandle, readData.AsMemory());
        BinaryReader reader = new BinaryReader(readStream);
        valueToRead = reader.ReadUInt16(); // Unmarshal the Value
    }
    finally
    {
        // Unregister VarHandle after Use
        client.DeleteVariableHandle(varHandle);
    }
}

4.6 Value marshalling with ANYTYPE concept

This topic describes reading and writing variables/symbols of 'any' type with the help of the ReadAny and WriteAny (ReadSymbol, WriteSymbol) methods. The value will be marshalled / cast directly from/to its appropriate .NET type, what eases the value access.

'Any' types in this context are all types that are 'blittable' to the process image - what means that the memory layout on both sides of the data transfer is equal (e.g. some primitive types) or can be marshalled by the marshalling mechanisms of .NET (see 'PlcStruct' in the example below). The memory layout specification can be customized with the 'System.Runtime.InteropServices.StructLayoutAttribute' on the .NET side (see MSDN) and the 'pack_mode' attribute on the TwinCAT PLC Side (TwinCAT 3). TwinCAT 2 only supports a memory layout of PACK = 1.
The appropriate .NET type must be known during compile time and is passed to the methods as parameter. In case of a ReadAny call, the read data will be returned as a object. The type of the object is marshalled to the type specified as parameter type. Because the data size and the memory alignment is taken from this type specification, it is so important that this specification fits to the memory representation in the ADS device (e.g. the PLC).

Because some data types (arrays and strings) need additional information, an overload of the method ReadAny exists, that takes an additional parameter args. A full list of supported types can be found in the documentation of the overloaded method.

### Reading and writing of structures

To be able to read or write PLC structures the memory layout of the structure or class in .NET must be the same as in the PLC. The layout of a structure or class can be specified with the attribute StructLayoutAttribute. The LayoutKind must be set to LayoutKind.Sequential and the pack must be set to 1

If arrays, strings or boolean values are define the class, one has to specify how these fields should be marshalled. This is accomplished with help of the MarshalAs attribute. Because arrays and strings do not have a fixed length in .NET, the property SizeConst is necessary for arrays and strings.

### Marshalling values with ‘ANY_TYPES’ asynchronously

```csharp
using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851);
    // Bool value
    bool boolValue = false;
    ResultAnyValue resultBoolValue = await client.ReadValueAsync("MAIN.bool1", typeof(bool), cancel);
    boolValue = (bool)resultBoolValue.Value;
    ResultWrite resultWrite = await client.WriteValueAsync("MAIN.bool1", boolValue, cancel);
    // or
    ResultHandle resultHandleBool = await client.CreateVariableHandleAsync("MAIN.bool1", cancel); // BOOL
    resultHandleBool.ThrowOnError(); // or
    if (resultHandleBool.Succeeded)
    {
        ResultAnyValue resultReadBool = await client.ReadAnyAsync(resultHandleBool.Handle, typeof(bool), cancel);
        boolValue = (bool)resultReadBool.Value;
        ResultWrite resultWriteBool = await client.WriteAnyAsync(resultHandleBool.Handle, boolValue, cancel);
    }
    ResultAdS resultHandleDeleteBool = await client.DeleteVariableHandleAsync(resultHandleBool.Handle, cancel);
}

// RealValue
resultHandleReal.ThrowOnError(); // or
if (resultHandleReal.Succeeded)
{
    ResultAnyValue resultReadReal = await client.ReadAnyAsync(resultHandleReal.Handle, typeof(float), cancel); // REAL
    ResultWrite resultWriteReal = await client.WriteAnyAsync(resultHandleReal.Handle, resultReadReal.Value, cancel);
    ResultAdS resultHandleDeleteReal = await client.DeleteVariableHandleAsync(resultHandleReal.Handle, cancel);
}

// String
resultHandleString.ThrowOnError(); // or
if (resultHandleString.Succeeded)
{
    ResultAnyValue resultReadString = await client.ReadAnyAsync(resultHandleString.Handle, typeof(string), cancel); // STRING[80]
    ResultWrite resultWriteString = await client.WriteAnyAsync(resultHandleString.Handle, resultReadString.Value, new int[] { 80 }, cancel);
    ResultAdS resultHandleDeleteString = await client.DeleteVariableHandleAsync(resultHandleString.Handle, cancel);
}
```
```csharp
// ARRAY [0..9] OF UINT
ResultHandle resultHandleArray = await client.CreateVariableHandleAsync("MAIN.uint1Arr", cancel);

if (resultHandleArray.Succeeded)
{
    ResultAnyValue resultReadArray = await client.ReadAnyAsync(resultHandleArray.Handle, typeof(ushort[]), new int[] { 10 }, cancel);

    ushort[] arrayValue = (ushort[])resultReadArray.Value;

    ResultWrite resultWriteArray = await client.WriteAnyAsync(resultHandleArray.Handle, arrayValue, new int[] { 10 }, cancel);

    ResultAds resultHandleDeleteArray = await client.DeleteVariableHandleAsync(resultHandleArray.Handle, cancel);
}

// ARRAY [0..9] OF STRING[80]
ResultHandle resultHandleStringArray = await client.CreateVariableHandleAsync("MAIN.stringArr", cancel);

if (resultHandleStringArray.Succeeded)
{
    ResultAnyValue resultReadStringArray = await client.ReadAnyAsync(resultHandleStringArray.Handle, typeof(string[]), new int[] { 80, 10 }, cancel);

    string[] stringArrValue = (string[])resultReadStringArray.Value;

    ResultWrite resultWriteStringArray = await client.WriteAnyAsync(resultHandleStringArray.Handle, stringArrValue, new int[] { 80, 10 }, cancel);

    ResultAds resultHandleDeleteStringArray = await client.DeleteVariableHandleAsync(resultHandleStringArray.Handle, cancel);
}
```

### Defining Memory layout of struct type.

// Attention: Dependent of the System where the PLC runs, the StructLayout of the exchanged
// Structures must match. With the ANY_TYPE concept this is realized with 'blittable' objects,
// that match on .NET and PLC side.

// Default Pack Modes:
// TC3 I64/x86: Normal, in this case Pack = 8
// TC2 x86: Pack = 1

// On TC3 PLC side we can force the packing of structures with the attribute
// [attribute 'pack_mode' := '1'], see also 'pack_mode' attribute in Beckhoff InfoSystem
// For TC2 is the Pack setting Pack = 1 the only possible way, because it is not selectable.

// We have to ensure that the pack mode on both sides is equal!

[StructLayout(LayoutKind.Sequential, Pack = 1, CharSet = CharSet.Ansi)]
public struct PlcStruct
{
    // Type must be 'blittable' to the corresponding PLC Struct Type
    // See MSDN for MarshalAs and Default Marshalling.

    [MarshalAs(UnmanagedType.I1)]
    public bool boolVal; // BOOL
    public byte byteVal; // BYTE

    // Type must be 'blittable' to the corresponding PLC Struct Type
    // See MSDN for MarshalAs and Default Marshalling.

    [MarshalAs(UnmanagedType.I1)]
    public bool boolVal; // BOOL
    public byte byteVal; // BYTE
}
public ushort ushortVal; // UINT
public short shortVal; // INT
public uint uintVal; // UDINT
public int dintVal; // DINT
public uint uintVal; // UDINT
public float realVal; // REAL
public double lrealVal; // LREAL
[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 81)]
public string stringVal; // STRING[80]
[MarshalAs(UnmanagedType.U4)]
public uint timeVal; // TIME
[MarshalAs(UnmanagedType.U4)]
public uint todVal; // TOD
[MarshalAs(UnmanagedType.U4)]
public uint dateVal; // DATE
[MarshalAs(UnmanagedType.U4)]
public uint dtVal; // DT
}

Marshalling values with 'ANY_TYPES' (synchronous)

using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851);

    // Bool value
    bool boolValue = (bool)client.ReadValue("MAIN.bool1", typeof(bool));
    client.WriteValue("MAIN.bool1", boolValue);

    // or
    uint handleBool = client.CreateVariableHandle("MAIN.bool1"); // BOOL
    boolValue = (bool)client.ReadAny(handleBool, typeof(bool));
    client.WriteAny(handleBool, boolValue);
    client.DeleteVariableHandle(handleBool);

    // RealValue
    uint handleReal = client.CreateVariableHandle("MAIN.real1"); // REAL
    float realValue = (float)client.ReadAny(handleReal, typeof(float));
    client.WriteAny(handleReal, realValue);
    client.DeleteVariableHandle(handleReal);

    // String
    uint handleString = client.CreateVariableHandle("MAIN.string1"); // STRING[80]
    string stringValue = (string)client.ReadAny(handleString, typeof(string), new int[] { 80 }); // Needs additional para for strlen
    client.WriteAny(handleString, stringValue, new int[] { 80 });
    client.DeleteVariableHandle(handleString);

    // ushort[]
    uint handleArray = client.CreateVariableHandle("MAIN.uint1Arr"); // ARRAY [0..9] OF UINT
    ushort[] arrayValue = (ushort[])client.ReadAny(handleArray, typeof(ushort[]), new int[] { 10 });
    client.WriteAny(handleArray, arrayValue, new int[] { 10 });
    client.DeleteVariableHandle(handleArray);

    // Complex Struct Type
    // Take care the the corresponding .NET Type is blittable / marshallable to the PLC type
    PlcStruct structValue = (PlcStruct)client.ReadValue("MAIN.struct");
    client.WriteValue("MAIN.struct", structValue);
    client.DeleteVariableHandle(handleArray);

    // ARRAY [0..9] OF STRING[80]
    // args[0] --> Number of Characters
    // args[1] --> Number of Array Elements
    // Needs additional para for strlen and number of Elements in Array
    uint handleStringArr = client.CreateVariableHandle("MAIN.stringArr"); // ARRAY [0..9] OF STRING[80]
    string[] stringArr = (string[])client.ReadAny(handleStringArr, typeof(string[]), new int[] { 80, 10 });
    client.WriteAny(handleStringArr, stringArr, new int[] { 80, 10 });
    client.DeleteVariableHandle(handleStringArr);
}
**ADS Notifications with Type marshalling (AdsNotificationEx)**

The method AddDeviceNotificationEx is used to register notifications for a PLC variable. If the value of a variable changes the event AdsNotificationEx is fired. The difference to the event AdsNotification, is that the value of the variable is stored in an object instead of in an AdsStream. Therefore one has to pass the type of the object to the method AddDeviceNotificationEx.

**Notifications with 'ANY_TYPES' (asynchronous)**

```csharp
CancellationToken cancel = CancellationToken.None;
using (AdsClient client = new AdsClient())
{
    client.AdsNotificationEx += Client_AdsNotificationEx;
    resultHandle = await client.AddDeviceNotificationExAsync("MAIN.udint", new NotificationSettings(AdsTransMode.OnChange, 200, 200), null, typeof(uint), null, cancel);
    await Task.Delay(5000, cancel); // Wait ....
    resultHandleDelete = await client.DeleteDeviceNotificationAsync(resultHandle.Handle, cancel); // Unregister Event
}
```

**Notifications with 'ANY_TYPES'**

```csharp
private void Client_AdsNotificationEx(object sender, AdsNotificationExEventArgs e)
{
    uint value = (uint)e.Value; // Marshalled value as .NET Type
}
```

**Notifications with 'ANY_TYPES' (synchronous)**

```csharp
using (AdsClient client = new AdsClient())
{
    client.AdsNotificationEx += Client_AdsNotificationEx;
    resultHandle = client.AddDeviceNotificationEx("MAIN.udint", new NotificationSettings(AdsTransMode.OnChange, 200, 200), null, typeof(uint));
    Thread.Sleep(5000); // ...
    client.DeleteDeviceNotification(notificationHandle); // Unregister Event
}
```

### 4.7 Access Data via Symbol Loader

Some ADS Devices (e.g. the TwinCAT PLC) provide symbolic information for download. That means all visible Symbols and DataTypes can be retrieved from the target system. While this need an extra effort to upload and hold the data, this feature helps to remove the dependency of the code/configuration running on the target device.

E.g. because the symbolic information can now be browsed and determined during runtime, the application can be written without knowing what's running on the target system. Even more having the Symbol information cached, the access of the process image data will be easier because the datasize and access (instance path) is stored in the symbol.

Dependent how it is parametrized, the symbol loader can work with 'ANY_TYPES' (marshallable Primitive types, Value marshalling with ANYTYPE concept [p. 27]) or full dynamic symbols (Automatic dynamic marshalling of values [p. 32]).

**Example**

**Accessing symbolic data by preloaded Symbolic information (asynchronous)**

```csharp
using (AdsClient client = new AdsClient())
{
    CancellationTokenToken cancel = CancellationToken.None;
```
Concepts
uint valueToRead = 0;
uint valueToWrite = 42;
client.Connect(AmsNetId.Local, 851);
// Load all Symbols + DataTypes
ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
ResultSymbols resultSymbols

= await loader.GetSymbolsAsync(cancel);

if (resultSymbols.Succeeded)
{
Symbol symbol = (Symbol)resultSymbols.Symbols["MAIN.nCounter"];
// Works for ALL Primitive 'ANY TYPES' Symbols
ResultWriteAccess resultWrite = await symbol.WriteValueAsync(valueToWrite, cancel);
ResultReadValueAccess resultRead = await symbol.ReadValueAsync(cancel);
if (resultRead.Succeeded)
valueToRead = (uint)resultRead.Value;
// Simple filtering of Symbols
Regex filterExpression = new Regex(pattern: @"^MAIN.*"); // Everything that starts with "MAIN"
// FilterFunction that filters for the InstancePath
Func<ISymbol, bool> filter = s => filterExpression.IsMatch(s.InstancePath);
SymbolIterator iterator = new SymbolIterator(symbols: resultSymbols.Symbols, recurse: true, pred
icate: filter);
foreach (ISymbol filteredSymbol in iterator)
{
Console.WriteLine(filteredSymbol.InstancePath);
}
}
}

Accessing symbolic data by preloaded Symbolic information (synchronous)
using (AdsClient client = new AdsClient())
{
uint valueToRead = 0;
uint valueToWrite = 42;
client.Connect(AmsNetId.Local, 851);
// Load all Symbols + DataTypes
ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
Symbol symbol = (Symbol)loader.Symbols["MAIN.nCounter"];
// Works for ALL Primitive 'ANY TYPES' Symbols
symbol.WriteValue(valueToWrite);
valueToRead = (uint)symbol.ReadValue();
// Simple filtering of Symbols
Regex filterExpression = new Regex(pattern: @"^MAIN.*"); // Everything that starts with "MAIN"
// FilterFunction that filters for the InstancePath
Func<ISymbol, bool> filter = s => filterExpression.IsMatch(s.InstancePath);
SymbolIterator iterator = new SymbolIterator(symbols: loader.Symbols, recurse: true, predicate:
filter);
foreach (ISymbol filteredSymbol in iterator)
{
Console.WriteLine(filteredSymbol.InstancePath);
}
}

4.8

Automatic dynamic marshalling of values

The 'Dynamic Symbol Loader' of the .NET ADS Communication API makes use of the .NET dynamic
language runtime (DLR). The dynamic language runtime is a runtime environment that adds a set of services
for dynamic languages to the common language runtime (CLR). The DLR makes it easier to develop
dynamic languages to run on the .NET Framework and to add dynamic features to statically typed
languages.

32

Version: 1.1

TC1000


Dynamic languages can identify the type of an object at run time, whereas in statically typed languages (without using the 'dynamic' keyword, specify object types at design time. The advantage here is - from the moment on the symbolic (and dataType) information is available from the ADS Device - Symbol/Variable values can be marshalled 'on-the-fly' during runtime in a type-safe manner.

This works not only with primitive types but also with complex types. This reduces the complexity of the written application code to access the values, because neither the type of the data must be known, nor how the value data must be marshalled from/to the process image. The price to be paid is simply that the full symbolic information and data types must be downloaded from the ADS Device by the symbol loader.

Example

**Automatic marshalling values with 'Dynamic Values' (asynchronous)**

```csharp
using (AdsClient client = new AdsClient())
{  
    CancellationToken cancel = CancellationToken.None;
    uint valueToRead = 0;
    uint valueToWrite = 42;
    client.Connect(AmsNetId.Local, 851);
    // Load all Symbols + DataTypes
    // Primitive Parts will be automatically resolved to .NET Primitive types.
    IDynamicSymbolLoader loader = (IDynamicSymbolLoader)SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);
    //dynamic symbols = loader.SymbolsDynamic;
    ResultDynamicSymbols resultSymbols = await loader.GetDynamicSymbolsAsync(cancel);
    if (resultSymbols.Succeeded)
    {
        dynamic symbols = resultSymbols.Symbols;
        dynamic main = symbols.Main;
        // Use typed object to use InfoTips
        DynamicSymbol nCounter = main.nCounter; // UDINT
        // or to be fullDynamic
        dynamic nCounter2 = main.nCounter;
        // Works for ALL sorts of types (not restricted to ANY_TYPE basing primitive types)
        ResultReadValueAccess resultRead = await nCounter.ReadValueAsync(cancel);
        if (resultRead.Succeeded)
        {
            valueToRead = (uint)resultRead.Value;
            // or
            var varValue = resultRead.Value;
            // or
            dynamic dynValue = resultRead.Value;
        }
        // Same for writing
        ResultWriteAccess resultWrite = await nCounter.WriteValueAsync(valueToWrite, cancel);
        // Or Notifications / Events (typed dynamically)
        nCounter.ValueChanged += NCounter_ValueChanged;
        //Reading complexTypes e.g. Struct
        DynamicSymbol myStructSymbol = main.plcStruct; // Dynamically created
        ResultReadValueAccess resultRead2 = await myStructSymbol.ReadValueAsync(cancel); // Takes an ADS Snapshot of the value
        if (resultRead2.Succeeded)
        {
            dynamic myStructVal = resultRead2.Value;
            dynamic int1Val = myStructVal.int1; // Value to an INT (short)
            dynamic valueNestedStruct = myStructVal.nestedStruct; //
            value to another complex type (here a nested Struct)
        }
    }
    //wait 5 seconds to get some events
```
Automatic marshalling values with 'Dynamic Values' (handler)

private void NCounter_ValueChanged(object sender, ValueChangedEventArgs e)
{
    var uintVal = e.Value;
}

private void MyStructSymbol_ValueChanged(object sender, ValueChangedEventArgs e)
{
    dynamic structValue = e.Value; // Snapshot of the whole Struct and all its contents
}

Automatic marshalling values with 'Dynamic Values' (synchronous)

using (AdsClient client = new AdsClient())
{
    uint valueToRead = 0;
    uint valueToWrite = 42;
    client.Connect(AmsNetId.Local, 851);

    // Load all Symbols + DataTypes
    // Primitive Parts will be automatically resolved to .NET Primitive types.
    IDynamicSymbolLoader loader = (IDynamicSymbolLoader)SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);
    dynamic symbols = loader.SymbolsDynamic;
    dynamic main = symbols.Main;

    // Use typed object to use InfoTips
    DynamicSymbol nCounter = main.nCounter; // UDINT

    // or to be fullDynamic
    // dynamic nCounter = main.nCounter;

    // Works for ALL sorts of types (not restricted to ANY_TYPE basing primitive types)
    valueToRead = (uint)nCounter.ReadValue(); // or
    var varValue = nCounter.ReadValue(); // or
    dynamic dynValue = nCounter.ReadValue();

    // Same for writing
    nCounter.WriteValue(valueToWrite);

    // Or Notifications / Events
    nCounter.ValueChanged += new EventHandler<ValueChangedEventArgs>(NCounter_ValueChanged);

    //Reading complexTypes e.g. Struct
    DynamicSymbol myStructSymbol = main.plcStruct; // Dynamically created
    dynamic myStructVal = myStructSymbol.ReadValue(); // Takes an ADS Snapshot of the value
    dynamic int1Val = myStructVal.int1; // Value to an INT (short)
    dynamic valueNestedStruct = myStructVal.nestedStruct; // value to another complex type (here a nested Struct)

    myStructSymbol.ValueChanged += new EventHandler<ValueChangedEventArgs>(MyStructSymbol_ValueChanged);
    //wait for notifications for 5 seconds
    Thread.Sleep(5000);
}

Calling 'ReadValue'/ReadValueAsync' or the incoming 'ValueChanged' notification takes a full snapshot (with snapshot time) of the value. That means, when for example subelements of a struct value will be accessed, all subvalues will represent the value of that snapshot time consistently. The starting point (or the instance that caches the consistent data) is always the 'DynamicSymbol' object that called 'ReadValue'.

An update of the value can be done directly on the value with 'UpdateValue', or with reading a new Value on the 'DynamicSymbol' ('ReadValue').

The 'ValueChanged' event on the 'DynamicSymbol' assigns a Notification for just this symbol. The 'ValueChanged' handler will contain the value completely marshalled as dynamic object.
4.9 Reactive Extensions and ADS

The Reactive Extensions (Rx) is a .NET library for composing asynchronous and event-based programs using observable sequences and LINQ-style query operators. Using Rx, developers represent asynchronous data streams with Observables, query asynchronous data streams using LINQ operators, and parameterize the concurrency in the asynchronous data streams using Schedulers. Simply put, Rx = Observables + LINQ + Schedulers.

In ADS terms, not only the reading and writing data or symbol values can be put into reactive data streams, also ADS Notifications are a perfect fit for reactive code. This eases not only data binding to reactive frameworks (e.g. reactive UI) but also supports enhanced data manipulation via synchronous and asynchronous observers. Multithreaded and parallelized code paths that support multiple CPU cores can be written very easily without the burden of deadlock and synchronization issues.

More about .NET reactive extensions can be read here: Reactive extensions project site.

The TwinCAT ADS Reactive extensions are available via a supplement Nuget Package: Beckhoff.TwinCAT.Ads.Reactive Nuget Package

Example

Observe for Notifications

```csharp
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Reactive Notification Handler
    var valueObserver = Observer.Create<ushort>(val =>
    {
        Console.WriteLine(string.Format("Value: {0}", val.ToString()));
    });

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Observe for Symbol Notifications

```csharp
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInfo[1].CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
        Console.WriteLine(string.Format("Instance: {0}, Value: {1}"..cycleCount.InstancePath, val.ToString()));
    });

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransModeOnChange, 500, 5000); // optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.WhenValueChanged().Take(20).Subscribe(valueObserver);
```
Observer for dynamic Symbol Notifications

// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = (IDynamicSymbolLoader)SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);
    dynamic symbols = symbolLoader.SymbolsDynamic;
    dynamic cycleCount = symbols.TwinCAT_SystemInfoVarList._TaskInfo[1].CycleCount;

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
        // Value objects can be dynamically (on the fly) created objects here (e.g. structs)
        Console.WriteLine(string.Format("Instance: {0}, Value: {1}\n", cycleCount.InstancePath, val.ToString()));
    });

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 500, 5000); // optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    // We have to give the 'hint' about IValueSymbol here, that the CLR finds the Extension Method 'WhenValueChanged' during runtime.
    IDisposable subscription = ((IValueSymbol)cycleCount).WhenValueChanged().Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}

Polling observer

// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInfo[1].CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
        Console.WriteLine(string.Format("Instance: {0}, Value: {1}\n", cycleCount.InstancePath, val.ToString()));
    });

    // Take 20 Values in an Interval of 500ms
    IDisposable subscription = cycleCount.PollValues(TimeSpan.FromMilliseconds(500)).Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}

Writing values with observable subject

using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information (Symbol 'i : INT' in PLC Global Variables list.
}
var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
IValueSymbol gvlIntSymbol = (IValueSymbol)symbolLoader.Symbols["GVL.i"]; // Produces object (short) Values 0,1,2,3 ... in seconds period
IObservable<object> timerObservable = Observable.Interval(TimeSpan.FromSeconds(1.0)).Select(i => (object)(short)i);

// Take 10 Values (0..9) and write them to GVL.i
IDisposable dispose = gvlIntSymbol.WriteValues(timerObservable.Take(10));

Console.ReadKey(); // Wait for Key press
dispose.Dispose(); // Dispose the Subscription
5 HowTo Samples

The topics in this section describe the various changes made to the 'TwinCAT.Ads .NET API' over the life of the project.

HowTo

How to use the .NET TwinCAT API

<table>
<thead>
<tr>
<th>Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading/writing (simple) values from/to ADS Servers.</td>
<td>Read/Write values [38]</td>
</tr>
<tr>
<td>Reading/writing string values.</td>
<td>Read/Write string values [42]</td>
</tr>
<tr>
<td>Event driven reading (ADS Notifications)</td>
<td>Event driven reading [49]</td>
</tr>
<tr>
<td>Read/Write data from/to ADS Servers using reactive extensions</td>
<td>Reactive Read/Write [50]</td>
</tr>
</tbody>
</table>

Other Resources

TwinCAT ADS .NET API Documentation [15]

5.1 Read/Write primitive values

Reading writing Values from ADS Devices is the most essential part of the communication API. There are several options for communication with your application.

- Accessing by IndexGroup / IndexOffset
- Symbolic access by instance path and optionally use handles for the symbol
- Holding the overall Symbolic information in the SymbolLoader and use easy access via symbol objects.
- Usage the symbolic interface ITcAdsSymbol when complete SymbolLoading by SymbolLoaders is not appropriate.
- Reading / Writing values as .NET managed Types (primitive types or compound primitive types called ANY_TYPES), or complex dynamic types typesafe generated at runtime.

The following section shows the different scenarios as code snippets.

HowTo Read/Write Values

Read/Write AnyType by IndexGroup/IndexOffset (asynchronously)

```csharp
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;
    uint valueToRead = 0;
    uint valueToWrite = 42;
    client.Connect(AmsNetId.Local, 851);
    ResultWrite resultWrite = await client.WriteAnyAsync(0x4020, 0x0, valueToWrite, cancel);
    bool succeeded = resultWrite.Succeeded;
    ResultValue<uint> resultRead = await client.ReadAnyAsync<uint>(0x4020, 0x0, cancel);
    if(resultRead.Succeeded)
```
Read/Write AnyType by IndexGroup/IndexOffset (synchronously)
using (AdsClient client = new AdsClient())
{
    uint valueToRead = 0;
    uint valueToWrite = 42;
    client.Connect(AmsNetId.Local, 851);
    adsClient.WriteAny(0x4020, 0x0, valueToWrite);
    valueToRead = (uint)adsClient.ReadAny(0x4020, 0x0, typeof(uint));
}

Read/Write AnyType by variable handle (asynchronously)
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;
    uint varHandle = 0;
    client.Connect(AmsNetId.Local, 851);
    uint valueToRead = 0;
    uint valueToWrite = 42;
    ResultHandle resultHandle = await client.CreateVariableHandleAsync("MAIN.nCounter", cancel);
    varHandle = resultHandle.Handle;
    if (resultHandle.Succeeded)
    {
        try
        {
            ResultWrite resultWrite = await client.WriteAnyAsync(varHandle, valueToWrite, cancel);
            ResultValue<uint> resultRead = await client.ReadAnyAsync<uint>(varHandle, cancel);
            if (resultRead.Succeeded)
            {
                valueToRead = resultRead.Value;
            }
        }
        finally
        {
            // Unregister VarHandle after Use
            ResultAds result = await client.DeleteVariableHandleAsync(varHandle, cancel);
        }
    }
}

Read/Write AnyType by variable handle (synchronously)
using (AdsClient client = new AdsClient())
{
    uint varHandle = 0;
    client.Connect(AmsNetId.Local, 851);
    try
    {
        uint valueToRead = 0;
        uint valueToWrite = 42;
        varHandle = client.CreateVariableHandle("MAIN.nCounter");
        adsClient.WriteAny(varHandle, valueToWrite);
        valueToRead = (uint)adsClient.ReadAny(varHandle, typeof(uint));
    }
    finally
    {
        // Unregister VarHandle after Use
        client.DeleteVariableHandle(varHandle);
    }
}

Read/Write AnyType by instance/symbol path (asynchronously)
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;
    uint valueToRead = 0;
    uint valueToWrite = 42;
}
client.Connect(AmsNetId.Local, 851);

resultWrite = await client.WriteValueAsync("MAIN.nCounter", valueToWrite, cancel);
resultRead = await client.ReadValueAsync<uint>("MAIN.nCounter", cancel);

if (resultRead.Succeeded)
  valueToRead = resultRead.Value;
}

**Read/Write AnyType by instance/symbol path (synchronously)**

```csharp
using (AdsClient client = new AdsClient())
{
  uint valueToRead = 0;
  uint valueToWrite = 42;

  client.Connect(AmsNetId.Local, 851);
  client.WriteValue("MAIN.nCounter", valueToWrite);
  valueToRead = (uint)client.ReadValue("MAIN.nCounter", typeof(uint));
}
```

**Read/Write AnyType by IAdsSymbol (asynchronously)**

```csharp
using (AdsClient client = new AdsClient())
{
  CancellationToken cancel = CancellationToken.None;
  uint valueToRead = 0;
  uint valueToWrite = 42;

  client.Connect(AmsNetId.Local, 851);
  ResultValue<IAdsSymbol> resultSymbol = await client.ReadSymbolAsync("MAIN.nCounter", cancel);
  if (resultSymbol.Succeeded)
  {
    ResultWrite resultWrite = await client.WriteValueAsync(resultSymbol.Value, valueToWrite, cancel);
    bool succeeded = resultWrite.Succeeded;

    ResultValue<uint> resultValue = await client.ReadValueAsync<uint>(resultSymbol.Value, cancel);
    if (resultValue.Succeeded)
      valueToRead = resultValue.Value;
  }
}
```

**Read/Write AnyType by IAdsSymbol (synchronously)**

```csharp
using (AdsClient client = new AdsClient())
{
  uint valueToRead = 0;
  uint valueToWrite = 42;

  IAdsSymbol symbol = client.ReadSymbol("MAIN.nCounter");
  client.WriteValue(symbol, valueToWrite);
  valueToRead = (uint)client.ReadValue(symbol);
}
```

**Read/Write AnyType by SymbolBrowser (asynchronously)**

```csharp
using (AdsClient client = new AdsClient())
{
  CancellationToken cancel = CancellationToken.None;
  uint valueToRead = 0;
  uint valueToWrite = 42;

  client.Connect(AmsNetId.Local, 851);
  // Load all Symbols + DataTypes
  ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
  ResultSymbols resultSymbols = await loader.GetSymbolsAsync(cancel);
```
if (resultSymbols.Succeeded)
{
Symbol symbol = (Symbol)resultSymbols.Symbols["MAIN.nCounter"];
// Works for ALL Primitive 'ANY TYPES' Symbols
ResultWriteAccess resultWrite = await symbol.WriteValueAsync(valueToWrite, cancel);
ResultReadValueAccess resultRead = await symbol.ReadValueAsync(cancel);
if (resultRead.Succeeded)
valueToRead = (uint)resultRead.Value;
// Simple filtering of Symbols
Regex filterExpression = new Regex(pattern: @"^ MAIN.*"); // Everything that starts with "MAIN"
// FilterFunction that filters for the InstancePath
Func<ISymbol, bool> filter = s => filterExpression.IsMatch(s.InstancePath);
SymbolIterator iterator = new SymbolIterator(symbols: resultSymbols.Symbols, recurse: true, predicate: filter);
foreach (ISymbol filteredSymbol in iterator)
{
    Console.WriteLine(filteredSymbol.InstancePath);
}
}

Read/Write AnyType by SymbolBrowser (synchronously)
using (AdsClient client = new AdsClient())
{
    uint valueToRead = 0;
    uint valueToWrite = 42;
    client.Connect(AmsNetId.Local, 851);
    // Load all Symbols + DataTypes
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    Symbol symbol = (Symbol)loader.Symbols["MAIN.nCounter"];
    // Works for ALL Primitive 'ANY TYPES' Symbols
    symbol.WriteValue(valueToWrite);
    valueToRead = (uint)symbol.ReadValue();
    // Simple filtering of Symbols
    Regex filterExpression = new Regex(pattern: @"^ MAIN.*"); // Everything that starts with "MAIN"
    // FilterFunction that filters for the InstancePath
    Func<ISymbol, bool> filter = s => filterExpression.IsMatch(s.InstancePath);
    SymbolIterator iterator = new SymbolIterator(symbols: loader.Symbols, recurse: true, predicate: filter);
    foreach (ISymbol filteredSymbol in iterator)
    {
        Console.WriteLine(filteredSymbol.InstancePath);
    }
}

Read/Write dynamic types by SymbolBrowser (asynchronously)
using (AdsClient client = new AdsClient())
{
CancellationToken cancel = CancellationToken.None;
    uint valueToRead = 0;
    uint valueToWrite = 42;
    client.Connect(AmsNetId.Local, 851);
    // Load all Symbols + DataTypes
    // Primitive Parts will be automatically resolved to .NET Primitive types.
    IDynamicSymbolLoader loader = (IDynamicSymbolLoader)SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);
    //dynamic symbols = loader.SymbolsDynamic;
    ResultDynamicSymbols resultSymbols = await loader.GetDynamicSymbolsAsync(cancel);
    if (resultSymbols.Succeeded)
    {
        dynamic symbols = resultSymbols.Symbols; // Symbols collection with 'dynamic' access.
        dynamic main = symbols.Main; // Get the 'dynamic' main FB as Property
    }
Read/Write dynamic types by SymbolBrowser (synchronously)

```csharp
using (AdsClient client = new AdsClient())
{
    uint valueToRead = 0;
    uint valueToWrite = 42;
    client.Connect(AmsNetId.Local, 851);
    // Load all Symbols + DataTypes
    // Primitive Parts will be automatically resolved to .NET Primitive types.
    IDynamicSymbolLoader loader = (IDynamicSymbolLoader)SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);
    dynamic symbols = loader.SymbolsDynamic;
    dynamic main = symbols.Main;
    // Use typed object to use InfoTips
    DynamicSymbol nCounter = main.nCounter;
    // or to be fullDynamic
    //dynamic nCounter = main.nCounter;
    // Works for ALL sorts of types (not restricted to ANY_TYPE basing primitive types)
    nCounter.WriteValue(valueToWrite);
    valueToRead = (uint)nCounter.ReadValue();
}
```

5.2 Read/Write string types

ADS Server usually support strings in 2 flavors. The Default (ANSI) and the Unicode encoding (STRING vs. WSTRING) The ANSI encoding reserves 1 byte per character. Unicode reserves 2.

The strings are of fixed size and therefore the length of the the reserved space within the process image is important.

HowTo Read/Write string values

Reading writing ANSI Streams:

Read/Write ANSI Strings (async)

```csharp
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local port 851 (PLC)
    ResultHandle resultHandle = await client.CreateVariableHandleAsync("MAIN.string", cancel); // Symbol "string" in MAIN defined as string
    if (resultHandle.Succeeded)
    {
        try
        {
```
// Read ANSI String string[80]
int byteSize = 81; // Size of 80 ANSI chars + /0 (STRING[80])
PrimitiveTypeMarshaler converter = new PrimitiveTypeMarshaler(StringMarshaler.DefaultEncoding);
byte[] buffer = new byte[byteSize];

ResultRead resultRead = await client.ReadAsync(resultHandle.Handle, buffer.AsMemory(), cancel);
if (resultRead.Succeeded)
{
    string value = null;
    converter.Unmarshal<string>(buffer.AsSpan(), out value);
    byte[] writeBuffer = new byte[byteSize];
    // Write ANSI String string[80]
    value = "Changed";
    converter.Marshal(value, writeBuffer);
    ResultWrite resultWrite = await client.WriteAsync(resultHandle.Handle, writeBuffer, cancel);
}
finally
{
    ResultAds r1 = await client.DeleteVariableHandleAsync(resultHandle.Handle, cancel);
}

Read/Write ANSI Strings (sync)

using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local port 851 (PLC)
    uint handle = client.CreateVariableHandle("MAIN.string"); // Symbol "string" in MAIN defined as string
    try
    {
        // Read ANSI String string[80]
        int byteSize = 81; // Size of 80 ANSI chars + /0 (STRING[80])
        PrimitiveTypeMarshaler converter = new PrimitiveTypeMarshaler(StringMarshaler.DefaultEncoding);
        byte[] buffer = new byte[byteSize];

        int readBytes = client.Read(handle, buffer.AsMemory());
        string value = null;
        converter.Unmarshal<string>(buffer.AsSpan(), out value);
        // Write ANSI String string[80]
        byte[] writeBuffer = new byte[byteSize];
        value = "Changed";
        converter.Marshal(value, writeBuffer);
        client.Write(handle, writeBuffer);
    }
    finally
    {
        client.DeleteVariableHandle(handle);
    }
}

Reading writing UNICODE Streams:

Read/Write Unicode Strings (async)

CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local port 851 (PLC)
    ResultHandle resultHandle = await client.CreateVariableHandleAsync("MAIN.wstring", cancel); // Symbol "wstring" defined in MAIN as WSTRING
    if (resultHandle.Succeeded)
    {
        try
        {
            // Read UNICODE String wstring[80]
            PrimitiveTypeMarshaler converter = new PrimitiveTypeMarshaler(Encoding.Unicode);
            int byteSize = converter.MarshalSize(80); // Size of 80 UNICODE chars + /0 (WSTRING[80])
            byte[] buffer = new byte[byteSize];
            int readBytes = client.Read(handle, buffer.AsMemory());
            string value = null;
            converter.Unmarshal<string>(buffer.AsSpan(), out value);
            // Write UNICODE String wstring[80]
            byte[] writeBuffer = new byte[byteSize];
            value = "Changed";
            converter.Marshal(value, writeBuffer);
            client.Write(handle, writeBuffer);
        }
        finally
        {
            client.DeleteVariableHandle(handle);
        }
    }
}
byte[] readBuffer = new byte[byteSize];
ResultRead resultRead = await client.ReadAsync(resultHandle.Handle, readBuffer, cancel);
if (resultRead.Succeeded)
{
    string value = null;
    converter.Unmarshal(readBuffer.AsSpan(), out value);
    // Write Unicode String string[80]
    value = "Changed";
    byte[] writeBuffer = new byte[byteSize];
    converter.Marshal(value, writeBuffer.AsSpan());
    ResultWrite resultWrite = await client.WriteAsync(resultHandle.Handle, writeBuffer.AsMemory(canceled), cancel);
    // Write Unicode String string[80]
    value = "Changed";
    byte[] writeBuffer = new byte[byteSize];
    converter.Marshal(value, writeBuffer.AsSpan());
    ResultWrite resultWrite = await client.WriteAsync(resultHandle.Handle, writeBuffer.AsMemory(canceled), cancel);
}
finally
{
    ResultAds r1 = await client.DeleteVariableHandleAsync(resultHandle.Handle, canceled);
}

Read/Write Unicode Strings (sync)

using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local port 851 (PLC)
    uint handle = client.CreateVariableHandle("MAIN.wstring"); // Symbol "wstring" defined in MAIN as WSTRING
    try
    {
        // Read UNICODE String wstr[80]
        PrimitiveTypeMarshaler converter = new PrimitiveTypeMarshaler(Encoding.Unicode);
        int byteSize = converter.MarshalSize(80); // Size of 80 UNICODE chars + /0 (WSTRING[80]) (162)
        byte[] readBuffer = new byte[byteSize];
        int readBytes = client.Read(handle, readBuffer);
        string value = null;
        converter.Unmarshal(readBuffer.AsSpan(), out value);
        // Write Unicode String string[80]
        value = "Changed";
        byte[] writeBuffer = new byte[byteSize];
        converter.Marshal(value, writeBuffer.AsSpan());
        client.Write(handle, writeBuffer.AsMemory());
    }
    finally
    {
        client.DeleteVariableHandle(handle);
    }
}

Reading writing strings with ReadAny/WriteAny group of methods:

Read/Write Anystring (async)

CancellationToken cancel = CancellationToken.None;
using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local port 851 (PLC)
    ResultHandle resultHandleStr = await client.CreateVariableHandleAsync("MAIN.string", cancel); // Symbol "string" defined in MAIN as STRING
    ResultHandle resultHandleWStr = await client.CreateVariableHandleAsync("MAIN.wstring", cancel); // Symbol "wstring" defined in MAIN as WSTRING
    if (resultHandleStr.Succeeded && resultHandleWStr.Succeeded)
    {
        try
        {

        }
    }
}
```csharp
ResultAnyValue resultReadStr = await client.ReadAnyStringAsync(resultHandleStr.Handle, 80, StringMarshaler.DefaultEncoding, cancel);
ResultAnyValue resultReadWStr = await client.ReadAnyStringAsync(resultHandleWStr.Handle, 80, Encoding.Unicode, cancel);

string changedValue = "Changed";

// Attention, take care that the memory of the string in the process image is not exceeded!
ResultWrite resultWriteStr = await client.WriteAnyStringAsync(resultHandleStr.Handle, changedValue, 80, StringMarshaler.DefaultEncoding, cancel);
ResultWrite resultWriteWStr = await client.WriteAnyStringAsync(resultHandleWStr.Handle, changedValue, 80, Encoding.Unicode, cancel);
}
finally
{
    ResultAds r1 = await client.DeleteVariableHandleAsync(resultHandleStr.Handle, cancel);
    ResultAds r2 = await client.DeleteVariableHandleAsync(resultHandleWStr.Handle, cancel);
}
}
```

### Read/Write Anystring (sync)

```csharp
using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local port 851 (PLC)
    uint stringHandle = client.CreateVariableHandle("MAIN.string"); // Symbol "string" defined in MAIN as STRING
    uint wStringHandle = client.CreateVariableHandle("MAIN.wstring"); // Symbol "string" defined in MAIN as WSTRING

    try
    {
        string str = client.ReadAnyString(stringHandle, 80, StringMarshaler.DefaultEncoding);
        string wStr = client.ReadAnyString(wStringHandle, 80, Encoding.Unicode);

        string changedValue = "Changed";

        // Attention, take care that the memory of the string in the process image is not exceeded!
        client.WriteAnyString(stringHandle, changedValue, 80, StringMarshaler.DefaultEncoding);
        client.WriteAnyString(wStringHandle, changedValue, 80, Encoding.Unicode);
    }
    finally
    {
        client.DeleteVariableHandle(stringHandle);
        client.DeleteVariableHandle(wStringHandle);
    }
}
```

### 5.3 Read/Write PlcOpen types (DATE, TIME ...)

The following PLCOpen types have specific representations within the TwinCAT.Ads Communication Library:

- **DT** [1570]
- **DATE** [1552]
- **TIME** [1598]
- **LTIME** [1581]
- **TOD** [1615]

The following section shows the different scenarios as code snippets.

### HowTo Read/Write PlcOpen values

Reading writing by streams:
Read/Write PlicOpen types (streamed, async)

CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local plc

    ResultHandle resultHandleTime = await client.CreateVariableHandleAsync("MAIN.time", cancel); // TIME
    ResultHandle resultHandleLTime = await client.CreateVariableHandleAsync("MAIN.lTime", cancel); // LTIME
    ResultHandle resultHandleDate = await client.CreateVariableHandleAsync("MAIN.date", cancel); // DATE

    if (resultHandleTime.Succeeded && resultHandleLTime.Succeeded && resultHandleDate.Succeeded)
    {
        try
        {
            byte[] readBuffer = new byte[LTIME.MarshalSize]; // Largest PlicOpen Type is 8 Bytes
            byte[] writeBuffer = new byte[LTIME.MarshalSize];

            // Reading raw value TIME
            await client.ReadAsync(resultHandleTime.Handle, readBuffer.AsMemory(0, TIME.MarshalSize), cancel);
            // Unmarshalling
            TIME plcTime = null;
            PrimitiveTypeMarshaler.Default.Unmarshal(readBuffer.AsSpan(0, TIME.MarshalSize), out plcTime);
            TimeSpan time = plcTime.Time;

            // Writing raw value TIME
            PrimitiveTypeMarshaler.Default.Marshal(time, writeBuffer.AsSpan());
            await client.WriteAsync(resultHandleTime.Handle, writeBuffer.AsMemory(0, TIME.MarshalSize), cancel);

            // Reading raw value LTIME
            await client.ReadAsync(resultHandleLTime.Handle, readBuffer.AsMemory(0, LTIME.MarshalSize), cancel);
            // Unmarshalling
            LTIME plcLTime = null;
            PrimitiveTypeMarshaler.Default.Unmarshal(readBuffer.AsSpan(0, LTIME.MarshalSize), out plcLTime);
            TimeSpan lTime = plcLTime.Time;

            // Writing raw value LTIME
            PrimitiveTypeMarshaler.Default.Marshal(lTime, writeBuffer.AsSpan());
            await client.WriteAsync(resultHandleLTime.Handle, writeBuffer.AsMemory(0, LTIME.MarshalSize), cancel);

            // Reading raw value DATE
            DATE plcDate = null;
            await client.ReadAsync(resultHandleDate.Handle, readBuffer.AsMemory(0, DATE.MarshalSize), cancel);
            // Unmarshalling
            PrimitiveTypeMarshaler.Default.Unmarshal(readBuffer.AsSpan(0, DATE.MarshalSize), out plcDate);
            DateTimeOffset dateTime = plcDate.Date;

            // Writing raw value DATE
            PrimitiveTypeMarshaler.Default.Marshal(plcDate, writeBuffer.AsSpan());
            await client.WriteAsync(resultHandleDate.Handle, writeBuffer.AsMemory(0, DATE.MarshalSize), cancel);
        }
        finally
        {
            ResultAds r1 = await client.DeleteVariableHandleAsync(resultHandleLTime.Handle, cancel);
            ResultAds r2 = await client.DeleteVariableHandleAsync(resultHandleTime.Handle, cancel);
            ResultAds r3 = await client.DeleteVariableHandleAsync(resultHandleDate.Handle, cancel);
        }
    }
}
Read/Write PlcOpen types (streamed, sync)

```csharp
using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local plc

    uint handleTime = 0;
    uint handleLTime = 0;
    uint handleDate = 0;

    try
    {
        handleTime = client.CreateVariableHandle("MAIN.time"); // TIME
        handleLTime = client.CreateVariableHandle("MAIN.lTime"); // LTIME
        handleDate = client.CreateVariableHandle("MAIN.date"); // DATE

        byte[] readBuffer = new byte[LTIME.MarshalSize]; // Largest PlcOpen Type is 8 Bytes
        byte[] writeBuffer = new byte[LTIME.MarshalSize];

        // Reading raw value TIME
        client.Read(handleTime, readBuffer.AsMemory(0, TIME.MarshalSize));
        // Unmarshalling
        TIME plcTime = null;
        PrimitiveTypeMarshaler.Default.Unmarshal(readBuffer.AsSpan(0, TIME.MarshalSize), out plcTime);
        TimeSpan time = plcTime.Time;

        // Writing raw value TIME
        PrimitiveTypeMarshaler.Default.Marshal(time, writeBuffer.AsSpan());
        client.Write(handleTime, writeBuffer.AsMemory(0, TIME.MarshalSize));

        // Reading raw value LTIME
        client.Read(handleLTime, readBuffer.AsMemory(0, LTIME.MarshalSize));
        // Unmarshalling
        LTIME plcLTime = null;
        PrimitiveTypeMarshaler.Default.Unmarshal(readBuffer.AsSpan(0, LTIME.MarshalSize), out plcLTime);
        TimeSpan lTime = plcLTime.Time;

        // Writing raw value LTIME
        PrimitiveTypeMarshaler.Default.Marshal(lTime, writeBuffer.AsSpan());
        client.Write(handleLTime, writeBuffer.AsMemory(0, LTIME.MarshalSize));

        // Reading raw value DATE
        DATE plcDate = null;
        client.Read(handleDate, readBuffer.AsMemory(0, DATE.MarshalSize));
        // Unmarshalling
        PrimitiveTypeMarshaler.Default.Unmarshal(readBuffer.AsSpan(0, DATE.MarshalSize), out plcDate);
        DateTimeOffset dateTime = plcDate.Date;

        // Writing raw value DATE
        PrimitiveTypeMarshaler.Default.Marshal(plcDate, writeBuffer.AsSpan());
        client.Write(handleDate, writeBuffer.AsMemory(0, DATE.MarshalSize));
    }
    finally
    {
        client.DeleteVariableHandle(handleLTime);
        client.DeleteVariableHandle(handleTime);
        client.DeleteVariableHandle(handleDate);
    }
}
```

Reading writing by ANY type concept:

Read/Write PlcOpen types (ANY, async)

```csharp
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local plc

    ResultHandle resultHandleTime = await client.CreateVariableHandleAsync("MAIN.time", cancel); // TIME
    ResultHandle resultHandleDate = await client.CreateVariableHandleAsync("MAIN.date", cancel); // DATE
    ResultHandle resultHandleLTime = await client.CreateVariableHandleAsync("MAIN.lTime", cancel); // LTIME
```
if (resultHandleTime.Succeeded && resultHandleDate.Succeeded && resultHandleLTime.Succeeded)
{
    try
    {
        ResultAnyValue resultTime = await client.ReadAnyAsync(resultHandleTime.Handle, typeof(TIME), cancel); // TIME
        TIME time = (TIME)resultTime.Value;
        TimeSpan timeSpan = time.Time;
        await client.WriteAnyAsync(resultHandleTime.Handle, time, cancel);

        ResultAnyValue resultData = await client.ReadAnyAsync(resultHandleDate.Handle, typeof(DATE), cancel); // DATE
        DATE date = (DATE)resultData.Value;
        DateTimeOffset dateTime = date.Date;
        await client.WriteAnyAsync(resultHandleDate.Handle, date, cancel);

        ResultAnyValue resultLTime = await client.ReadAnyAsync(resultHandleLTime.Handle, typeof(LTIME), cancel); // LTIME
        LTIME lTime = (LTIME)resultLTime.Value;
        TimeSpan lTimeSpan = lTime.Time;
        await client.WriteAnyAsync(resultHandleLTime.Handle, lTime, cancel);
    }
    finally
    {
        ResultAds r1 = await client.DeleteVariableHandleAsync(resultHandleTime.Handle, cancel);
        ResultAds r2 = await client.DeleteVariableHandleAsync(resultHandleDate.Handle, cancel);
        ResultAds r3 = await client.DeleteVariableHandleAsync(resultHandleLTime.Handle, cancel);
    }
}

Read/Write PlcOpen types (ANY, sync)
using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local plc
    uint handleTime = 0;
    uint handleDate = 0;
    uint handleLTime = 0;
    try
    {
        handleTime = client.CreateVariableHandle("MAIN.time"); // TIME
        handleDate = client.CreateVariableHandle("MAIN.date"); // DATE
        handleLTime = client.CreateVariableHandle("MAIN.lTime"); // LTIME

        TIME time = (TIME)client.ReadAny(handleTime, typeof(TIME)); // TIME
        TimeSpan timeSpan = time.Time;
        client.WriteAny(handleTime, time);

        DATE date = (DATE)client.ReadAny(handleDate, typeof(DATE)); // DATE
        DateTimeOffset dateOffset = date.Date;
        client.WriteAny(handleDate, date);

        LTIME lTime = (LTIME)client.ReadAny(handleLTime, typeof(LTIME)); // LTIME
        TimeSpan lTimeSpan = lTime.Time;
        client.WriteAny(handleLTime, lTime);
    }
    finally
    {
        client.DeleteVariableHandle(handleTime);
        client.DeleteVariableHandle(handleDate);
        client.DeleteVariableHandle(handleLTime);
    }
5.4 Event driven read with ADS Notifications

Use of ADS Notifications (Async)

private async Task RegisterNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;
    using (AdsClient client = new AdsClient())
    {
        // Add the Notification event handler
        client.AdsNotification += Client_AdsNotification2;
        // Connect to target
        client.Connect(AmsNetId.Local, 851);
        uint notificationHandle = 0;
        // Notification to a DINT Type (UINT32)
        // Check for change every 200 ms
        byte[] notificationBuffer = new byte[sizeof(UInt32)];
        int size = sizeof(UInt32);
        ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", size, new NotificationSettings(AdsTransMode.OnChange, 200, 0), null, cancel);
        if (result.Succeeded)
        {
            notificationHandle = result.Handle;
            await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
            // Unregister the Event / Handle
            ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
        }
        client.AdsNotification -= Client_AdsNotification2;
    }
}

private void Client_AdsNotification2(object sender, AdsNotificationEventArgs e)
{
    // Or here we know about UDINT type --> can be marshalled as UINT32
    uint nCounter = BinaryPrimitives.ReadUInt32LittleEndian(e.Data.Span);
    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread
    /*SynchronizationContext syncContext = SynchronizationContext.Current;
     _context.Post(status => someLabel.Text = nCounter.ToString(), null); // Non-blocking post */
}

Use of ADS Notifications (Synchronous)

Trigger on changed values by ADS Notifications

private void RegisterNotifications()
{
    using (AdsClient client = new AdsClient())
    {
        // Add the Notification event handler
        client.AdsNotification += Client_AdsNotification;
        // Connect to target
        client.Connect(AmsNetId.Local, 851);
        uint notificationHandle = 0;
        try
        {
            // Notification to a DINT Type (UINT32)
            // Check for change every 200 ms
            int size = sizeof(UInt32);
            byte[] notificationBuffer = new byte[sizeof(UInt32)];
            notificationHandle = client.AddDeviceNotificationAsync("MAIN.nCounter", size, new NotificationSettings(AdsTransMode.OnChange, 200, 0), null, cancel);
        }
    }
5.5 Reactive Read/Write with Reactive Extensions

Observation of Notifications

Notifications (address specified by InstancePath) will be received cyclically as defined in Default.[984] and put into the Observer pipeline for further processing. This example takes 20 Notification samples before returning.

Observe for Notifications

// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Reactive Notification Handler
    var valueObserver = Observer.Create<ushort>(val =>
    {
        Console.WriteLine(string.Format("Value: {0}", val.ToString()));
    });

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values) and subscribe to them.
    IDisposable subscription = client.WhenNotification<ushort>("TwinCAT_SystemInfoVarList._TaskInfo.
    CycleCount", NotificationSettings.Default).Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}

Observation of Symbolic Notifications

This example determines a symbol via Symbolloader and samples its values by Notifications with customized NotificationSettings.[979]. Again 20 samples are taken before the Observation finishes.

Observe for Symbol Notifications

// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
Observation of dynamic Symbol Notifications

Here, the symbol is determined via Symbolloader again, but now the Notifications will be processed as 'dynamic' values.

Observer for dynamic Symbol Notifications

// To Test the Observer run a project on the local PLC System (Port 851) using {AdsClient client = new AdsClient()}
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = (IDynamicSymbolLoader)SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);
    dynamic symbols = symbolLoader.SymbolsDynamic;
    dynamic cycleCount = symbols.TwinCAT_SystemInfoVarList._TaskInfo[1].CycleCount;

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
        // Value objects can be dynamically (on the fly) created objects here (e.g. structs)
        Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToString()));
    });

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransModeOnChange, 500, 5000); // optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.WhenValueChanged().Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}

Polling observer

A polling observer doesn't use ADS notifications, but instead the value read is triggered by a time interval (polling) or a customized trigger function (on request).

Polling observer

// To Test the Observer run a project on the local PLC System (Port 851) using {AdsClient client = new AdsClient()}
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));
// Create Symbol information
var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf o.CycleCount"];  
// Reactive Notification Handler
var valueObserver = Observer.Create<object>(val =>
{
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToString()));
});
// Take 20 Values in an Interval of 500ms
IDisposable subscription = cycleCount.PollValues(TimeSpan.FromMilliseconds(500)).Take(20).Subscr ibe(valueObserver);
Console.ReadKey(); // Wait for Key press
subscription.Dispose(); // Dispose the Subscription

Writing values with observable subject

In this example, a symbolic value is written in a static time interval (1 second). The writing stops after 10 values. The WriteValues [1116] extension method can be used to seamlessly bind value writing into a reactive application.

Writing values with observable subject

using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information (Symbol 'i : INT' in PLC Global Variables list.
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol gvlIntSymbol = (IValueSymbol)symbolLoader.Symbols["GVL.i"];  
    // Produces object (short) Values 0,1,2,3 ... in seconds period
    IObservable<object> timerObservable = Observable.Interval(TimeSpan.FromSeconds(1.0)).Select(i =>
        (object)(short)i);

    // Take 10 Values (0..9) and write them to GVL.i
    IDisposable dispose = gvlIntSymbol.WriteValues(timerObservable.Take(10));

    Console.ReadKey(); // Wait for Key press
    dispose.Dispose(); // Dispose the Subscription
}

5.6 Upgrading existing ADS Application code (Version 4.X --&gt; 5.X)

Code migration path for application code Beckhoff.TwinCAT.Ads 4.X --&gt; 5.X

The Beckhoff.TwinCAT.Ads package Version 4.X implements the Beckhoff ADS API for the Microsoft .NET FullFramework (>= 4.0) / CompactFramework (>= 2.0). Newer .NET framework implementations are basing on '.NET Core' (e.g. .NET 5.0).

To support these newer flavours like .NET Core, .NET Standard together with its new features, a new Version 5.X of the Beckhoff.TwinCAT.Ads API is available.

Because new platform features need some breaking changes within the API for seamless support, the decision was taken to remove some legacy features and burdens in the reimplementation. Nevertheless most of the interfaces remained stable and only a few approaches changed.

The breaking changes and some guidelines will be discussed here.
Use ‘async’ wherever possible.

Internally, the Beckhoff.TwinCAT.Ads package are implemented asynchronously (Methodnames ending with ‘Async’). The synchronous counterparts of the async methods are implemented as thin wrapper around the async method with synchronization.

As consequence the synchronous method is less flexible (only usable in synchronous code paths) and less performant (blocking calls, more thread switches). Furthermore, with the ‘async’ keyword support of .NET they are also not easier to use. So take more than a thought about migrate to async code.

**Synchronous Read value**

```csharp
uint variableHandle = ...;
int value;
AdsErrorCode errorCode = adsClient.TryReadValue<int>(variableHandle, out int value);
```

**Asynchronous**

```csharp
uint variableHandle = ...;
ResultValue<int> result = await adsClient.ReadValueAsync(variableHandle, CancellationToken.None);
AdsErrorCode errorCode = result.ErrorCode;
int value = result.Value;
```

However in synchronous application code paths the await keyword is not allowed. Using the synchronous methods is the only option besides migrating the whole application code up to the main root. Further information can be found here: [Asynchronous programming with async and await](#)

**Name change TcAdsClient --> AdsClient**

This should simply document that there is another implementation. Just rename the class instance.

**Version 4.X**

```csharp
TcAdsClient client = new TcAdsClient()
```

**Version 5.X**

```csharp
AdsClient client = new AdsClient()
```

**Marshalling of byte[] and missing AdsStream class.**

Newer .NET implementations have more efficient possibilities to exchange memory data between software layers and classes (see [Memory- and span-related types](#)).

All API interface methods that use the combination of parameters (bytes, offset, length) or AdsStream

**Version 4.X**

```csharp
public int TcAdsClient.Read(uint indexGroup,uint indexOffset,AdsStream dataStream)
```

**Version 4.X**

```csharp
public int TcAdsClient.Write(uint indexGroup,uint indexOffset,AdsStream dataStream)
```

or

**Version 4.X**

```csharp
public int TcAdsClient.Read(uint indexGroup,uint indexOffset,byte[] bytes, int offset, length)
```

**Version 4.X**

```csharp
public int TcAdsClient.Write(uint indexGroup,uint indexOffset,byte[] bytes, int offset, length)
```

are changed like this:

**Version 5.X**

```csharp
public int AdsClient.Read(uint indexGroup,uint indexOffset,Memory<byte> buffer)
```
Version 5.X
public int AdsClient.Write(uint indexGroup, uint indexOffset, ReadOnlyMemory<byte> buffer)

With this change a lot of interfaces will look much cleaner and act more efficient because of less array copy operations.

Handles are defined as uint type.
Handles are now streamlined consistently as uint type instead of int.

Version 4.X
public int TcAdsClient.CreateVariableHandle(string symbolPath)

Version 5.X
public uint AdsClient.CreateVariableHandle(string symbolPath)

Notification parameters bundled in one settings class.

Version 4.X
public int AddDeviceNotification(string symbolPath, AdsStream dataStream, int offset, int length, AdsTransMode transMode, int cycleTime, int maxDelay, Object userData)

Version 4.X
public int TcAdsClient.AddDeviceNotification(string symbolPath, AdsStream dataStream, int offset, int length, AdsTransMode transMode, int cycleTime, int maxDelay, Object userData)

The new parameterset simply needs the (byte) size of the transferred data instead of AdsStream, offset and length. Furthermore transMode, cycleTime and maxDelay are bundled in the NotificationSettings object.

Version 5.X
public uint AdsClient.AddDeviceNotification(string symbolPath, int dataSize, NotificationSettings settings, object userData)
## 6 TwinCAT.Ads Namespaces

**TwinCAT ADS .NET API for .NET CORE**

### Namespaces

<table>
<thead>
<tr>
<th>Namespace</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT</td>
<td>Common namespace for types that are not specific to ADS.</td>
</tr>
<tr>
<td>TwinCAT.Ads</td>
<td>ADS root namespace.</td>
</tr>
<tr>
<td>TwinCAT.Ads.SumCommand</td>
<td>ADS offers powerful and fast communication to exchange any kind of information. It's possible to read single variables or complete arrays and structures with each one single ADS-API call. ADS Sum-Commands offer to read/write with one single ADS call multiple variables which are not structured within a linear memory, effectively reducing roundtrips.</td>
</tr>
<tr>
<td>TwinCAT.Ads.TcpRouter</td>
<td></td>
</tr>
<tr>
<td>TwinCAT.Ads.TypeSystem</td>
<td>Root namespace for the ADS type system.</td>
</tr>
<tr>
<td>TwinCAT.Ads.ValueAccess</td>
<td>Root namespace for ADS value access.</td>
</tr>
<tr>
<td>TwinCAT.Ams</td>
<td></td>
</tr>
<tr>
<td>TwinCAT.PlcOpen</td>
<td></td>
</tr>
<tr>
<td>TwinCAT.TypeSystem</td>
<td>Namespace for the common (non ADS dependent) type system.</td>
</tr>
<tr>
<td>TwinCAT.TypeSystem.Genric</td>
<td>Namespace for the dynamic part of the common type system.</td>
</tr>
<tr>
<td>TwinCAT.ValueAccess</td>
<td>Namespace for the common (non ADS dependent) value access.</td>
</tr>
</tbody>
</table>

### 6.1 TwinCAT Namespace

Common namespace for types that are not specific to ADS.

#### Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="#">ADSException</a></td>
<td>Base class for all exceptions thrown by the TwinCAT.Ads component</td>
</tr>
<tr>
<td><a href="#">ClientNotConnectedException</a></td>
<td>Class ClientNotConnectedException</td>
</tr>
<tr>
<td><a href="#">ConnectionStateChangedEventArgs</a></td>
<td>Event arguments for the Connection status changed event.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session [101]</td>
<td>Abstract Session base class.</td>
</tr>
<tr>
<td>SessionConnectionStateChangedEventArgs [116]</td>
<td>EventArgs for the ConnectionStatusChanged events.</td>
</tr>
<tr>
<td>SessionException [122]</td>
<td>Session Exception</td>
</tr>
<tr>
<td>SessionNotConnectedException [129]</td>
<td>Class SessionNotConnectedException.</td>
</tr>
<tr>
<td>SessionProvider.TSession, TAddress, TSettings. [134]</td>
<td>Abstract base class for a Custom Session provider</td>
</tr>
<tr>
<td>SymbolLoaderSettings [140]</td>
<td>Settings object for the SymbolLoader initialization.</td>
</tr>
</tbody>
</table>

### Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IConnection [74]</td>
<td>Interface IConnection</td>
</tr>
<tr>
<td>IConnectionStateObserver [80]</td>
<td>Interface for a Connection state watcher (observer)</td>
</tr>
<tr>
<td>IConnectionStateProvider [84]</td>
<td>Interface IConnectionStateProvider</td>
</tr>
<tr>
<td>ISession [88]</td>
<td>Interface ISession</td>
</tr>
<tr>
<td>ISessionProvider [94]</td>
<td>Interface ISessionProvider</td>
</tr>
<tr>
<td>ISessionProvider.TSession, TAddress, TSettings. [97]</td>
<td>Generic ISessionProvider interface</td>
</tr>
<tr>
<td>ISymbolLoaderSettings [99]</td>
<td>Interface ISymbolLoaderSettings</td>
</tr>
<tr>
<td>ISymbolServerProvider [99]</td>
<td>Interface ISymbolServerProvider</td>
</tr>
</tbody>
</table>

### Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionState [67]</td>
<td>Connection state enumeration</td>
</tr>
</tbody>
</table>
## 6.1.1 AdsException Class

Base class for all exceptions thrown by the TwinCAT.Ads component

### Inheritance Hierarchy

- System.Object
  - System.Exception
    - TwinCAT.AdsException

### Namespace

- Namespace: TwinCAT.Ads

### Assembly

- Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
[SerializableAttribute]
public class AdsException : Exception
```

The AdsException type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsException</td>
<td>Initializes a new Instance of the AdsException class.</td>
</tr>
<tr>
<td>AdsException(String)</td>
<td>Initializes a new Instance of the AdsException class.</td>
</tr>
<tr>
<td>AdsException(SerializationInfo, StreamingContext)</td>
<td>Initializes a new instance of the AdsException class.</td>
</tr>
<tr>
<td>AdsException(String, Exception)</td>
<td>Initializes a new Instance of the AdsException class.</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT Namespace [55]
Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException
      TwinCAT.Ads.AdsErrorException
      TwinCAT.Ads.AdsInvalidNotificationException
      TwinCAT.ClientNotConnectedException
      TwinCAT.SessionException
      TwinCAT.TypeSystem.DataTypeException
      TwinCAT.TypeSystem.MarshalException
      TwinCAT.TypeSystem.SymbolException

6.1.1.1 AdsException Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsException()</td>
<td>Initializes a new Instance of the AdsException class.</td>
</tr>
<tr>
<td>AdsException(string)</td>
<td>Initializes a new Instance of the AdsException class.</td>
</tr>
<tr>
<td>AdsException(SerializationInfo, StreamingContext)</td>
<td>Initializes a new instance of the AdsException class.</td>
</tr>
<tr>
<td>AdsException(Exception)</td>
<td>Initializes a new Instance of the AdsException class.</td>
</tr>
</tbody>
</table>

Reference

AdsException Class

TwinCAT Namespace

6.1.1.1 AdsException Constructor

Initializes a new Instance of the AdsException class.

Namespace: TwinCAT
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsException()
```
6.1.1.1.2 AdsException Constructor (String)

Initializes a new Instance of the AdsException class.

**Namespace:** TwinCAT

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public AdsException(string message)
```

**Parameters**

- `message`  
  - *Type:* System.String  
  - A message that describes the error.

**Reference**

AdsException Class [57]

AdsException Overload [59]

TwinCAT Namespace [55]

6.1.1.1.3 AdsException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the AdsException class.

**Namespace:** TwinCAT

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
protected AdsException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

**Parameters**

- `serializationInfo`  
  - *Type:* System.Runtime.Serialization.SerializationInfo  
  - The serialization information.

- `streamingContext`  
  - *Type:* System.Runtime.Serialization.StreamingContext  
  - The streaming context.
6.1.1.1.4 AdsException Constructor (String, Exception)

Initializes a new Instance of the AdsException class.

**Namespace:** TwinCAT

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AdsException(
    string message,
    Exception innerException
)
```

**Parameters**

- **message**
  - Type: `System.String`
  - The error message that explains the reason for the exception.

- **innerException**
  - Type: `System.Exception`
  - The exception that is the cause of the current exception. If the innerException parameter is not a null reference, the current exception is raised in a catch block that handles the inner exception.

**Reference**

- AdsException Class [57]
- AdsException Overload [59]
- TwinCAT Namespace [55]

### AdsException Properties

The AdsException type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Inner Exception</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### AdsException Class [57]

TwinCAT Namespace [55]

#### 6.1.1.3 AdsException Methods

The AdsException [57] type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

#### Reference

AdsException Class [57]

TwinCAT Namespace [55]

#### 6.1.1.4 AdsException Events

The AdsException [57] type exposes the following members.
Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

AdsException Class  [57]

TwinCAT Namespace  [55]

6.1.2 ClientNotConnectedException Class

Class ClientNotConnectedException

Inheritance Hierarchy

System.Object
  
  System.Exception
    
    TwinCAT.AdsException  [57]
      
      TwinCAT.ClientNotConnectedException

Namespace: TwinCAT  [55]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  

[SerializableAttribute]
public class ClientNotConnectedException : AdsException

The ClientNotConnectedException type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Constructor Icon]</td>
<td>ClientNotConnectedException  [65]</td>
</tr>
<tr>
<td>![Constructor Icon]</td>
<td>ClientNotConnectedException(SerializationInfo, StreamingContext)  [65]</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Property Icon]</td>
<td>Data</td>
</tr>
<tr>
<td>![Property Icon]</td>
<td>HelpLink</td>
</tr>
</tbody>
</table>

Payment Method: 1
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HRESULT</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT Namespace [65]

### 6.1.2.1 ClientNotConnectedException Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientNotConnectedException</td>
<td>Initializes a new instance of the ClientNotConnectedException [63] class.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><img src="image.png" alt="Image" /> ClientNotConnectedException[SerialInfo, StreamingContext] [65]</td>
<td>Initializes a new instance of the ClientNotConnectedException [63] class.</td>
</tr>
</tbody>
</table>

**Reference**

ClientNotConnectedException Class [63]

TwinCAT Namespace [55]

### 6.1.2.1.1 ClientNotConnectedException Constructor

Initializes a new instance of the ClientNotConnectedException [63] class.

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public ClientNotConnectedException()
```

**Reference**

ClientNotConnectedException Class [63]

ClientNotConnectedException Overload [64]

TwinCAT Namespace [55]

### 6.1.2.1.2 ClientNotConnectedException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the ClientNotConnectedException [63] class.

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected ClientNotConnectedException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext)
```

**Parameters**

- `serializationInfo`: Type: System.Runtime.Serialization.SerializationInfo
  The serialization information.
streamingContext Type: System.Runtime.Serialization.StreamingContext
The streaming context.

Reference

ClientNotConnectedException Class [63]
ClientNotConnectedException Overload [64]
TwinCAT Namespace [55]

6.1.2.2 ClientNotConnectedException Properties

The ClientNotConnectedException [63] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

ClientNotConnectedException Class [63]
TwinCAT Namespace [55]

6.1.2.3 ClientNotConnectedException Methods

The ClientNotConnectedException [63] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Reference

ClientNotConnectedException Class [63]

TwinCAT Namespace [55]

#### 6.1.2.4 ClientNotConnectedException Events

The ClientNotConnectedException [63] type exposes the following members.

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Reference

ClientNotConnectedException Class [63]

TwinCAT Namespace [55]

#### 6.1.3 ConnectionState Enumeration

Connection state enumeration

### Syntax

C#

```csharp
public enum ConnectionState
```

### Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>Unknown / Uninitialized</td>
</tr>
<tr>
<td>Disconnected</td>
<td>1</td>
<td>Disconnected</td>
</tr>
</tbody>
</table>

Reference

ClientNotConnectedException Class [63]

TwinCAT Namespace [55]
### Member name | Value | Description
--- | --- | ---
Connected | 2 | Connected
Lost | 3 | Connection lost

**Reference**

TwinCAT Namespace [» 55]

### 6.1.4 ConnectionStateChangedEventArgs Class

Event arguments for the Connection status changed event.

**Inheritance Hierarchy**

- System.Object
  - System.EventArgs
    - TwinCAT.ConnectionStateChangedEventArgs
      - TwinCAT.SessionConnectionStateChangedEventArgs [» 116]

**Namespace:** TwinCAT [» 55]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public class ConnectionStateChangedEventArgs : EventArgs
```

The `ConnectionStateChangedEventArgs` type exposes the following members.

**Constructors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ConnectionStateChangedEventArgs(Conn</code>CAR<code>stionStateChangedReason, ConnectionState, ConnectionState)</code></td>
<td>Constructs the <code>ConnectionStateChangedEventArgs</code> arguments.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ConnectionStateChangedEventArgs(Conn</code>CAR<code>stionStateChangedReason, ConnectionState, Exception)</code></td>
<td>Constructs the <code>ConnectionStateChangedEventArgs</code> arguments.</td>
</tr>
</tbody>
</table>

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception</td>
<td>Exception, (only for Error [» 73]</td>
</tr>
</tbody>
</table>
TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NewState [⑦2]</td>
<td>New connection state</td>
</tr>
<tr>
<td>OldState [⑦2]</td>
<td>Old connection state</td>
</tr>
<tr>
<td>Reason [⑦2]</td>
<td>Reason for the event</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT Namespace [⑤5]

6.1.4.1 ConnectionStateChangedEventArgs Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionStateChangedEventArgs(ConnectionStateChangeReason, ConnectionState) [⑦0]</td>
<td>Constructs the ConnectionStateChangedEventArgs [⑥8] arguments.</td>
</tr>
<tr>
<td>ConnectionStateChangedEventArgs(ConnectionStateChangeReason, ConnectionState, Exception) [⑦0]</td>
<td>Constructs the ConnectionStateChangedEventArgs [⑥8] arguments.</td>
</tr>
</tbody>
</table>

Reference

ConnectionStateChangedEventArgs Class [⑥8]

TwinCAT Namespace [⑤5]
6.1.4.1.1 ConnectionStateChangedEventArgs Constructor
(ConnectionStateChangedReason, ConnectionState, ConnectionState)

Constructs the ConnectionStateChangedEventArgs arguments.

Namespace: TwinCAT
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public ConnectionStateChangedEventArgs(
    ConnectionStateChangedReason reason,
    ConnectionState newState,
    ConnectionState oldState
)
```

Parameters

- **reason**
  - Type: TwinCAT.ConnectionStateChangedReason
  - The reason.

- **newState**
  - Type: TwinCAT.ConnectionState
  - The new state.

- **oldState**
  - Type: TwinCAT.ConnectionState
  - The old state.

Reference

ConnectionStateChangedEventArgs Class
ConnectionStateChangedEventArgs Overload
TwinCAT Namespace

6.1.4.1.2 ConnectionChangedEventArgs Constructor
(ConnectionStateChangedReason, ConnectionState, ConnectionState, Exception)

Constructs the ConnectionChangedEventArgs arguments.

Namespace: TwinCAT
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public ConnectionChangedEventArgs(
    ConnectionStateChangedReason reason,
    ConnectionState newState,
    ConnectionState oldState,
    Exception e
)
```

Parameters

- **reason**
  - Type: TwinCAT.ConnectionStateChangedReason
  - The reason.
newState
Type: TwinCAT.ConnectionState
The new state.

oldState
Type: TwinCAT.ConnectionState
The old state.

e
Type: System.Exception
The e.

Reference
ConnectionStateChangedEventArgs Class [68]
ConnectionStateChangedEventArgs Overload [69]
TwinCAT Namespace [55]

6.1.4.2 ConnectionStateChangedEventArgs Properties

The ConnectionStateChangedEventArgs [68] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception</td>
<td>Exception, (only for Error)</td>
</tr>
<tr>
<td>NewState</td>
<td>New connection state</td>
</tr>
<tr>
<td>OldState</td>
<td>Old connection state</td>
</tr>
<tr>
<td>Reason</td>
<td>Reason for the event</td>
</tr>
</tbody>
</table>

Reference
ConnectionStateChangedEventArgs Class [68]
TwinCAT Namespace [55]

6.1.4.2.1 ConnectionStateChangedEventArgs.Exception Property

Exception, (only for Error)

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Exception Exception { get; }
```

Property Value

Type: Exception
6.1.4.2.2 ConnectionStateChangedEventArgs.NewState Property

New connection state

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public ConnectionState NewState { get; }
```

Property Value

Type: ConnectionState [67]

Reference

ConnectionStateChangedEventArgs Class [68]
TwinCAT Namespace [55]

6.1.4.2.3 ConnectionChangedEventArgs.OldState Property

Old connection state

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public ConnectionState OldState { get; }
```

Property Value

Type: ConnectionState [67]

Reference

ConnectionChangedEventArgs Class [68]
TwinCAT Namespace [55]

6.1.4.2.4 ConnectionStateChangedEventArgs.Reason Property

Reason for the event
**Namespace:** TwinCAT [55]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ConnectionStateChangedReason Reason { get; }
```

**Property Value**

Type: `ConnectionStateChangedReason` [73]

**Reference**

`ConnectionStateChangedEventArgs` Class [68]

TwinCAT Namespace [55]

### 6.1.4.3 `ConnectionStateChangedEventArgs` Methods

The `ConnectionStateChangedEventArgs` [68] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td><code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

**Reference**

`ConnectionStateChangedEventArgs` Class [68]

TwinCAT Namespace [55]

### 6.1.5 `ConnectionStateChangedReason` Enumeration

Reason for the Connection status changed event.

**Namespace:** TwinCAT [55]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public enum ConnectionStateChangedReason

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>None / Uninitialized</td>
</tr>
<tr>
<td>Established</td>
<td>1</td>
<td>The connection to the target has been established</td>
</tr>
<tr>
<td>Closed</td>
<td>2</td>
<td>The Connection was closed</td>
</tr>
<tr>
<td>Lost</td>
<td>3</td>
<td>The connection to the target has been lost</td>
</tr>
<tr>
<td>Error</td>
<td>4</td>
<td>Communication error to the target (the connection is not shutting down)</td>
</tr>
<tr>
<td>Resurrected</td>
<td>5</td>
<td>Communication was resurrected (available again)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT Namespace [55]

6.1.6 IConnection Interface

Interface IConnection

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IConnection : IConnectionStateProvider
```

The IConnection type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionState</td>
<td>Gets the current Connection state of the IConnectionStateProvider [84] (Inherited from IConnectionStateProvider [84].)</td>
</tr>
<tr>
<td>DefaultValueEncoding</td>
<td>Gets the default value encoding.</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the Connection Identifier .</td>
</tr>
<tr>
<td>IsConnected</td>
<td>Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available.</td>
</tr>
<tr>
<td>Session</td>
<td>Gets the session that initiated this IConnection</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Close</strong> [78]</td>
<td>Closes this IConnection</td>
</tr>
<tr>
<td><strong>Connect</strong> [78]</td>
<td>(Re)Connects the IConnection when disconnected.</td>
</tr>
<tr>
<td><strong>Disconnect</strong> [79]</td>
<td>Disconnects this IConnection.</td>
</tr>
</tbody>
</table>

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ConnectionStateChanged</strong> [86]</td>
<td>Occurs when connection status of the IConnectionStateProvider [84] has been changed. (Inherited from IConnectionStateProvider [84].)</td>
</tr>
</tbody>
</table>

#### Reference

TwinCAT Namespace [55]

### 6.1.6.1 IConnection Properties

The IConnection [74] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ConnectionState</strong> [85]</td>
<td>Gets the current Connection state of the IConnectionStateProvider [84] (Inherited from IConnectionStateProvider [84].)</td>
</tr>
<tr>
<td><strong>DefaultValueEncoding</strong> [76]</td>
<td>Gets the default value encoding.</td>
</tr>
<tr>
<td><strong>Id</strong> [76]</td>
<td>Gets the Connection Identifier.</td>
</tr>
<tr>
<td><strong>IsConnected</strong> [76]</td>
<td>Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available.</td>
</tr>
<tr>
<td><strong>Session</strong> [77]</td>
<td>Gets the session that initiated this IConnection [74]</td>
</tr>
<tr>
<td><strong>Timeout</strong> [77]</td>
<td>Gets the timeout (in milliseconds)</td>
</tr>
</tbody>
</table>

#### Reference

IConnection Interface [74]

TwinCAT Namespace [55]
6.1.6.1.1 IConnection.DefaultValueEncoding Property

Gets the default value encoding.

**Namespace:**  TwinCAT [55]

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
Encoding DefaultValueEncoding { get; }
```

**Property Value**

Type: Encoding  
The default value encoding.

**Reference**

IConnection Interface [74]  

TwinCAT Namespace [55]

---

6.1.6.1.2 IConnection.Id Property

Gets the Connection Identifier .

**Namespace:**  TwinCAT [55]

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
int Id { get; }
```

**Property Value**

Type: Int32  
The identifier.

**Reference**

IConnection Interface [74]  

TwinCAT Namespace [55]

---

6.1.6.1.3 IConnection.IsConnected Property

Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available.

**Namespace:**  TwinCAT [55]

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
bool IsConnected { get; }
```

Property Value

Type: Boolean

Reference

IConnection Interface [74]
TwinCAT Namespace [55]

6.1.6.1.4 IConnection.Session Property

Gets the session that initiated this IConnection [74]

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
ISession Session { get; }
```

Property Value

Type: ISession [88]
The session or NULL

Remarks

The Session can be null on standalone connections.

Reference

IConnection Interface [74]
TwinCAT Namespace [55]

6.1.6.1.5 IConnection.Timeout Property

Gets the timeout (in milliseconds)

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
int Timeout { get; set; }
```
Property Value

Type: Int32
The timeout.

Reference

IConnection Interface [74]
TwinCAT Namespace [55]

6.1.6.2 IConnection Methods

The IConnection [74] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Closes this IConnection [74]</td>
</tr>
<tr>
<td>Connect</td>
<td>(Re)Connects the IConnection [74] when disconnected.</td>
</tr>
<tr>
<td>Disconnect</td>
<td>Disconnects this IConnection [74].</td>
</tr>
</tbody>
</table>

Reference

IConnection Interface [74]
TwinCAT Namespace [55]

6.1.6.2.1 IConnection.Close Method

Closes this IConnection [74]

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

void Close()

Reference

IConnection Interface [74]
TwinCAT Namespace [55]

6.1.6.2.2 IConnection.Connect Method

(Re)Connects the IConnection [74] when disconnected.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

**C#**

```csharp
bool Connect()
```

**Return Value**

Type: Boolean
true if connected, false otherwise.

**Reference**

[IConnection Interface] [74]
[TwinCAT Namespace] [55]

### 6.1.6.2.3 IConnection.Disconnect Method

Disconnects this [IConnection] [74].

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
bool Disconnect()
```

**Return Value**

Type: Boolean
true if disconnected, false otherwise.

**Reference**

[IConnection Interface] [74]
[TwinCAT Namespace] [55]

### 6.1.6.3 IConnection Events

The [IConnection] [74] type exposes the following members.

**Events**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ConnectionStateChanged] [86]</td>
<td>Occurs when connection status of the [IConnectionStateProvider] [84] has been changed. (Inherited from [IConnectionStateProvider] [84].)</td>
</tr>
</tbody>
</table>

**Reference**

[IConnection Interface] [74]
[TwinCAT Namespace] [55]
6.1.7 IConnectionStateObserver Interface

Interface for a Connection state watcher (observer)

Namespace: TwinCAT\55
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IConnectionStateObserver : IConnectionStateProvider
```

The IConnectionStateObserver type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionState [85]</td>
<td>Gets the current Connection state of the IConnectionStateProvider [84] (Inherited from IConnectionStateProvider [84].)</td>
</tr>
<tr>
<td>ErrorsSinceLastSucceeded [81]</td>
<td>Gets the number of errors since the last successful access</td>
</tr>
<tr>
<td>LastAccess [81]</td>
<td>Gets the DateTimeOffset of the last tried access</td>
</tr>
<tr>
<td>LastSucceededAccess [82]</td>
<td>Gets the DateTimeOffset of the last successful data communication</td>
</tr>
<tr>
<td>Quality [82]</td>
<td>Gets the quality of the current cached value (the age of the data)</td>
</tr>
<tr>
<td>TotalCycles [83]</td>
<td>Gets the number of successful reads / writes</td>
</tr>
<tr>
<td>TotalErrors [83]</td>
<td>Gets the error count of accesses</td>
</tr>
</tbody>
</table>

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionStateChanged [86]</td>
<td>Occurs when connection status of the IConnectionStateProvider [84] has been changed. (Inherited from IConnectionStateProvider [84].)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT Namespace \[55\]

6.1.7.1 IConnectionStateObserver Properties

The IConnectionStateObserver \[80\] type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ConnectionState</code></td>
<td>Gets the current Connection state of the <code>IConnectionStateProvider</code> (Inherited from <code>IConnectionStateProvider</code>.)</td>
</tr>
<tr>
<td><code>ErrorsSinceLastSucceeded</code></td>
<td>Gets the number of errors since the last successful access</td>
</tr>
<tr>
<td><code>LastAccess</code></td>
<td>Gets the <code>DateTimeOffset</code> of the last tried access</td>
</tr>
<tr>
<td><code>LastSucceededAccess</code></td>
<td>Gets the <code>DateTimeOffset</code> of the last successful data communication</td>
</tr>
<tr>
<td><code>Quality</code></td>
<td>Gets the quality of the current cached value (the age of the data)</td>
</tr>
<tr>
<td><code>TotalCycles</code></td>
<td>Gets the number of successful reads / writes</td>
</tr>
<tr>
<td><code>TotalErrors</code></td>
<td>Gets the error count of accesses</td>
</tr>
</tbody>
</table>

### Reference

- `IConnectionStateObserver Interface` ![80]
- `TwinCAT Namespace` ![55]

### 6.1.7.1.1 `IConnectionStateObserver.ErrorsSinceLastSucceeded` Property

Gets the number of errors since the last successful access

**Namespace:** `TwinCAT` ![55]  
**Assembly:** `TwinCAT.Ads.Abstractions` (in `TwinCAT.Ads.Abstractions.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
int ErrorsSinceLastSucceeded { get; }
```

**Property Value**

Type: `Int32`

**Reference**

- `IConnectionStateObserver Interface` ![80]
- `TwinCAT Namespace` ![55]

### 6.1.7.1.2 `IConnectionStateObserver.LastAccess` Property

Gets the `DateTimeOffset` of the last tried access

**Namespace:** `TwinCAT` ![55]  
**Assembly:** `TwinCAT.Ads.Abstractions` (in `TwinCAT.Ads.Abstractions.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
int LastAccess { get; }
```
### TwinCAT.Ads Namespaces

**Namespace:** TwinCAT

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
DateTimeOffset LastAccess { get; }
```

**Property Value**

**Type:** DateTimeOffset

**Reference**

IConnectionStateObserver Interface [80]

TwinCAT Namespace [55]

### 6.1.7.1.3 IConnectionStateObserver.LastSucceededAccess Property

Gets the DateTimeOffset of the last successful data communication

**Namespace:** TwinCAT

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
DateTimeOffset LastSucceededAccess { get; }
```

**Property Value**

**Type:** DateTimeOffset

**Reference**

IConnectionStateObserver Interface [80]

TwinCAT Namespace [55]

### 6.1.7.1.4 IConnectionStateObserver.Quality Property

Gets the quality of the current cached value (the age of the data)

**Namespace:** TwinCAT

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
TimeSpan Quality { get; }
```
6.1.7.1.5  IConnectionStateObserver.TotalCycles Property

Gets the number of successful reads / writes

Namespace:  TwinCAT
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
```csharp
int TotalCycles { get; }
```

6.1.7.1.6  IConnectionStateObserver.TotalErrors Property

Gets the error count of accesses

Namespace:  TwinCAT
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
```csharp
int TotalErrors { get; }
```
6.1.7.2 IConnectionStateObserver Events

The IConnectionStateObserver [80] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionStateChanged [86]</td>
<td>Occurs when connection status of the IConnectionStateProvider [84] has been changed. (Inherited from IConnectionStateProvider [84].)</td>
</tr>
</tbody>
</table>

Reference

IConnectionStateObserver Interface [80]

TwinCAT Namespace [55]

6.1.8 IConnectionStateProvider Interface

Interface IConnectionStateProvider

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IConnectionStateProvider
```

The IConnectionStateProvider type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionState [85]</td>
<td>Gets the current Connection state of the IConnectionStateProvider</td>
</tr>
</tbody>
</table>

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionStateChanged [86]</td>
<td>Occurs when connection status of the IConnectionStateProvider has been changed.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT Namespace [55]

6.1.8.1 IConnectionStateProvider Properties

The IConnectionStateProvider [84] type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionState</td>
<td>Gets the current Connection state of the IConnectionStateProvider [84]</td>
</tr>
</tbody>
</table>

Reference

IConnectionStateProvider Interface [84]

TwinCAT Namespace [55]

6.1.8.1.1 IConnectionStateProvider.ConnectionState Property

Gets the current Connection state of the IConnectionStateProvider [84]

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1e3b60895934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
ConnectionState ConnectionState { get; }
```

Property Value

Type: ConnectionState [67]
The state of the connection.

Remarks

The Connection state changes only if the IConnection [74] is established / shut down or active communication is triggered by the User of the IConnection [74] object.

Examples

The following sample shows how to keep the ConnectionState updated by triggering ADS Communication.

Trigger ConnectionState changes in WPF Applications

```csharp
private DispatcherTimer _timer = null;
private AdsSession _session = null;
//private AdsConnection _connection = null;

private void Window_Loaded(object sender, RoutedEventArgs e)
{
    _session = new AdsSession(AmsNetId.Local, 10000);
    IConnection connection = _session.Connect();
    tbConnectionState.Text = connection.ConnectionState.ToString();
    _session.ConnectionStateChanged += _session_ConnectionStateChanged;

    _timer = new DispatcherTimer();
    _timer.Interval = TimeSpan.FromMilliseconds(200);
    _timer.Tick += TimerOnTick;
    _timer.Start();
}

private void Window_Unloaded(object sender, RoutedEventArgs e)
{
    _timer.Stop();
}
TwinCAT.Ads Namespaces

```csharp
    _session.Dispose();
}

private void _session_ConnectionStateChanged(object sender, TwinCAT.ConnectionStateChangedEventArgs e)
{
    // ConnectionStateChanged will be triggered by communication Invokes
    tbConnectionState.Text = e.NewState.ToString();
}

private void TimerOnTick(object sender, EventArgs eventArgs)
{
    // The Timer Event will occur here in the UI thread because its an DispatcherTimer event!
    // An active ADS request will trigger Connection State periodically!
    StateInfo stateInfo;
    if (_session.Connection.TryReadState(out stateInfo) == AdsErrorCode.NoError)
    {
        tbAdsState.Text = stateInfo.AdsState.ToString();
    }
    else
    {
        tbAdsState.Text = "Invalid";
    }
}
```

**Reference**

[ICconnectionStateProvider Interface][84]

[TwinCAT Namespace][55]

[ICconnectionStateProvider.ConnectionStateChanged][86]

### 6.1.8.2 IConnectionStateProvider Events

The [IConnectionStateProvider][84] type exposes the following members.

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ConnectionStateChanged][86]</td>
<td>Occurs when connection status of the [IConnectionStateProvider][84] has been changed.</td>
</tr>
</tbody>
</table>

**Reference**

[ICConnectionStateProvider Interface][84]

[TwinCAT Namespace][55]

#### 6.1.8.2.1 IConnectionStateProvider.ConnectionStateChanged Event

Occurs when connection status of the [IConnectionStateProvider][84] has been changed.

**Namespace:** TwinCAT.[55]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
    event EventHandler<ConnectionStateChangedEventArgs> ConnectionStateChanged
```
Value

Type: `System.EventHandler<ConnectionStateChangedEventArgs>`

Remarks

The Connection state changes only if the `IConnection` is established / shut down or active communication is triggered by the User of the `IConnection` object.

Examples

The following sample shows how to keep the `ConnectionState` updated by triggering ADS Communication.

**Trigger ConnectionState changes in WPF Applications**

```csharp
private DispatcherTimer _timer = null;
private AdsSession _session = null;
//private AdsConnection _connection = null;
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    _session = new AdsSession(AmsNetId.Local, 10000);
    IConnection connection = _session.Connect();
    tbConnectionState.Text = connection.ConnectionState.ToString();
    _session.ConnectionStateChanged += _session_ConnectionStateChanged;

    _timer = new DispatcherTimer();
    _timer.Interval = TimeSpan.FromMilliseconds(200);
    _timer.Tick += TimerOnTick;
    _timer.Start();
}
private void Window_Unloaded(object sender, RoutedEventArgs e)
{
    _timer.Stop();
    _session.Dispose();
}
private void _session_ConnectionStateChanged(object sender, TwinCAT.ConnectionStateChangedEventArgs e)
{
    // ConnectionStateChanged will be triggered by communication Invokes
    tbConnectionState.Text = e.NewState.ToString();
}
private void TimerOnTick(object sender, EventArgs eventArgs)
{
    // The Timer Event will occur here in the UI thread because its an DispatcherTimer event!
    // An active ADS request will trigger Connection State periodically!
    StateInfo stateInfo;
    if (_session.Connection.TryReadState(out stateInfo) == AdsErrorCode.NoError)
    {
        tbAdsState.Text = stateInfo.AdsState.ToString();
    }
    else
    {
        tbAdsState.Text = "Invalid";
    }
}
```

Reference

- `IConnectionStateProvider Interface`
- `TwinCAT Namespace`
- `IConnectionStateProvider.ConnectionState`
6.1.9  ISession Interface

Interface ISession

Namespace: TwinCAT
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface ISession : IConnectionStateProvider
```

The ISession type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddressSpecifier</td>
<td>Gets the communication endpoint address string representation.</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the Connection object.</td>
</tr>
<tr>
<td>ConnectionState</td>
<td>Gets the current Connection state of the IConnectionStateProvider.</td>
</tr>
<tr>
<td>EstablishedAt</td>
<td>Gets the UTC time when the session was established.</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the Session Id</td>
</tr>
<tr>
<td>IsConnected</td>
<td>Gets a value indicating whether the session is connected.</td>
</tr>
<tr>
<td>Provider</td>
<td>Gets the Session Provider</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Closes this ISession</td>
</tr>
<tr>
<td>Connect</td>
<td>Connects the session and returns the established IConnection object.</td>
</tr>
<tr>
<td>Disconnect</td>
<td>Disconnects the ISession</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionStateChanged</td>
<td>Occurs when connection status of the IConnectionStateProvider has been changed.</td>
</tr>
</tbody>
</table>
6.1.9.1  ISession Properties

The ISession type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddressSpecifier</td>
<td>Gets the communication endpoint address string representation.</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the Connection object.</td>
</tr>
<tr>
<td>ConnectionState</td>
<td>Gets the current Connection state of the IConnectionStateProvider (Inherited from IConnectionStateProvider.)</td>
</tr>
<tr>
<td>EstablishedAt</td>
<td>Gets the UTC time when the session was established.</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the Session Id</td>
</tr>
<tr>
<td>IsConnected</td>
<td>Gets a value indicating whether the session is connected.</td>
</tr>
<tr>
<td>Provider</td>
<td>Gets the Session Provider</td>
</tr>
</tbody>
</table>

Reference

[ISession Interface](#)

TwinCAT Namespace

6.1.9.1.1  ISession.AddressSpecifier Property

Gets the communication endpoint address string representation.

Namespace:  TwinCAT
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
string AddressSpecifier { get; }`

Property Value

Type:  String
The address.

Reference

[ISession Interface](#)

TwinCAT Namespace
6.1.9.1.2  ISession.Connection Property

Gets the Connection object.

**Namespace:** TwinCAT

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
IConnection Connection { get; }
```

**Property Value**

Type: `IConnection`

The connection.

**Remarks**

The `IConnection` object is established by the `ISession` via `Connect` and is valid until the `Disconnect` method or the Dispose method is called. Any possible resurrections after communication losses will be done transparently under the hood of the `IConnection` so that the `IConnection` instance and `ISession` instance.

**Reference**

`ISession Interface`  
`TwinCAT Namespace`

6.1.9.1.3  ISession.EstablishedAt Property

Gets the UTC time when the session was established.

**Namespace:** TwinCAT

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
DateTimeOffset EstablishedAt { get; }
```

**Property Value**

Type: `DateTimeOffset`

The session established at.

**Reference**

`ISession Interface`  
`TwinCAT Namespace`
6.1.9.1.4  ISession.Id Property

Gets the Session Id

**Namespace:** TwinCAT [55]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
int Id { get; }
```

**Property Value**

Type: Int32  
The identifier.

**Reference**

ISession Interface [88]  
TwinCAT Namespace [55]

6.1.9.1.5  ISession.IsConnected Property

Gets a value indicating whether the session is connected.

**Namespace:** TwinCAT [55]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
bool IsConnected { get; }
```

**Property Value**

Type: Boolean  
true if session is connected; otherwise, false.

**Reference**

ISession Interface [88]  
TwinCAT Namespace [55]

6.1.9.1.6  ISession.Provider Property

Gets the Session Provider

**Namespace:** TwinCAT [55]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
ISessionProvider Provider { get; }
```

Property Value

Type: ISessionProvider

The provider.

Reference

ISession Interface

TwinCAT Namespace

6.1.9.2 ISession Methods

The ISession type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Closes this ISession</td>
</tr>
<tr>
<td>Connect</td>
<td>Connects the session and returns the established IConnection object.</td>
</tr>
<tr>
<td>Disconnect</td>
<td>Disconnects the ISession</td>
</tr>
</tbody>
</table>

Reference

ISession Interface

TwinCAT Namespace

6.1.9.2.1 ISession.Close Method

Closes this ISession

Namespace: TwinCAT

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
void Close()
```

Remarks

Closes also the IConnection.

Reference

ISession Interface
### 6.1.9.2.2 ISession.Connect Method

Connects the session and returns the established [Connection](#) object.

**Namespace:** TwinCAT

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
IConnection Connect()
```

#### Return Value

**Type:** [Connection](#)

The [Connection](#) object.

#### Remarks

The [Connection](#) will be valid until the [Session](#) is disconnected via the Disconnect.[#] method or the Dispose method is called. Any possible resurrections after communication losses will be done transparently within the [Connection](#) so that the [Connection](#) instance and [Session](#) instance remains.

**Reference**

[Session Interface](#)

### 6.1.9.2.3 ISession.Disconnect Method

Disconnects the [Session](#)

**Namespace:** TwinCAT

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
bool Disconnect()
```

#### Return Value

**Type:** Boolean

true if Session was disconnected, false if the session was already closed.

#### Remarks

Disposes also the [Connection](#).
6.1.9.3 **ISession Events**

The **ISession** type exposes the following members.

**Events**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionStateChanged</td>
<td>Occurs when connection status of the <strong>IConnectionStateProvider</strong> has been changed. (Inherited from <strong>IConnectionStateProvider</strong>.)</td>
</tr>
</tbody>
</table>

**Reference**

**ISession Interface**

**TwinCAT Namespace**

6.1.10 **ISessionProvider Interface**

Interface **ISessionProvider**

**Namespace:** **TwinCAT**

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public interface ISessionProvider
```

The **ISessionProvider** type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capabilities</td>
<td>Gets the capabilities.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the SessionProvider</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>Creates the Session with address and communication settings.</td>
</tr>
</tbody>
</table>

**Remarks**

Interface for Supporting Session / Communication providers
### 6.1.10.1 ISessionProvider Properties

The `ISessionProvider` type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capabilities</td>
<td>Gets the capabilities.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the SessionProvider</td>
</tr>
</tbody>
</table>

### 6.1.10.1.1 ISessionProvider.Capabilities Property

Gets the capabilities.

**Namespace:** `TwinCAT`  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)  
**Version:** 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
SessionProviderCapabilities Capabilities { get; }
```

#### Property Value

**Type:** `SessionProviderCapabilities`  
The capabilities.

**Reference**

`ISessionProvider Interface [94]`  
`TwinCAT Namespace [55]`

### 6.1.10.1.2 ISessionProvider.Name Property

Gets the name of the SessionProvider

**Namespace:** `TwinCAT`  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)  
**Version:** 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
string Name { get; }
```

Property Value

Type: String
The name.

Reference

[ISessionProvider Interface](#)

[TwinCAT Namespace](#)

6.1.10.2  **ISessionProvider Methods**

The [ISessionProvider](#) type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>Creates the Session with address and communication settings.</td>
</tr>
</tbody>
</table>

Reference

[ISessionProvider Interface](#)

[TwinCAT Namespace](#)

6.1.10.2.1  **ISessionProvider.Create Method**

Creates the Session with address and communication settings.

**Namespace:** TwinCAT

**Assembly:** TwinCAT.Ads.Abstracts (in TwinCAT.Ads.Abstracts.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
ISession Create(
    Object address,
    ISessionSettings settings
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>System.Object</td>
<td>The address.</td>
</tr>
<tr>
<td>settings</td>
<td>ISessionSettings</td>
<td>The settings.</td>
</tr>
</tbody>
</table>
Return Value

Type: ISession
ISession.

Reference

ISessionProvider Interface [94]
TwinCAT Namespace [55]

6.1.11 ISessionProvider.TSession, TAddress, TSettings. Interface

Generic ISessionProvider interface

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public interface ISessionProvider<TSession, TAddress, TSettings> : ISessionProvider
where TSession : ISession

Type Parameters

TSession Session type
TAddress Address specifier
TSettings Communication settings type

The ISessionProvider.TSession, TAddress, TSettings. type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capabilities</td>
<td>Gets the capabilities. (Inherited from ISessionProvider [94].)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the SessionProvider (Inherited from ISessionProvider [94].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create(Object, ISessionSettings)</td>
<td>Creates the Session with address and communication settings. (Inherited from ISessionProvider [94].)</td>
</tr>
<tr>
<td>Create(TAddress, TSettings)</td>
<td>Creates the Session with specified address and communication settings.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT Namespace [55]
TwinCAT.ISessionProvider [94]
### 6.1.11.1 ISessionProvider.TSession, TAddress, TSettings. Properties

The ISessionProvider.TSession, TAddress, TSettings. ❯ 97 generic type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capabilities ❯ 95</td>
<td>Gets the capabilities. (Inherited from ISessionProvider ❯ 94.)</td>
</tr>
<tr>
<td>Name ❯ 95</td>
<td>Gets the name of the SessionProvider (Inherited from ISessionProvider ❯ 94.)</td>
</tr>
</tbody>
</table>

#### Reference

ISessionProvider.TSession, TAddress, TSettings. Interface ❯ 97

TwinCAT Namespace ❯ 55

### 6.1.11.2 ISessionProvider.TSession, TAddress, TSettings. Methods

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create(Object, ISessionSettings) ❯ 96</td>
<td>Creates the Session with address and communication settings. (Inherited from ISessionProvider ❯ 94.)</td>
</tr>
<tr>
<td>Create(TAddress, TSettings) ❯ 99</td>
<td>Creates the Session with specified address and communication settings.</td>
</tr>
</tbody>
</table>

#### Reference

ISessionProvider.TSession, TAddress, TSettings. Interface ❯ 97

TwinCAT Namespace ❯ 55

### 6.1.11.2.1 ISessionProvider.TSession, TAddress, TSettings..Create Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create(Object, ISessionSettings) ❯ 96</td>
<td>Creates the Session with address and communication settings. (Inherited from ISessionProvider ❯ 94.)</td>
</tr>
<tr>
<td>Create(TAddress, TSettings) ❯ 99</td>
<td>Creates the Session with specified address and communication settings.</td>
</tr>
</tbody>
</table>

#### Reference

ISessionProvider.TSession, TAddress, TSettings. Interface ❯ 97

TwinCAT Namespace ❯ 55
I SessionProvider.TSession, TAddress, TSettings..Create Method (TAddress, TSettings)

Creates the Session with specified address and communication settings.

**Namespace:**  TwinCAT [55]

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```
TSession Create(
    TAddress address,
    TSettings settings
)
```

**Parameters**

- **address**  
  Type: TAddress [97]  
  The address.

- **settings**  
  Type: TSettings [97]  
  The communicationSettings.

**Return Value**

Type: TSession [97]

Reference

I SessionProvider.TSession, TAddress, TSettings. Interface [97]

Create Overload [98]

TwinCAT Namespace [55]

6.1.12  ISymbolLoaderSettings Interface

Interface ISymbolLoaderSettings

**Namespace:**  TwinCAT [55]

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```
public interface ISymbolLoaderSettings
```

**Reference**

TwinCAT Namespace [55]

6.1.13  ISymbolServerProvider Interface

Interface ISymbolServerProvider
The ISymbolServerProvider type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolServer</td>
<td>Gets the symbol server.</td>
</tr>
</tbody>
</table>

The ISymbolServerProvider.SymbolServer Property

Gets the symbol server.

Syntax

C#

```csharp
ISymbolServer SymbolServer { get; }
```

Property Value

Type: ISymbolServer

The symbol server. To initially create this information the connection must be established.
6.1.14 Session Class

Abstract Session base class.

Inheritance Hierarchy

System.Object
  TwinCAT.Session
    TwinCAT.Ads.AdsSessionBase [614]

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

public abstract class Session : ISession, IConnectionStateProvider, ISymbolServerProvider, IDisposable

The Session type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddressSpecifier</td>
<td>Gets the communication endpoint address string representation.</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the (established) connection.</td>
</tr>
<tr>
<td>ConnectionState</td>
<td>Gets the current Connection state of the Session</td>
</tr>
<tr>
<td>Disposed</td>
<td>Gets a value indicating whether this Session is disposed.</td>
</tr>
<tr>
<td>EstablishedAt</td>
<td>Gets the UTC time when the session was established.</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the Session Identifier</td>
</tr>
<tr>
<td>IsConnected</td>
<td>Gets a value indicating whether this instance is connected.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the session</td>
</tr>
<tr>
<td>SymbolServer</td>
<td>Gets the symbol server.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Closes this ISession</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

**Name** | **Description**
--- | ---
![Connect](110) | Connects the session.
![Disconnect](111) | Disconnects the session from the target.
![Dispose](112) | Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
![Dispose(Boolean)](112) | Releases unmanaged and - optionally - managed resources.
![Equals](112) | Determines whether the specified object is equal to the current object. (Inherited from `Object`.)
![Finalize](112) | Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from `Object`.)
![GetHashCode](112) | Serves as the default hash function. (Inherited from `Object`.)
![GetSessionName](112) | Gets the name/string identifier of the session.
![GetType](112) | Gets the `Type` of the current instance. (Inherited from `Object`.)
![MemberwiseClone](112) | Creates a shallow copy of the current `Object`. (Inherited from `Object`.)
![OnConnect](113) | Handler function connecting the Session.
![OnCreateSymbolServer](113) | Handler function creating the `ISymbolServer`.
![OnDisconnect](114) | Handler function disconnecting the session.
![OnGetAddress](114) | Handler function getting the address of the session.
![ToString](114) | Returns a string that represents the current object. (Inherited from `Object`.)

### Events

**Name** | **Description**
--- | ---
![ConnectionStateChanged](115) | Occurs when connection status of the `IConnectionStateProvider` has been changed.

### Reference

- TwinCAT Namespace ![55](55)
- TwinCAT.ISession ![88](88)
- `System.IDisposable`

#### 6.1.14.1 Session Properties

The `Session` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddressSpecifier</td>
<td>Gets the communication endpoint address string representation.</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the (established) connection.</td>
</tr>
<tr>
<td>ConnectionState</td>
<td>Gets the current Connection state of the Session.</td>
</tr>
<tr>
<td>Disposed</td>
<td>Gets a value indicating whether this Session is disposed.</td>
</tr>
<tr>
<td>EstablishedAt</td>
<td>Gets the UTC time when the session was established.</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the Session Identifier</td>
</tr>
<tr>
<td>IsConnected</td>
<td>Gets a value indicating whether this instance is connected.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the session</td>
</tr>
<tr>
<td>SymbolServer</td>
<td>Gets the symbol server.</td>
</tr>
</tbody>
</table>

## Reference

- **Session Class** [101]
- **TwinCAT Namespace** [55]

### 6.1.14.1.1 Session.AddressSpecifier Property

Gets the communication endpoint address string representation.

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public string AddressSpecifier { get; }
```

**Property Value**

Type: `String`

The address.

**Implements**

I`Session.AddressSpecifier` [89]

**Reference**

- **Session Class** [101]
TwinCAT Namespace [55]

6.1.14.1.2 Session.Connection Property

Gets the (established) connection.

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public IConnection Connection { get; protected set; }
```

**Property Value**

Type: [IConnection [74]]

The [IConnection [74]] if connection established, or null if not connected.

**Implements**

[ISession.Connection [90]]

**Reference**

Session Class [101]

TwinCAT Namespace [55]

6.1.14.1.3 Session.ConnectionState Property

Gets the current Connection state of the Session [101]

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ConnectionState ConnectionState { get; }
```

**Property Value**

Type: [ConnectionState [67]]

The state of the connection.

**Implements**

[IConnectionStateProvider.ConnectionState [85]]
Remarks

The Connection state changes only if the IConnection is established / shut down or active communication is triggered by the User of the IConnection object.

Examples

The following sample shows how to keep the ConnectionState updated by triggering ADS Communication.

Trigger ConnectionState changes in WPF Applications

```csharp
private DispatcherTimer _timer = null;
private AdsSession _session = null;
private AdsConnection _connection = null;

private void Window_Loaded(object sender, RoutedEventArgs e)
{
    _session = new AdsSession(AmsNetId.Local, 10000);
    IConnection connection = _session.Connect();
    _session.ConnectionStateChanged += _session_ConnectionStateChanged;
    _timer = new DispatcherTimer();
    _timer.Interval = TimeSpan.FromMilliseconds(200);
    _timer.Tick += TimerOnTick;
    _timer.Start();
}

private void Window_Unloaded(object sender, RoutedEventArgs e)
{
    _timer.Stop();
    _session.Dispose();
}

private void _session_ConnectionStateChanged(object sender, TwinCAT.ConnectionStateChangedEventArgs e)
{
    // ConnectionStateChanged will be triggered by communication Invokes
    tbConnectionState.Text = e.NewState.ToString();
}

private void TimerOnTick(object sender, EventArgs eventArgs)
{
    // The Timer Event will occur here in the UI thread because its an DispatcherTimer event!
    // An active ADS request will trigger Connection State periodically!
    StateInfo stateInfo;
    if (_session.Connection.TryReadState(out stateInfo) == AdsErrorCode.NoError)
    {
        tbAdsState.Text = stateInfo.AdsState.ToString();
    }
    else
    {
        tbAdsState.Text = "Invalid";
    }
}
```

Reference

Session Class [101]
TwinCAT Namespace [55]
Session.ConnectionStateChanged [115]

6.1.14.1.4 Session.Disposed Property

Gets a value indicating whether this Session is disposed.
### Session.Disposed Property

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool Disposed { get; }
```

**Property Value**

Type: Boolean
true if disposed; otherwise, false.

**Reference**

Session Class [101]
TwinCAT Namespace [55]

### Session.EstablishedAt Property

Gets the UTC time when the session was established.

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public DateTimeOffset EstablishedAt { get; }
```

**Property Value**

Type: DateTimeOffset
The session established at.

**Implements**

I_Session.EstablishedAt [90]

**Reference**

Session Class [101]
TwinCAT Namespace [55]

### Session.Id Property

Gets the Session Identifier

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#
public int Id { get; }

Property Value
Type: Int32
The identifier.

Implements
[Session.Id] [91]

Reference
Session Class [101]
TwinCAT Namespace [55]

6.1.14.1.7 Session.IsConnected Property
Gets a value indicating whether this instance is connected.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool IsConnected { get; }

Property Value
Type: Boolean
true if this instance is connected; otherwise, false.

Implements
[Session.IsConnected] [91]

Reference
Session Class [101]
TwinCAT Namespace [55]

6.1.14.1.8 Session.Name Property
Gets the name of the session

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
### TwinCAT.Ads Namespaces

#### Syntax

**C#**

```csharp
public string Name { get; }
```

#### Property Value

Type: `String`

The name.

#### Reference

- **Session Class** [▸ 101]
- **TwinCAT Namespace** [▸ 55]

5.6.14.1.9 **Session.SymbolServer Property**

Gets the symbol server.

**Namespace:** TwinCAT [▸ 55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public ISymbolServer SymbolServer { get; }
```

#### Property Value

Type: `ISymbolServer` [▸ 2205]

The symbol server.

**Implements**

- `ISymbolServerProvider.SymbolServer` [▸ 100]

**Remarks**

The `Session` [▸ 101] object holds and caches the symbolic information. To initially create this information, the Connection must be established.

#### Reference

- **Session Class** [▸ 101]
- **TwinCAT Namespace** [▸ 55]

5.6.14.2 **Session Methods**

The `Session` [▸ 101] type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close ![109]</td>
<td>Closes this ISession ![88]</td>
</tr>
<tr>
<td>Connect ![110]</td>
<td>Connects the session.</td>
</tr>
<tr>
<td>Disconnect ![111]</td>
<td>Disconnects the session from the target.</td>
</tr>
<tr>
<td>Dispose ![112]</td>
<td>Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.</td>
</tr>
<tr>
<td>Dispose(Boolean) ![112]</td>
<td>Releases unmanaged and - optionally - managed resources.</td>
</tr>
<tr>
<td>Equals ![112]</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetSessionName ![112]</td>
<td>Gets the name/string identifier of the session.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>OnConnect ![113]</td>
<td>Handler function connecting the Session.</td>
</tr>
<tr>
<td>OnCreateSymbolServer ![113]</td>
<td>Handler function creating the ISymbolServer ![2205]</td>
</tr>
<tr>
<td>OnDisconnect ![114]</td>
<td>Handler function disconnecting the session.</td>
</tr>
<tr>
<td>OnGetAddress ![114]</td>
<td>Handler function getting the address of the session.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

## Reference

Session Class ![101]

TwinCAT Namespace ![55]

### 6.1.14.2.1 Session.Close Method

Closes this ISession ![88]

**Namespace:** TwinCAT ![55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll)

Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void Close()
```
Implements

ISession.Close. [92]

Remarks

Closes also the IConnection. [74].

Reference

Session Class [101]
TwinCAT Namespace [55]

6.1.14.2.2 Session.Connect Method

Connects the session.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public IConnection Connect()

Return Value

Type: IConnection [74]
true if XXXX, false otherwise.

Implements

ISession.Connect. [93]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

The IConnection [74] will be valid until the ISession [88] is disconnected via the Disconnect. [111] method or the Dispose method is called. Any possible resurrections after communication losses will be done transparently within the IConnection [74] so that the IConnection [74] instance and ISession [88] instance remains.

Reference

Session Class [101]
TwinCAT Namespace [55]
6.1.14.2.3 Session.Disconnect Method

Disconnects the session from the target.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public bool Disconnect()

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

ISession.Disconnect. [93]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

Closes (and disposes) the underlying IConnection [74]. The Session [101] itself will not be Disposed and can be reconnected.

Reference

Session Class [101]
TwinCAT Namespace [55]

6.1.14.2.4 Session.Dispose Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.</td>
</tr>
<tr>
<td>Dispose(Boolean)</td>
<td>Releases unmanaged and - optionally - managed resources.</td>
</tr>
</tbody>
</table>

Reference

Session Class [101]
TwinCAT Namespace [55]
Session.Dispose Method

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void Dispose()
```

Implements

IDisposable.Dispose.

Reference

Session Class [101]
Dispose Overload [111]
TwinCAT Namespace [55]

Session.Dispose Method (Boolean)

Releases unmanaged and - optionally - managed resources.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected virtual void Dispose(
    bool disposing
)
```

Parameters

disposing Type: System.Boolean
true to release both managed and unmanaged resources; false to release only unmanaged resources.

Reference

Session Class [101]
Dispose Overload [111]
TwinCAT Namespace [55]

6.1.14.2.5 Session.GetSessionName Method

Gets the name/string identifier of the session.
6.1.14.2.6 Session.OnConnect Method

Handler function connecting the Session.

Syntax

C#
protected virtual IConnection OnConnect(
    bool reconnect
)

Parameters

reconnect Type: System.Boolean
    if set to true [reconnect].

Return Value

Type: IConnection [74]
    IConnection.

Reference

Session Class [101]
TwinCAT Namespace [55]

6.1.14.2.7 Session.OnCreateSymbolServer Method

Handler function creating the ISymbolServer [2205]

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
protected abstract ISymbolServer OnCreateSymbolServer()
```

Return Value

Type: ISymbolServer [2205]

ISymbolServer.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SessionNotConnectedException [129]</td>
<td>The connection is not established!</td>
</tr>
</tbody>
</table>

Reference

Session Class [101]

TwinCAT Namespace [55]

6.1.14.2.8 Session.OnDisconnect Method

Handler function disconnecting the session.

Namespace: TwinCAT [55]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected virtual bool OnDisconnect()
```

Return Value

Type: Boolean
ture if XXXX, false otherwise.

Reference

Session Class [101]

TwinCAT Namespace [55]

6.1.14.2.9 Session.OnGetAddress Method

Handler function getting the address of the session.

Namespace: TwinCAT [55]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
protected abstract string OnGetAddress()
```

Return Value

Type: String
System.String.

Reference

Session Class [101]
TwinCAT Namespace [55]

6.1.14.3 Session Events

The Session [101] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionStateChanged</td>
<td>Occurs when connection status of the IConnectionStateProvider [84] has been changed.</td>
</tr>
</tbody>
</table>

Reference

Session Class [101]
TwinCAT Namespace [55]

6.1.14.3.1 Session.ConnectionStateChanged Event

Occurs when connection status of the IConnectionStateProvider [84] has been changed.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public event EventHandler<ConnectionStateChangedEventArgs> ConnectionStateChanged
```

Value

Type: System.EventHandler<ConnectionStateChangedEventArgs> [68].

Implements

IConnectionStateProvider.ConnectionStateChanged [86]
Remarks

The Connection state changes only if the `IConnection` is established / shut down or active communication is triggered by the User of the `IConnection` object.

Examples

The following sample shows how to keep the `ConnectionState` updated by triggering ADS Communication.

**Trigger ConnectionState changes in WPF Applications**

```csharp
private DispatcherTimer _timer = null;
private AdsSession _session = null;
//private AdsConnection _connection = null;

private void Window_Loaded(object sender, RoutedEventArgs e)
{
    _session = new AdsSession(AmsNetId.Local, 10000);
    IConnection connection = _session.Connect();
    tbConnectionState.Text = connection.ConnectionState.ToString();
    _session.ConnectionStateChanged += _session_ConnectionStateChanged;
    _timer = new DispatcherTimer();
    _timer.Interval = TimeSpan.FromMilliseconds(200);
    _timer.Tick += TimerOnTick;
    _timer.Start();
}

private void Window_Unloaded(object sender, RoutedEventArgs e)
{
    _timer.Stop();
    _session.Dispose();
}

private void _session_ConnectionStateChanged(object sender, TwinCAT.ConnectionStateChangedEventArgs e)
{
    // ConnectionStateChanged will be triggered by communication Invokes
    tbConnectionState.Text = e.NewState.ToString();
}

private void TimerOnTick(object sender, EventArgs eventArgs)
{
    // The Timer Event will occur here in the UI thread because its an DispatcherTimer event!
    // An active ADS request will trigger ConnectionState periodically!
    StateInfo stateInfo;
    if (_session.Connection.TryReadState(out stateInfo) == AdsErrorCode.NoError)
    {
        tbAdsState.Text = stateInfo.AdsState.ToString();
    }
    else
    {
        tbAdsState.Text = "Invalid";
    }
}
```

Reference

- **Session Class**
- **TwinCAT Namespace**
- **Session.ConnectionState**

### 6.1.15 SessionConnectionStateChangedEventArgs Class

EventArguments for the ConnectionStatusChanged events.
Inheritance Hierarchy

System.Object
  TwinCAT.ConnectionStateChangedEventArgs [68]
    TwinCAT.SessionConnectionStateChangedEventArgs

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class SessionConnectionStateChangedEventArgs : ConnectionChangedEventArgs
```

The SessionConnectionStateChangedEventArgs type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="icon.png" alt="Constructor" /></td>
<td>SessionConnectionStateChangedEventArgs(ConnectionStateChangedReason, ConnectionState, ConnectionState, ISession, IConnection) ![119]</td>
</tr>
<tr>
<td><img src="icon.png" alt="Constructor" /></td>
<td>SessionConnectionStateChangedEventArgs(ConnectionStateChangedReason, ConnectionState, ConnectionState, ISession, IConnection, Exception) ![119]</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="icon.png" alt="Property" /></td>
<td>Connection ![121]</td>
</tr>
<tr>
<td><img src="icon.png" alt="Property" /></td>
<td>Exception ![71]</td>
</tr>
<tr>
<td><img src="icon.png" alt="Property" /></td>
<td>NewState ![72]</td>
</tr>
<tr>
<td><img src="icon.png" alt="Property" /></td>
<td>OldState ![72]</td>
</tr>
<tr>
<td><img src="icon.png" alt="Property" /></td>
<td>Reason ![72]</td>
</tr>
<tr>
<td><img src="icon.png" alt="Property" /></td>
<td>Session ![121]</td>
</tr>
</tbody>
</table>
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT Namespace [55]

TwinCAT.ConnectionStateChangedEventArgs [68]

6.1.15.1 SessionConnectionStateChangedEventArgs Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SessionConnectionStateChangedEventArgs(ConnectionStateChangedReason, ConnectionState, ConnectionState, ISession, IConnection)</td>
<td>Initializes a new instance of the SessionConnectionStateChangedEventArgs [116] class.</td>
</tr>
<tr>
<td>SessionConnectionStateChangedEventArgs(ConnectionStateChangedReason, ConnectionState, ConnectionState, ISession, IConnection, Exception)</td>
<td>Initializes a new instance of the SessionConnectionStateChangedEventArgs [116] class.</td>
</tr>
</tbody>
</table>

Reference

SessionConnectionStateChangedEventArgs Class [116]

TwinCAT Namespace [55]
6.1.15.1.1 SessionConnectionStateChangedEventArgs Constructor (ConnectionStateChangedReason, ConnectionState, ConnectionState, ISession, IConnection)

Initializes a new instance of the SessionConnectionStateChangedEventArgs class.

**Namespace:** TwinCAT.[55]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public SessionConnectionStateChangedEventArgs(
    ConnectionStateChangedReason reason,
    ConnectionState newState,
    ConnectionState oldState,
    ISession session,
    IConnection connection
)
```

**Parameters**

- `reason` Type: TwinCAT.ConnectionStateChangedReason
  The reason.
- `newState` Type: TwinCAT.ConnectionState
  The new state.
- `oldState` Type: TwinCAT.ConnectionState
  The old state.
- `session` Type: TwinCAT.ISession
  The session.
- `connection` Type: TwinCAT.IConnection
  The connection.

**Reference**

- SessionConnectionStateChangedEventArgs Class [116]
- SessionConnectionStateChangedEventArgs Overload [118]
- TwinCAT Namespace [55]

6.1.15.1.2 SessionConnectionStateChangedEventArgs Constructor (ConnectionStateChangedReason, ConnectionState, ConnectionState, ISession, IConnection, Exception)

Initializes a new instance of the SessionConnectionStateChangedEventArgs class.

**Namespace:** TwinCAT.[55]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public SessionConnectionStateChangedEventArgs(
    ConnectionStateChangedReason reason,
    ConnectionState newState,
    ConnectionStateChangedReason oldReason
)```
Parameters

reason
Type: `TwinCAT.ConnectionStateChangedReason` [73]
The reason.

newState
Type: `TwinCAT.ConnectionState` [67]
The new state.

oldState
Type: `TwinCAT.ConnectionState` [67]
The old state.

session
Type: `TwinCAT.ISession` [88]
The session.

connection
Type: `TwinCAT.IConnection` [74]
The connection.

e
Type: `System.Exception`
The e.

Reference

SessionConnectionStateChangedEventArgs Class [116]

SessionConnectionStateChangedEventArgs Overload [118]

TwinCAT Namespace [55]

6.1.15.2 SessionConnectionStateChangedEventArgs Properties

The `SessionConnectionStateChangedEventArgs` [116] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>The connection</td>
</tr>
<tr>
<td>Exception</td>
<td>Exception, (only for Error [73]) (Inherited from <code>ConnectionStateChangedEventArgs</code> [68].)</td>
</tr>
<tr>
<td>NewState</td>
<td>New connection state (Inherited from <code>ConnectionStateChangedEventArgs</code> [68].)</td>
</tr>
<tr>
<td>OldState</td>
<td>Old connection state (Inherited from <code>ConnectionStateChangedEventArgs</code> [68].)</td>
</tr>
<tr>
<td>Reason</td>
<td>Reason for the event (Inherited from <code>ConnectionStateChangedEventArgs</code> [68].)</td>
</tr>
<tr>
<td>Session</td>
<td>The session</td>
</tr>
</tbody>
</table>

Reference

SessionConnectionStateChangedEventArgs Class [116]

TwinCAT Namespace [55]
6.1.15.2.1 SessionConnectionStateChangedEventArgs.Connection Property

The connection

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9ab1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public IConnection Connection { get; }
```

**Property Value**

Type: [IConnection][74]

**Reference**

SessionConnectionStateChangedEventArgs Class [116]

TwinCAT Namespace [55]

6.1.15.2.2 SessionConnectionStateChangedEventArgs.Session Property

The session

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9ab1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ISession Session { get; }
```

**Property Value**

Type: [ISession][88]

**Reference**

SessionConnectionStateChangedEventArgs Class [116]

TwinCAT Namespace [55]

6.1.15.3 SessionConnectionStateChangedEventArgs Methods

The SessionConnectionStateChangedEventArgs [116] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

**Reference**

SessionConnectionStateChangedEventArgs Class [116]

TwinCAT Namespace [55]

### 6.1.16 SessionException Class

Session Exception

**Inheritance Hierarchy**

```
System.Object
    \ System.Exception
        \ TwinCAT.AdsException [57]
        \ TwinCAT.SessionException
            \ TwinCAT.SessionNotConnectedException [129]
```

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
[SerializableAttribute]
public class SessionException : AdsException
```

The `SessionException` type exposes the following members.

**Constructors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SessionException(.SerializationInfo, StreamingContext) [124]</td>
<td>Initializes a new instance of the <code>SessionException</code> class.</td>
</tr>
<tr>
<td>SessionException(String, ISession) [125]</td>
<td>Initializes a new instance of the <code>SessionException</code> class.</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SessionException(String, ISession, Exception)</td>
<td>Initializes a new instance of the SessionException class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Session</td>
<td>Gets the session.</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT Namespace [55]
TwinCAT.AdsException [57]

6.1.16.1 SessionException Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![SessionException](SerializationInfo, StreamingContext) ![124]</td>
<td>Initializes a new instance of the SessionException [122] class.</td>
</tr>
<tr>
<td>![SessionException](String, ISession) ![125]</td>
<td>Initializes a new instance of the SessionException [122] class.</td>
</tr>
<tr>
<td>![SessionException](String, ISession, Exception) ![125]</td>
<td>Initializes a new instance of the SessionException [122] class.</td>
</tr>
</tbody>
</table>

Reference

SessionException Class [122]
TwinCAT Namespace [55]

6.1.16.1.1 SessionException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the SessionException [122] class.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

protected SessionException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
Parameters

- `serializationInfo` Type: `System.Runtime.Serialization.SerializationInfo`  
  The serialization information.
  The streaming context.

Reference

- `SessionException Class [122]`
- `SessionException Overload [124]`
- `TwinCAT Namespace [55]`

### 6.1.16.1.2 SessionException Constructor (String, ISession)

Initializes a new instance of the `SessionException [122]` class.

**Namespace:** TwinCAT [55]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public SessionException(
    string message,
    ISession session
)
```

**Parameters**

- `message` Type: `System.String`  
  The message.
- `session` Type: `TwinCAT.ISession [88]`  
  The session.

Reference

- `SessionException Class [122]`
- `SessionException Overload [124]`
- `TwinCAT Namespace [55]`

### 6.1.16.1.3 SessionException Constructor (String, ISession, Exception)

Initializes a new instance of the `SessionException [122]` class.

**Namespace:** TwinCAT [55]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public SessionException(
    string message,
    ISession session,
    Exception innerException
)
```

Parameters

- **message**
  - **Type:** System.String
  - The message.

- **session**
  - **Type:** TwinCAT.ISession
  - The session.

- **innerException**
  - **Type:** System.Exception
  - The inner exception.

Reference

SessionException Class

SessionException Overload

TwinCAT Namespace

6.1.16.2 SessionException Properties

The `SessionException` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Session</td>
<td>Gets the session.</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

SessionException Class
TwinCAT Namespace [55]

6.1.16.2.1  SessionException.Session Property

Gets the session.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public ISession Session { get; }

Property Value

Type: ISession [88]
The session.

Reference

SessionException Class [122]
TwinCAT Namespace [55]

6.1.16.3  SessionException Methods

The SessionException [122] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

SessionException Class [122]
### 6.1.16.3.1 SessionException.GetObjectObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

**Parameters**

- `info` Type: System.Runtime.Serialization.SerializationInfo
  The SerializationInfo that holds the serialized object data about the exception being thrown.

- `context` Type: System.Runtime.Serialization.StreamingContext
  The StreamingContext that contains contextual information about the source or destination.

**Implements**

- ISerializable.GetObjectData(SerializationInfo, StreamingContext)
- Exception.GetObjectObjectData(SerializationInfo, StreamingContext)

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>info</td>
</tr>
</tbody>
</table>

**Reference**

SessionException Class [122]

TwinCAT Namespace [55]

### 6.1.16.4 SessionException Events

The SessionException [122] type exposes the following members.

**Events**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
6.1.17 SessionNotConnectedException Class

Class SessionNotConnectedException.

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException
      TwinCAT.SessionException
        TwinCAT.SessionNotConnectedException

Namespace: TwinCAT
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
[SerializableAttribute]
public class SessionNotConnectedException : SessionException

The SessionNotConnectedException type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SessionNotConnectedException(ISession)</td>
<td>Initializes a new instance of the SessionNotConnectedException class.</td>
</tr>
<tr>
<td>SessionNotConnectedException(SerializationInfo, StreamingContext)</td>
<td>Initializes a new instance of the SessionNotConnectedException class.</td>
</tr>
<tr>
<td>SessionNotConnectedException(String, ISession)</td>
<td>Initializes a new instance of the SessionNotConnectedException class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### TwinCAT.SessionException

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Session [127]</td>
<td>Gets the session. (Inherited from SessionException [122].)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData [128]</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from SessionException [122].)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

#### Reference

- TwinCAT Namespace [55]
- TwinCAT.SessionException [122]
6.1.17.1 SessionNotConnectedException Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="ISession" alt="SessionNotConnectedException" /> ![131]</td>
<td>Initializes a new instance of the SessionNotConnectedException ![129] class.</td>
</tr>
<tr>
<td>![SessionNotConnectedException](SerializationInfo, StreamingContext) ![132]</td>
<td>Initializes a new instance of the SessionNotConnectedException ![129] class.</td>
</tr>
<tr>
<td>![SessionNotConnectedException](String, ISession) ![132]</td>
<td>Initializes a new instance of the SessionNotConnectedException ![129] class.</td>
</tr>
</tbody>
</table>

Reference

SessionNotConnectedException Class ![129]
TwinCAT Namespace ![55]

6.1.17.1.1 SessionNotConnectedException Constructor (ISession)

Initializes a new instance of the SessionNotConnectedException ![129] class.

Namespace: TwinCAT ![55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public SessionNotConnectedException(
    ISession session
)
```

Parameters

- `session` Type: TwinCAT.ISession ![88]
  The session.

Reference

SessionNotConnectedException Class ![129]
SessionNotConnectedException Overload ![131]
TwinCAT Namespace ![55]
6.1.17.1.2  SessionNotConnectedException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the SessionNotConnectedException [129] class.

Namespace:  TwinCAT [55]
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

protected SessionNotConnectedException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)

Parameters

serializationInfo  Type: System.Runtime.Serialization.SerializationInfo
The serialization information.

streamingContext  Type: System.Runtime.Serialization.StreamingContext
The streaming context.

Reference

SessionNotConnectedException Class [129]
SessionNotConnectedException Overload [131]
TwinCAT Namespace [55]

6.1.17.1.3  SessionNotConnectedException Constructor (String, ISession)

Initializes a new instance of the SessionNotConnectedException [129] class.

Namespace:  TwinCAT [55]
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

public SessionNotConnectedException(
    string message,
    ISession session
)

Parameters

message  Type: System.String
The message.

session  Type: TwinCAT.ISession [88]
The session.

Reference

SessionNotConnectedException Class [129]
6.1.17.2 SessionNotConnectedException Properties

The `SessionNotConnectedException` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Session</td>
<td>Gets the session. (Inherited from <code>SessionException</code>).</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

Reference

`SessionNotConnectedException Class` [129]

TwinCAT Namespace [55]

6.1.17.3 SessionNotConnectedException Methods

The `SessionNotConnectedException` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the <code>Exception</code> that is the root cause of one or more subsequent exceptions. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the <code>SerializationInfo</code> with information about the exception. (Inherited from <code>SessionException</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

**Reference**

SessionNotConnectedException Class [129]

TwinCAT Namespace [55]

### 6.1.17.4 SessionNotConnectedException Events

The SessionNotConnectedException [129] type exposes the following members.

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

**Reference**

SessionNotConnectedException Class [129]

TwinCAT Namespace [55]

### 6.1.18 SessionProvider.TSession, TAddress, TSettings. Class

Abstract base class for a Custom Session provider

**Inheritance Hierarchy**

System.Object
  TwinCAT.SessionProvider.TSession, TAddress, TSettings.

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public abstract class SessionProvider<TSession, TAddress, TSettings> : ISessionProvider<TSession, TAddress, TSettings>, ISessionProvider
    where TSession : ISession
    where TAddress : class
```

**Type Parameters**

- TSession: SessionType
- TAddress: Address type
- TSettings: Communication settings type
The SessionProvider.TSession, TAddress, TSettings. type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SessionProvider.TSession, TAddress, TSettings..[†_136]</code></td>
<td>Initializes a new instance of the SessionProvider.TSession, TAddress, TSettings. class.</td>
</tr>
<tr>
<td><code>SessionProvider.TSession, TAddress, TSettings. (SessionProviderCapabilities)[†_136]</code></td>
<td>Initializes a new instance of the SessionProvider.TSession, TAddress, TSettings. class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capabilities[†_137]</td>
<td>Gets the capabilities.</td>
</tr>
<tr>
<td>Name[†_138]</td>
<td>Gets the name of the SessionProvider</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>s_self[†_139]</code></td>
<td>Singleton Instance.</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT Namespace[†_55]

TwinCAT.ISessionProvider.TSession, TAddress, TSettings.[†_97]
6.1.18.1  SessionProvider.TSession, TAddress, TSettings. Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Initializes a new instance of the SessionProvider.TSession, TAddress, TSettings. class.</td>
</tr>
<tr>
<td></td>
<td>Initializes a new instance of the SessionProvider.TSession, TAddress, TSettings. (SessionProviderCapabilities) class.</td>
</tr>
</tbody>
</table>

Reference

SessionProvider.TSession, TAddress, TSettings. Class [134]
TwinCAT Namespace [55]

6.1.18.1.1  SessionProvider.TSession, TAddress, TSettings. Constructor

Initializes a new instance of the SessionProvider.TSession, TAddress, TSettings. class.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected SessionProvider();

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Session provider already instantiated!</td>
</tr>
</tbody>
</table>

Reference

SessionProvider.TSession, TAddress, TSettings. Class [134]
SessionProvider.TSession, TAddress, TSettings. Overload [136]
TwinCAT Namespace [55]

6.1.18.1.2  SessionProvider.TSession, TAddress, TSettings. Constructor (SessionProviderCapabilities)

Initializes a new instance of the SessionProvider.TSession, TAddress, TSettings. class.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

protected SessionProvider(
    SessionProviderCapabilities cap
)

Parameters

cap Type: TwinCAT.SessionProviderCapabilities [140]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception</td>
<td>Session provider already instantiated!</td>
</tr>
</tbody>
</table>

Reference

SessionProvider.TSession, TAddress, TSettings. Class [134]

SessionProvider.TSession, TAddress, TSettings. Overload [136]

TwinCAT Namespace [55]

6.1.18.2 SessionProvider.TSession, TAddress, TSettings. Properties

The SessionProvider.TSession, TAddress, TSettings. [134] generic type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capabilities [137]</td>
<td>Gets the capabilities.</td>
</tr>
<tr>
<td>Name [138]</td>
<td>Gets the name of the SessionProvider</td>
</tr>
</tbody>
</table>

Reference

SessionProvider.TSession, TAddress, TSettings. Class [134]

TwinCAT Namespace [55]

6.1.18.2.1 SessionProvider.TSession, TAddress, TSettings..Capabilities Property

Gets the capabilities.

Namespace: TwinCAT [55]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

public SessionProviderCapabilities Capabilities { get; }
Property Value

Type: SessionProviderCapabilities [140]
The capabilities.

Implements

ISessionProvider.Capabilities [95]

Reference

SessionProvider.TSession, TAddress, TSettings. Class [134]
TwinCAT Namespace [55]

6.1.18.2.2 SessionProvider.TSession, TAddress, TSettings..Name Property

Gets the name of the SessionProvider

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public abstract string Name { get; }

Property Value

Type: String
The name.

Implements

ISessionProvider.Name [95]

Reference

SessionProvider.TSession, TAddress, TSettings. Class [134]
TwinCAT Namespace [55]

6.1.18.3 SessionProvider.TSession, TAddress, TSettings. Methods

The SessionProvider.TSession, TAddress, TSettings. [134] generic type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

#### Reference

SessionProvider.TSession, TAddress, TSettings. Class [› 134]

TwinCAT Namespace [› 55]

#### 6.1.18.4 SessionProvider.TSession, TAddress, TSettings. Fields

The SessionProvider.TSession, TAddress, TSettings. Class [› 134] generic type exposes the following members.

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_self [› 139]</td>
<td>Singleton Instance.</td>
</tr>
</tbody>
</table>

#### Reference

SessionProvider.TSession, TAddress, TSettings. Class [› 134]

TwinCAT Namespace [› 55]

#### 6.1.18.4.1 SessionProvider.TSession, TAddress, TSettings..s_self Field

Singleton Instance.

**Namespace:** TwinCAT [› 55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected static ISessionProvider<TSession, TAddress, TSettings> s_self
```

#### Field Value

**Type:** ISessionProvider [› 97], TSession [› 134], TAddress [› 134], TSettings [› 134].

#### Reference

SessionProvider.TSession, TAddress, TSettings. Class [› 134]

TwinCAT Namespace [› 55]
6.1.19 SessionProviderCapabilities Enumeration

Enum SessionProviderCapabilities

Namespace: TwinCAT

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
[FlagsAttribute]
public enum SessionProviderCapabilities
```

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypeSupport</td>
<td>1</td>
<td>Supports DataTypes</td>
</tr>
<tr>
<td>SymbolBrowsing</td>
<td>2</td>
<td>Supports Symbol Browsing</td>
</tr>
<tr>
<td>ValueRead</td>
<td>4</td>
<td>Supports Value Read</td>
</tr>
<tr>
<td>ValueWrite</td>
<td>8</td>
<td>Support Value Write</td>
</tr>
<tr>
<td>ValueNotifications</td>
<td>16</td>
<td>Supports Value changed Notifications</td>
</tr>
<tr>
<td>None</td>
<td>0</td>
<td>Uninitialized / None</td>
</tr>
<tr>
<td>Mask_All</td>
<td>31</td>
<td>All Capabilities active</td>
</tr>
</tbody>
</table>

Reference

TwinCAT Namespace

6.1.20 SymbolLoaderSettings Class

Settings object for the SymbolLoader initialization.

Inheritance Hierarchy

System.Object
TwinCAT.SymbolLoaderSettings

Namespace: TwinCAT

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class SymbolLoaderSettings : ISymbolLoaderSettings
```

The SymbolLoaderSettings type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolLoaderSettings(SymbolLoadMode)</td>
<td>Initializes a new instance of the SymbolLoaderSettings class with IndexGroupOffsetPreferred.</td>
</tr>
</tbody>
</table>


## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolLoaderSettings(SymbolsLoadMode, ValueAccessMode)</td>
<td>Initializes a new instance of the SymbolLoaderSettings class.</td>
</tr>
<tr>
<td>SymbolLoaderSettings(SymbolsLoadMode, ValueCreationMode, ValueAccessMode)</td>
<td>Initializes a new instance of the SymbolLoaderSettings class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutomaticReconnection</td>
<td>Gets or sets a value indicating whether Disconnect connections can be reconnected.</td>
</tr>
<tr>
<td>Default</td>
<td>Gets the default settings object for standard symbols.</td>
</tr>
<tr>
<td>DefaultDynamic</td>
<td>Gets the default settings object for Dynamic symbols.</td>
</tr>
<tr>
<td>NonCachedArrayElements</td>
<td>Gets or sets the setting to create ArrayElements &quot;On-The-Fly&quot; (Default True)</td>
</tr>
<tr>
<td>SymbolsLoadMode</td>
<td>Gets or sets the symbols load mode.</td>
</tr>
<tr>
<td>ValueAccessMode</td>
<td>Gets or sets the value access mode.</td>
</tr>
<tr>
<td>ValueCreation</td>
<td>Gets or sets the value creation mode.</td>
</tr>
<tr>
<td>ValueUpdateMode</td>
<td>Gets or sets the value update mode.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Remarks**

This settings object is used for the initialization of the internal Symbol loader object.

**Reference**

TwinCAT Namespace [55]

TwinCAT.ISymbolLoaderSettings [99]

TwinCAT.Ads.TypeSystem.SymbolLoaderFactory [1523]

TwinCAT.TypeSystem.ISymbolLoader [2200]

TwinCAT.SymbolsLoadMode [149]


### SymbolLoaderSettings Constructor

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolLoaderSettings(SymbolsLoadMode) [143]</td>
<td>Initializes a new instance of the SymbolLoaderSettings [140] class with IndexGroupOffsetPreferred [1542].</td>
</tr>
<tr>
<td>SymbolLoaderSettings(SymbolsLoadMode, ValueAccessMode) [143]</td>
<td>Initializes a new instance of the SymbolLoaderSettings [140] class.</td>
</tr>
<tr>
<td>SymbolLoaderSettings(SymbolsLoadMode, ValueCreationMode, ValueAccessMode) [144]</td>
<td>Initializes a new instance of the SymbolLoaderSettings [140] class.</td>
</tr>
</tbody>
</table>

**Reference**

SymbolLoaderSettings Class [140]

TwinCAT Namespace [55]
6.1.20.1.1 SymbolLoaderSettings Constructor (SymbolsLoadMode)

Initializes a new instance of the `SymbolLoaderSettings` class with `IndexGroupOffsetPreferred`.

**Namespace:** TwinCAT
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public SymbolLoaderSettings(
    SymbolsLoadMode loadMode
)
```

**Parameters**

- `loadMode` Type: `TwinCAT.SymbolsLoadMode`
  The load mode.

**Reference**

- `SymbolLoaderSettings Class`
- `SymbolLoaderSettings Overload`
- `TwinCAT Namespace`

6.1.20.1.2 SymbolLoaderSettings Constructor (SymbolsLoadMode, ValueAccessMode)

Initializes a new instance of the `SymbolLoaderSettings` class.

**Namespace:** TwinCAT
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public SymbolLoaderSettings(
    SymbolsLoadMode loadMode,
    ValueAccessMode valueAccess
)
```

**Parameters**

- `loadMode` Type: `TwinCAT.SymbolsLoadMode`
  The load mode.

  The value access.

**Reference**

- `SymbolLoaderSettings Class`
- `SymbolLoaderSettings Overload`
- `TwinCAT Namespace`
6.1.20.1.3 SymbolLoaderSettings Constructor (SymbolsLoadMode, ValueCreationModes, ValueAccessMode)

Initializes a new instance of the SymbolLoaderSettings class.

**Namespace:** TwinCAT
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 6.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

C#

```csharp
public SymbolLoaderSettings(
    SymbolsLoadMode loadMode,
    ValueCreationModes valueCreation,
    ValueAccessMode valueAccess
)
```

### Parameters

- **loadMode**
  - Type: TwinCAT.SymbolsLoadMode
  - The load mode.

- **valueCreation**
  - Type: TwinCAT.ValueAccess.ValueCreationModes
  - The dynamic value creation.

- **valueAccess**
  - The value access.

### Reference

- SymbolLoaderSettings Class
- SymbolLoaderSettings Overload
- TwinCAT Namespace

6.1.20.2 SymbolLoaderSettings Properties

The SymbolLoaderSettings class type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="icon.png" alt="AutomaticReconnection" /> ![145]</td>
<td>Gets or sets a value indicating whether Disconnect connections can be reconnected.</td>
</tr>
<tr>
<td><img src="icon.png" alt="Default" /> ![145]</td>
<td>Gets the default settings object for standard symbols.</td>
</tr>
<tr>
<td><img src="icon.png" alt="DefaultDynamic" /> ![146]</td>
<td>Gets the default settings object for Dynamic symbols.</td>
</tr>
<tr>
<td><img src="icon.png" alt="NonCachedArrayElements" /> ![147]</td>
<td>Gets or sets the setting to create ArrayElements &quot;On-The-Fly&quot; (Default True)</td>
</tr>
<tr>
<td><img src="icon.png" alt="SymbolsLoadMode" /> ![147]</td>
<td>Gets or sets the symbols load mode.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>ValueAccessMode</td>
<td>Gets or sets the value access mode.</td>
</tr>
<tr>
<td>ValueCreation</td>
<td>Gets or sets the value creation mode.</td>
</tr>
<tr>
<td>ValueUpdateMode</td>
<td>Gets or sets the value update mode.</td>
</tr>
</tbody>
</table>

Reference

SymbolLoaderSettings Class [140]
TwinCAT Namespace [55]

6.1.20.2.1 SymbolLoaderSettings.AutomaticReconnection Property

Gets or sets a value indicating whether Disconnect connections can be reconnected.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool AutomaticReconnection { get; set; }
```

Property Value

Type: Boolean
true if Disconnect connections can be reconnecte; otherwise, false.

Reference

SymbolLoaderSettings Class [140]
TwinCAT Namespace [55]

6.1.20.2.2 SymbolLoaderSettings.Default Property

Gets the default settings object for standard symbols.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static SymbolLoaderSettings Default { get; }
```

Property Value

Type: SymbolLoaderSettings [140]
The default settings object.
Remarks

The following defaults are set here:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbols load mode (SymbolsLoadMode [147])</td>
<td>Create virtual tree (VirtualTree [149]).</td>
</tr>
<tr>
<td>Value access mode (ValueAccessMode [147])</td>
<td>Prefer Symbolic access of values (Symbolic [1542]).</td>
</tr>
<tr>
<td>Value creation mode ValueCreation [148]</td>
<td>Create .NET integral primitives if possible (Default [2580]).</td>
</tr>
</tbody>
</table>

Reference

SymbolLoaderSettings Class [140]

TwinCAT Namespace [55]

SymbolLoaderSettings.DefaultDynamic [146]

6.1.20.2.3 SymbolLoaderSettings.DefaultDynamic Property

Gets the default settings object for Dynamic symbols.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static SymbolLoaderSettings DefaultDynamic { get; }
```

Property Value

Type: SymbolLoaderSettings [140]
The dynamic default settings object.

Remarks

The following defaults are set here:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbols load mode (SymbolsLoadMode [147])</td>
<td>Create dynamic tree (DynamicTree [149]).</td>
</tr>
<tr>
<td>Value access mode (ValueAccessMode [147])</td>
<td>Prefer Symbolic access of values (Symbolic [1542]).</td>
</tr>
<tr>
<td>Value creation mode ValueCreation [148]</td>
<td>Create .NET integral primitives if possible (Default [2580]).</td>
</tr>
</tbody>
</table>

Reference

SymbolLoaderSettings Class [140]
6.1.20.2.4  SymbolLoaderSettings.NonCachedArrayElements Property

Gets or sets the setting to create ArrayElements "On-The-Fly" (Default True)

**Namespace:**  TwinCAT [55]
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public bool NonCachedArrayElements { get; set; }
```

**Property Value**

Type: Boolean  
The value access mode.

**Reference**

SymbolLoaderSettings Class [140]
TwinCAT Namespace [55]

6.1.20.2.5  SymbolLoaderSettings.SymbolsLoadMode Property

Gets or sets the symbols load mode.

**Namespace:**  TwinCAT [55]
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public SymbolsLoadMode SymbolsLoadMode { get; set; }
```

**Property Value**

Type: SymbolsLoadMode [149]  
The symbols load mode.

**Reference**

SymbolLoaderSettings Class [140]
TwinCAT Namespace [55]

6.1.20.2.6  SymbolLoaderSettings.ValueAccessMode Property

Gets or sets the value access mode.
**TwinCAT.Ads Namespaces**

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ValueAccessMode ValueAccessMode { get; set; }
```

**Property Value**

**Type:** ValueAccessMode [1542]

The value access mode.

**Reference**

SymbolLoaderSettings Class [140]

TwinCAT Namespace [55]

**6.1.20.2.7 SymbolLoaderSettings.ValueCreation Property**

Gets or sets the value creation mode.

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ValueCreationModes ValueCreation { get; set; }
```

**Property Value**

**Type:** ValueCreationModes [2580]

The dynamic value mode.

**Reference**

SymbolLoaderSettings Class [140]

TwinCAT Namespace [55]

**6.1.20.2.8 SymbolLoaderSettings.ValueUpdateMode Property**

Gets or sets the value update mode.

**Namespace:** TwinCAT [55]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ValueUpdateMode ValueUpdateMode { get; set; }
```
Property Value

Type: ValueUpdateMode

The value update mode.

Remarks

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immediately</td>
<td>Writes the values of this DynamicValue instantly when setting its value or</td>
</tr>
<tr>
<td>Triggered</td>
<td>Caches internally the value of this DynamicValue until the Write method</td>
</tr>
<tr>
<td></td>
<td>is called. This reduces ADS rountrips if one or more member/element values</td>
</tr>
<tr>
<td></td>
<td>should be changed. Furthermore the write on the destination system happens</td>
</tr>
<tr>
<td></td>
<td>consistently in one ADS Write operation, which could be important for</td>
</tr>
<tr>
<td></td>
<td>dependent properties/members/elements.</td>
</tr>
</tbody>
</table>

Reference

SymbolLoaderSettings Class

TwinCAT Namespace

6.1.20.3 SymbolLoaderSettings Methods

The SymbolLoaderSettings type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
</tbody>
</table>

Reference

SymbolLoaderSettings Class

TwinCAT Namespace

6.1.21 SymbolsLoadMode Enumeration

Enum SymbolsLoadMode

Namespace: TwinCAT

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
public enum SymbolsLoadMode
```

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat</td>
<td>0</td>
<td>Loads the Symbols organized as Flat List</td>
</tr>
<tr>
<td>VirtualTree</td>
<td>1</td>
<td>Loads the Symbols organized as Virtual tree (Symbol Parent - Child relationships)</td>
</tr>
<tr>
<td>DynamicTree</td>
<td>2</td>
<td>Loads the Symbols as a Virtual tree with Dynamic Symbols (Only available within versions &gt; 4.X of this ADS Api)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT Namespace [55]

6.1.22 ValueUpdateMode Enumeration

Value Update Mode.

Namespace: TwinCAT [55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public enum ValueUpdateMode
```

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>No automatic Value Update / Uninitialized</td>
</tr>
<tr>
<td>Immediately</td>
<td>1</td>
<td>Update Value immediately on property set access.</td>
</tr>
<tr>
<td>Triggered</td>
<td>2</td>
<td>Triggers the ValueUpdate explicitly</td>
</tr>
</tbody>
</table>

Remarks

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immediately</td>
<td>Writes the values of this DynamicValue instantly when setting its value or the value of its child members/elements.</td>
</tr>
<tr>
<td>Triggered</td>
<td>Caches internally the value of this DynamicValue until the DynamicValue.Write method is called. This reduces ADS rountrips, if one or more member/element values should be changed. Furthermore the write on the destination system happens consistently in one ADS Write operation, which could be important for dependent properties/members/elements.</td>
</tr>
</tbody>
</table>
6.2 TwinCAT.Ads Namespace

ADS root namespace.

### Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsClient [154]</td>
<td>ADS Client / ADS Communication object.</td>
</tr>
<tr>
<td>AdsClientSettings [344]</td>
<td>Settings object for the AdsClient [154] class.</td>
</tr>
<tr>
<td>AdsConnection [357]</td>
<td>ADS Connection class</td>
</tr>
<tr>
<td>AdsDataTypeArrayInfo [572]</td>
<td>Array definition for a single dimension.</td>
</tr>
<tr>
<td>AdsErrorException [583]</td>
<td>The exception that is thrown when an ADS error occurs.</td>
</tr>
<tr>
<td>AdsInvalidNotificationException [591]</td>
<td>This AdsInvalidNotificationException is created if the length of the notification data is 0. This indicates that the notification handle is not valid any more. This exception is passed to the AdsNotificationErrorEvent.</td>
</tr>
<tr>
<td>AdsSession [603]</td>
<td>AdsSession class</td>
</tr>
<tr>
<td>Class</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AdsStateChangedEventArgs</td>
<td>Event Arguments for AdsStateChanged events.</td>
</tr>
<tr>
<td>AdsSumCommandException</td>
<td>The exception that is thrown when an ADS SumCommandBase error occurs.</td>
</tr>
<tr>
<td>AdsSymbolVersionChangedEventArgs</td>
<td>Arguments for the AdsSymbolVersionChanged event.</td>
</tr>
<tr>
<td>AdsVersion</td>
<td>The structure contains the version number, revision number and build number.</td>
</tr>
<tr>
<td>AmsAddress</td>
<td>Ams/Ads Address</td>
</tr>
<tr>
<td>AmsNetId</td>
<td>AMS/ADS Net ID</td>
</tr>
<tr>
<td>AmsRouterNotificationEventArgs</td>
<td>Arguments for the IRouterNotificationProvider events.</td>
</tr>
<tr>
<td>DeviceInfo</td>
<td>The structure contains the name and the version information of the device.</td>
</tr>
<tr>
<td>Notification</td>
<td>Class Notification. Implements the INotification interface.</td>
</tr>
<tr>
<td>NotificationSettings</td>
<td>Notification communication settings</td>
</tr>
<tr>
<td>ResultAds</td>
<td>Base class for an (asynchronous) ADS Task Result</td>
</tr>
<tr>
<td>ResultAnyValue</td>
<td>Result object for asynchronous reading an 'AnyValue'/Primitive Value via tasks.</td>
</tr>
<tr>
<td>ResultDeviceInfo</td>
<td>Ads Task Result for DeviceInfo requests (async operations).</td>
</tr>
<tr>
<td>ResultHandle</td>
<td>Result object for asynchronous registering an ADS Handle via tasks.</td>
</tr>
<tr>
<td>ResultRead</td>
<td>Asynchronous ADS Read result.</td>
</tr>
<tr>
<td>ResultReadAdsState</td>
<td>Result object for asynchronous reading AdsStates via tasks.</td>
</tr>
<tr>
<td>ResultReadBytes</td>
<td>ADS Task Result returning Read data for async Read operations.</td>
</tr>
<tr>
<td>ResultReadDeviceState</td>
<td>Result object for asynchronous ADS ReadDeviceState tasks.</td>
</tr>
<tr>
<td>ResultReadWrite</td>
<td>Result object for asynchronous ADS ReadWrite tasks.</td>
</tr>
<tr>
<td>ResultReadWriteBytes</td>
<td>Result object for asynchronous ADS ReadWrite tasks.</td>
</tr>
<tr>
<td>ResultRpcMethod</td>
<td>Class representing a result of an asynchronous RpcMethod call. Implements the ResultAds interface.</td>
</tr>
<tr>
<td>ResultValue.TValue</td>
<td>ADS Result object returning a generic value result (TValue) (asynchronous read). Implements the ResultAds interface.</td>
</tr>
</tbody>
</table>
### Class

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultWrite</td>
<td>Result for asynchronous ADS write tasks.</td>
</tr>
<tr>
<td>SessionSettings</td>
<td>Session settings class</td>
</tr>
<tr>
<td>TaskExtensions</td>
<td>TaskExtensions for Task Cancellation and Timeout</td>
</tr>
<tr>
<td>ValueNotificationEv</td>
<td>Arguments for AdsNotificationEx [868] events.</td>
</tr>
</tbody>
</table>

### Structures

<table>
<thead>
<tr>
<th>Structure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StateInfo</td>
<td>The structure contains the ADS state and device state.</td>
</tr>
</tbody>
</table>

### Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAdsAnyAccess</td>
<td>Interface for accessing ADS 'Any' objects.</td>
</tr>
<tr>
<td>IAdsConnectAddress</td>
<td>Interface for method to connect the ADS client via AMS Address.</td>
</tr>
<tr>
<td>IAdsConnection</td>
<td>ADS Connection interface</td>
</tr>
<tr>
<td>IAdsDisposableConnec</td>
<td>Interface IAdsDisposableConnection implements the IAdsConnectAddress [733] implements the IRouterNotificationProvider [973] implements the IAdsSymbolChangedProvider [934] implements the IDisposable</td>
</tr>
<tr>
<td>IAdsHandle</td>
<td>Interface for ADS access via variable handle</td>
</tr>
<tr>
<td>IAdsNotifications</td>
<td>Interface for Notification management.</td>
</tr>
<tr>
<td>IAdsReadWrite</td>
<td>Interface for ADS Read/Write access via IndexGroup / IndexOffset</td>
</tr>
<tr>
<td>IAdsReadWrite2</td>
<td>Interface for ADS Read/Write access via IndexGroup / IndexOffset</td>
</tr>
<tr>
<td>IAdsReadWriteTimeout</td>
<td>Interface IAdsReadWriteTimeoutAccess</td>
</tr>
<tr>
<td>IAdsRpcInvoke</td>
<td>Interface IAdsRpcInvoke</td>
</tr>
<tr>
<td>IAdsSession</td>
<td>Interface IAdsSession</td>
</tr>
<tr>
<td>IAdsSessionSettings</td>
<td>Interface for ADS Session Settings</td>
</tr>
<tr>
<td>IAdsStateControl</td>
<td>Interface for reading and controlling the ADS state.</td>
</tr>
<tr>
<td>IAdsStateControlTim</td>
<td>Interface IAdsStateControlTimeout</td>
</tr>
</tbody>
</table>

---

TC1000  Version: 1.1  153
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAdsStateObserver</td>
<td>Interface for an AdsState observer</td>
</tr>
<tr>
<td>IAdsStateProvider</td>
<td>Interface IAdsStateProvider</td>
</tr>
<tr>
<td>IAdsSymbolChange</td>
<td>Interface IAdsConnectionLegacy</td>
</tr>
<tr>
<td>IAdsSymbolicAccess</td>
<td>Interface for symbolic ads access.</td>
</tr>
<tr>
<td>IAdsSymbolicTableProvider</td>
<td>Interface IAdsSymbolicTableProvider</td>
</tr>
<tr>
<td>INotification</td>
<td>Common INotification interface</td>
</tr>
<tr>
<td>INotificationSettings</td>
<td>Interface for Notification Settings Implements the IComparable.T.</td>
</tr>
<tr>
<td>IRouterNotificationProvider</td>
<td>Interface for AMS Router Notifications.</td>
</tr>
</tbody>
</table>

### Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsCommandId</td>
<td>AdsCommandId Enumeration</td>
</tr>
<tr>
<td>AdsDataTypeId</td>
<td>ADS data types.</td>
</tr>
<tr>
<td>AdsErrorCode</td>
<td>Describes the ADS error that occurred.</td>
</tr>
<tr>
<td>AdsState</td>
<td>Describes the AdsState.</td>
</tr>
<tr>
<td>AdsTransMode</td>
<td>ADS Transmission Mode for ADS Notifications.</td>
</tr>
<tr>
<td>AmsPort</td>
<td>AmsPorts</td>
</tr>
<tr>
<td>AmsRouterState</td>
<td>State of the AMS Router.</td>
</tr>
<tr>
<td>TransportProtocols</td>
<td>Enum ADS TransportProtocol</td>
</tr>
</tbody>
</table>

### 6.2.1 AdsClient Class

ADS Client / ADS Communication object.

**Inheritance Hierarchy**

System.Object
   TwinCAT.Ads.AdsClient

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

```csharp
public sealed class AdsClient : IAdsDisposableConnection,
    IAdsConnectAddress, IAdsConnection, IConnection, IConnectionServiceProvider, IAdsNotifications,
    IAdsStateProvider, IAdsStateControl, IAdsSymbolChangedProvider, IAdsRpcInvoke, IRouterNotificationProvider,
    IDisposable, IAdsSymbolTableProvider
```

The AdsClient type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address [175]</td>
<td>Gets the target <code>AmsAddress [648]</code> of the established ADS connection (Destination side).</td>
</tr>
<tr>
<td>ClientAddress [176]</td>
<td>Gets the client <code>AmsAddress [648]</code> (Source side).</td>
</tr>
<tr>
<td>DefaultValueEncoding [176]</td>
<td>Gets the default value encoding.</td>
</tr>
<tr>
<td>Id [177]</td>
<td>Gets the AdsClient Identifier.</td>
</tr>
<tr>
<td>IsConnected [177]</td>
<td>Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method <code>ReadState</code> to determine if the target port is available.</td>
</tr>
<tr>
<td>IsDisposed [178]</td>
<td>Gets a value indicating whether this instance is disposed.</td>
</tr>
<tr>
<td>IsLocal [178]</td>
<td>Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.</td>
</tr>
<tr>
<td>Session [179]</td>
<td>Gets the session that initiated this <code>IConnection [74]</code>.</td>
</tr>
<tr>
<td>SymbolEncoding [179]</td>
<td>Gets the symbol encoding.</td>
</tr>
<tr>
<td>Timeout [180]</td>
<td>Sets the timeout for the ads communication. Unit is in ms.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotification(String, Int32, NotificationSettings, Object) [195]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [338] event.</td>
</tr>
<tr>
<td>AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [196]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) ![198]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification ![338] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) ![199]</td>
<td>Adds a device notification as an asynchronous operation.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type) ![201]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx ![340] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) ![202]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx ![340] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) ![204]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx ![340] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) ![205]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) ![206]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx ![340] event.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(UInt32,UInt32,NotificationSettings,Object,Type,.Int32.,CancellationToken) [208]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [340] event.</td>
</tr>
<tr>
<td>CleanupSymbolTable [209]</td>
<td>Clears the internal symbol cache.</td>
</tr>
<tr>
<td>Close [210]</td>
<td>Closes this AdsClient</td>
</tr>
<tr>
<td>Connect(Int32) [211]</td>
<td>Connects to the local target ADS Device.</td>
</tr>
<tr>
<td>Connect(AmsAddress) [211]</td>
<td>Connects the target</td>
</tr>
<tr>
<td>Connect(AmsPort) [212]</td>
<td>Connects to the local target ADS Device.</td>
</tr>
<tr>
<td>Connect(String,Int32) [212]</td>
<td>Connects to the target ADS Device.</td>
</tr>
<tr>
<td>Connect(AmsNetId,Int32) [213]</td>
<td>Connects to the target ADS Device.</td>
</tr>
<tr>
<td>Connect(AmsNetId,AmsPort) [214]</td>
<td>Connects to the target ADS Device.</td>
</tr>
<tr>
<td>ConnectAndWaitAsync [214]</td>
<td>Connects to the target address and waits until the AdsClient is disconnected asynchronously.</td>
</tr>
<tr>
<td>CreateVariableHandle [215]</td>
<td>Determines the Symbol handle by its instance path synchronously.</td>
</tr>
<tr>
<td>CreateVariableHandleAsync [216]</td>
<td>Determines the Symbol handle by its instance path asynchronously.</td>
</tr>
<tr>
<td>DeleteDeviceNotification [216]</td>
<td>Deletes a registered notification.</td>
</tr>
<tr>
<td>DeleteDeviceNotificationAsync [217]</td>
<td>Deletes a registered notification asynchronously.</td>
</tr>
<tr>
<td>DeleteVariableHandle [218]</td>
<td>Releases the specified symbol/variable handle synchronously.</td>
</tr>
<tr>
<td>DeleteVariableHandleAsync [219]</td>
<td>Releases the specified symbol/variable handle asynchronously.</td>
</tr>
<tr>
<td>Disconnect [220]</td>
<td>Disconnects this AdsClient from the local ADS router.</td>
</tr>
<tr>
<td>Dispose [220]</td>
<td>Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Finalizes an instance of the AdsClient class. (Overrides <code>Object.Finalize</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>InvokeRpcMethod</code></td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td><code>InvokeRpcMethod</code></td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td><code>InvokeRpcMethod</code></td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td><code>InvokeRpcMethod</code></td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td><code>InvokeRpcMethod</code></td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td><code>InvokeRpcMethod</code></td>
<td>Invokes the specified RPC Method as an asynchronous operation.</td>
</tr>
<tr>
<td><code>Read</code></td>
<td></td>
</tr>
<tr>
<td><code>Read</code></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The `InvokeRpcMethod` and `Read` methods are provided with different parameter types to accommodate various use cases in communication and data handling within the Ads namespace.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAny(UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, UInt32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, UInt32, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, CancellationToken)</td>
<td>Reads the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, Int32., CancellationToken)</td>
<td>Reads the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, CancellationToken)</td>
<td>Reads the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, Int32., CancellationToken)</td>
<td>Reads the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td>ReadAnyAsync.T.(UInt32, CancellationToken)</td>
<td>Reads data synchronously from an ADS device.</td>
</tr>
<tr>
<td>ReadAnyAsync.T.(UInt32, Int32., CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadAnyAsync_T. (UInt32, UInt32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync_T. (UInt32, UInt32, Int32, CancellationToken)</td>
<td>reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyString(UInt32, Int32, Encoding)</td>
<td>Reads a string from the specified symbol/variable.</td>
</tr>
<tr>
<td>ReadAnyString(UInt32, UInt32, Int32, Encoding)</td>
<td>Reads as string from a specified address.</td>
</tr>
<tr>
<td>ReadAnyStringAsync (UInt32, Int32, Encoding, CancellationToken)</td>
<td>Reads a string asynchronously from the specified symbol/variable.</td>
</tr>
<tr>
<td>ReadAnyStringAsync (UInt32, UInt32, Int32, Encoding, CancellationToken)</td>
<td>reads a string asynchronously from the specified symbol/variable.</td>
</tr>
<tr>
<td>ReadAsync(UInt32, Memory, Void)</td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>ReadAsync(UInt32, UInt32, Memory, Void)</td>
<td>reads data type as an asynchronous operation.</td>
</tr>
<tr>
<td>ReadDataType [P.256]</td>
<td></td>
</tr>
<tr>
<td>ReadDataTypeAsync [P.257]</td>
<td></td>
</tr>
<tr>
<td>ReadDeviceInfo [P.258]</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>ReadDeviceInfoAsync [P.259]</td>
<td></td>
</tr>
<tr>
<td>ReadState [P.260]</td>
<td></td>
</tr>
<tr>
<td>ReadStateAsync [P.260]</td>
<td></td>
</tr>
<tr>
<td>ReadSymbol [P.261]</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadSymbolAsync</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>ReadValue(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>ReadValue(String, Type)</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>ReadValue.T.(String)</td>
<td>Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>ReadValue.T.(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>ReadValueAsync(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously and returns it as an object. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.</td>
</tr>
<tr>
<td>ReadValueAsync(String, Type, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>ReadValueAsync.T.Value.(String, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>ReadValueAsync.T.Value.(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously and returns it as an object.</td>
</tr>
<tr>
<td>ReadWrite(UInt32, Memory, Void)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>ReadWrite(UInt32, UInt32, Memory, Void)</td>
<td>Registers for AdsStateChanged events as an asynchronous operation.</td>
</tr>
<tr>
<td>ReadWriteAsync(UInt32, Memory, Void)</td>
<td></td>
</tr>
<tr>
<td>ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte)</td>
<td></td>
</tr>
<tr>
<td>RegisterAdsStateChangedAsync</td>
<td>Registers for AdsStateChanged events as an asynchronous operation.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>RegisterSymbolVersionChangedAsync</td>
<td>Registers for an AdsSymbolVersionChanged event as an asynchronous operation.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32., UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32., UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>TryCreateVariableHandle</td>
<td>Determines the Symbol handle by its instance path synchronously.</td>
</tr>
<tr>
<td>TryDeleteDeviceNotification(UInt32)</td>
<td>Deletes a registered notification.</td>
</tr>
<tr>
<td>TryDeleteDeviceNotification(UInt32, Int32)</td>
<td>Deletes a registered notification.</td>
</tr>
<tr>
<td>TryDeleteVariableHandle</td>
<td>Releases the specified symbol/variable handle synchronously.</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object., Object.)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object) [288]</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object) [290]</td>
<td>Invokes the rpc method.</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object) [291]</td>
<td>Tries the invoke RPC method.</td>
</tr>
<tr>
<td>TryRead(UInt32, Memory, Void) [293]</td>
<td></td>
</tr>
<tr>
<td>TryRead(UInt32, UInt32, Memory, Void) [294]</td>
<td></td>
</tr>
<tr>
<td>TryReadDataType [294]</td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>TryReadState [295]</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.</td>
</tr>
<tr>
<td>TryReadSymbol [296]</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>TryReadValue(ISymbol, Object) [298]</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>TryReadValue(String , Type, Object) [300]</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
<tr>
<td>TryReadValue&lt;T, (String, T) [297]</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
<tr>
<td>TryReadValue&lt;T, (ISymbol, T) [299]</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, Memory, Void, Byte) [301]</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, UInt32, Memory, Void, Byte)</td>
<td>Resurrects the connection</td>
</tr>
<tr>
<td>TryResurrect</td>
<td></td>
</tr>
<tr>
<td>TryWrite(UInt32, ReadOnlyMemory)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>TryWrite(UInt32, UInt32, ReadOnlyMemory)</td>
<td></td>
</tr>
<tr>
<td>TryWriteControl(StateInfo)</td>
<td></td>
</tr>
<tr>
<td>TryWriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryWriteValue(ISymbol, T)</td>
<td></td>
</tr>
<tr>
<td>UnregisterAdsStateChangedAsync</td>
<td></td>
</tr>
<tr>
<td>UnregisterSymbolVersionChangedAsync</td>
<td></td>
</tr>
<tr>
<td>Write(UInt32, ReadOnlyMemory)</td>
<td></td>
</tr>
<tr>
<td>Write(UInt32, UInt32, ReadOnlyMemory)</td>
<td>Triggers a ‘Write’ call to the ADS device at the specified address.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object, Int32)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAny(UInt32, UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, Int32, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)</td>
<td>Write the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, Int32, CancellationToken)</td>
<td>Write the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td>WriteAnyAsync(String, String, Int32, Encoding, CancellationToken)</td>
<td>Writes the string (Potentially unsafe!)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, String, Int32, Encoding, CancellationToken)</td>
<td>Writes the string (Potentially unsafe!)</td>
</tr>
<tr>
<td>WriteAsync(UInt32, ReadOnlyMemory, Void)</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, CancellationToken)</td>
<td>Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.</td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void)</td>
<td></td>
</tr>
<tr>
<td>WriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>WriteControl(StateInfo, ReadOnlyMemory)</td>
<td></td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, CancellationToken)</td>
<td>Changes the ADS status and device status of the ADS server asynchronously.</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void)</td>
<td></td>
</tr>
<tr>
<td>WriteSymbolAsync</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter</td>
</tr>
<tr>
<td></td>
<td>type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter</td>
</tr>
<tr>
<td></td>
<td>type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue(ISymbol, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32,</td>
</tr>
<tr>
<td></td>
<td>Int32, Bool etc.) are supported. Array and structures are not supported. If</td>
</tr>
<tr>
<td></td>
<td>a string is passed as parameter, the method attempts to parse the string</td>
</tr>
<tr>
<td></td>
<td>according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValue.T.(String, T)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter</td>
</tr>
<tr>
<td></td>
<td>type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue.T.(ISymbol, T)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32,</td>
</tr>
<tr>
<td></td>
<td>Int32, Bool etc.) are supported. Array and structures are not supported. If</td>
</tr>
<tr>
<td></td>
<td>a string is passed as parameter, the method attempts to parse the string</td>
</tr>
<tr>
<td></td>
<td>according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValueAsync(ISymbol, Object, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32,</td>
</tr>
<tr>
<td></td>
<td>Int32, Bool etc.) are supported. Array and structures are not supported. If</td>
</tr>
<tr>
<td></td>
<td>a string is passed as parameter, the method attempts to parse the string</td>
</tr>
<tr>
<td></td>
<td>according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValueAsync.T.(String, T, CancellationToken)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter</td>
</tr>
<tr>
<td></td>
<td>type must have the same layout as the ADS symbol.</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValueAsync&lt;T, Int32, CancellationToken&gt;</td>
<td>Writes a value to the symbol. Strings and all primitive data types (Int32, UInt32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsNotification</td>
<td>Occurs when the ADS device sends a notification to the client.</td>
</tr>
<tr>
<td>AdsNotificationError</td>
<td>Occurs when a exception has occurred during notification management.</td>
</tr>
<tr>
<td>AdsSymbolVersionChanged</td>
<td>Occurs when the symbol version has been changed changes.</td>
</tr>
<tr>
<td>ConnectionStateChanged</td>
<td>Occurs when the connection state has been changed.</td>
</tr>
<tr>
<td>RouterStateChanged</td>
<td>Occurs when the state of the local Router has changed.</td>
</tr>
</tbody>
</table>

### Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollAdsState(TimeSpan)</td>
<td>Overloaded. Gets an observable sequence of AdsState via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollAdsStateAsync(I...</td>
<td>Overloaded. Gets an observable sequence of AdsState via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollAdsStateAsync(TimeSpan, CancellationToken)</td>
<td>Overloaded. Gets an observable sequence of AdsState's via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, IObservable.Unit.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, IObservable.Unit., Func.Exception, Object.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan, Func.Exception, Object.)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, TimeSpan, Func.Exception, Object.)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, IObservable.Unit.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, IObservable.Unit., Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, TimeSpan, Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PollValues.T. (String, Int32, IObservable.Unit.) [1085]</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T. (String, Int32, TimeSpan) [1086]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T. (String, Int32, IObservable.Unit, Func.Exception, T.) [1091]</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T. (String, Int32, TimeSpan, Func.Exception, T.) [1092]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>WhenAdsStateChanges [1066]</td>
<td>Gets an observable sequence of AdsState [626]s. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol) [1068]</td>
<td>Overloaded. Gets an observable sequence of Notification [974]s. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection) [1069]</td>
<td>Overloaded. Gets an observable sequence of Notification [974] objects. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol, NotificationSettings) [1070]</td>
<td>Overloaded. Gets an observable sequence of SymbolValueNotification [1104]s. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection, NotificationSettings) [1071]</td>
<td>Overloaded. Gets an observable sequence of Notification [974] objects. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(String, Type, NotificationSettings) [1099]</td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>WhenNotification.T. (String, NotificationSettings) [1098]</td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhenSymbolVersionChanges(ISchedule)</td>
<td>Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenValueChanged</td>
<td>Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td>WriteValues.T. (String, I.Observable.T.)</td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>WriteValues.T. (String, I.Observable.T., Action.Exception.)</td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
</tbody>
</table>

**Remarks**

The class AdsClient enables synchronous/asynchronous access to data of an ADS Device.

**Examples**

The following sample shows how to instantiate and use the AdsClient class.

**AdsClient Demo (async)**

```csharp
using System;
using System.Buffers.Binary;
using System.Threading;
using System.Threading.Tasks;
using TwinCAT.Ads;
using TwinCAT.TypeSystem;

namespace Sample
{
    class ClientAsync
    {
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static async void Main(string[] args)
        {
            AmsAddress address = ArgParser.Parse(args);
            CancellationTokenSource cancelSource = new CancellationTokenSource();
            CancellationToken cancel = cancelSource.Token;

            using (AdsClient client = new AdsClient())
            {
                // Connect to Address
                client.Connect(address.NetId, address.Port); // Connect to Port (851, first PLC by default)

                // Read the identification and version number of the device
                ResultDeviceInfo resultDeviceInfo = await client.ReadDeviceInfoAsync(cancel);
                if (resultDeviceInfo.Succeeded)
                {
                    DeviceInfo deviceInfo = resultDeviceInfo.DeviceInfo;
                    Console.WriteLine(string.Format("DeviceName: {0}", deviceInfo.Name));
                    Console.WriteLine(string.Format("DeviceVersion: {0}", version.ToString(3)));
                }

                // Read the state of the device
                ResultReadDeviceState resultReadDeviceState = await client.ReadStateAsync(cancel);
            }
        }
    }
}
```
TwinCAT.Ads Namespaces

AdsState state = AdsState.Invalid;

if (resultReadDeviceState.Succeeded)
{
    StateInfo stateInfo = resultReadDeviceState.State;
    state = stateInfo.AdsState;
    short deviceState = stateInfo.DeviceState;
    Console.WriteLine(string.Format("DeviceState: {0}", deviceState));
    Console.WriteLine(string.Format("AdsState : {0}", state));
}

// Write ADS Commands (write state) to target
// Set PLC to Run
if (state == AdsState.Stop)
{
    await client.WriteControlAsync(AdsState.Run, 0, cancel);
}

//create variable handle for Plc Project Name (automatic generated symbol in PLC)
ResultHandle resultHandle = await client.CreateVariableHandleAsync("TwinCAT_SystemInfoVarList._AppInfo.ProjectName", cancel);

if (resultHandle.Succeeded)
{
    uint handleProjectName = resultHandle.Handle;
    uint handleNotification = 0; // Notification Handle for Task1 CycleCount changes
    try
    {
        // Read value from target and Marshal data into string
        byte[] readData = new byte[256];
        ResultRead resultRead = await client.ReadAsync(handleProjectName, readData.AsMemory(), cancel);

        if (resultRead.Succeeded)
        {
            PrimitiveTypeMarshaler marshaler = PrimitiveTypeMarshaler.Default;
            string projectName = null;
            int unmarshaledBytes = marshaler.Unmarshal(readData, out projectName);
            Console.WriteLine(string.Format("ProjectName : {0}", projectName));
        }
    }
    finally
    {
        // Cleanup all handles
        // Dispose all Streams
        ResultAds result = await client.DeleteDeviceNotificationAsync(handleNotification, cancel);
        // Always delete all variable handles.
        result = await client.DeleteVariableHandleAsync(handleProjectName, cancel);
    }
}

Console.WriteLine("\n");
Console.WriteLine("Press [Enter] for leave:");
Console.ReadLine();
using System;
using System.Buffers.Binary;
using System.Threading;
using System.Threading.Tasks;
using TwinCAT.Ads;
using TwinCAT.TypeSystem;

namespace Sample
{
    class Client
    {
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static void Main(string[] args)
        {
            AmsAddress address = ArgParser.Parse(args);
            using (AdsClient client = new AdsClient())
            {
                // Connect to Address
                client.Connect(address.NetId, address.Port); // Connect to Port (851, first PLC by default)

                // Read the identification and version number of the device
                DeviceInfo deviceInfo = client.ReadDeviceInfo();
                Console.WriteLine(string.Format("DeviceName: {0}" , deviceInfo.Name));
                Console.WriteLine(string.Format("DeviceVersion: {0}" , version.ToString(3)));

                // Read the state of the device
                StateInfo stateInfo = client.ReadState();
                AdsState state = AdsState.Invalid;
                state = stateInfo.AdsState;
                short deviceState = stateInfo.DeviceState;
                Console.WriteLine(string.Format("DeviceState: {0}" , deviceState));
                Console.WriteLine(string.Format("AdsState : {0}" , state));

                // Write ADS Commands (write state) to target
                // Set PLC to Run
                if (state == AdsState.Stop)
                {
                    client.WriteControl(new StateInfo(AdsState.Run, 0));
                }

                //create variable handle for Plc Project Name (automatic generated symbol in PLC)
                uint handleProjectName = client.CreateVariableHandle("TwinCAT_SystemInfoVarList._AppInfo.ProjectName");

                uint handleNotification = 0; // Notification Handle for Task1 CycleCount changes
                try
                {
                    // Read value from target and Marshal data into string
                    byte[] readData = new byte[256];
                    int readBytes = client.Read(handleProjectName, readData.AsMemory());
                    PrimitiveTypeMarshaler marasher = PrimitiveTypeMarshaler.Default;
                    string projectName = null;
                    int unmarshaledBytes = marasher.Unmarshal(readData, client.DefaultValueEncoding, out projectName);
                    Console.WriteLine(string.Format("ProjectName : {0}" , projectName));

                    //notificationBuffer = new byte[4]; // Sizeof UDINT
                    int size = sizeof(UInt32); // Sizeof UDINT
                    client.AdsNotification += client_NotificationEvent; // Register for Notification event
                    //Register Notification for Task1 CycleCount symbol (automatic generated symbol in PLC)
                    handleNotification = client.AddDeviceNotification("TwinCAT_SystemInfoVarList._TaskInfo[1].CycleCount", size, NotificationSettings.Default, null);
                    // Sleep 10 Seconds to receive events
                }
The following sample shows how to call (Remote Procedures / Methods) within the PLC directly from the 
AdsClient class.

**RPC Call Example (async)**

```csharp
namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT.Ads;
    using TwinCAT.TypeSystem;
    class RpcCallAsync
    {
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static async void Main(string[] args)
        {
            // Parse the AmsAddress from command-line arguments
            AmsAddress address = ArgParser.Parse(args);

            // Create the ADS Client
            using (AdsClient client = new AdsClient())
            {
                // Establish Connection
                client.Connect(address);

                // Call a Method that has the following signature (within MAIN Program)
                /*  {attribute 'TcRpcEnable'}
                METHOD PUBLIC M_Add : INT
                VAR_INPUT */
            }
        }
    }
}
```

**Argument Parser**

```csharp
public static class ArgParser
{
    /// <summary>
    /// Parses the arguments.
    /// </summary>
    /// <param name="args">The arguments.</param>
    /// <returns>AmsAddress.</returns>
    public static AmsAddress Parse(string[] args)
    {
        AmsNetId netId = AmsNetId.Local;
        int port = 851;
        if (args != null)
        {
            if (args.Length > 0 && args[0] != null)
                netId = AmsNetId.Parse(args[0]);
            if (args.Length > 1 && args[1] != null)
                port = int.Parse(args[1]);
        }
        return new AmsAddress(netId, port);
    }
}
```
**RPC Call Example (sync)**

```csharp
namespace Sample
{
    using System;
    using System.Diagnostics;
    using TwinCAT.Ads;
    using TwinCAT.TypeSystem;
    class RpcCall
    {
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static void Main(string[] args)
        {
            //Parse the AmsAddress from command-line arguments
            AmsAddress address = ArgParser.Parse(args);
            // Create the ADS Client
            using (AdsClient client = new AdsClient())
            {
                // Establish Connection
                client.Connect(address);
                // Call a Method that has the following signature (within MAIN Program)
                /* {attribute 'TcRpcEnable'}
                METHOD PUBLIC M_Add : INT
                VAR_INPUT
                i1 : INT := 0;
                i2 : INT := 0;
                END_VAR
                */
                short result = (short)client.InvokeRpcMethod("MAIN", "M_Add", new object[] { (short)1, (short)4 }, CancellationToken.None);
                // Call a Method that has no parameter and returns VOID
                client.InvokeRpcMethod("MAIN", "M_Method1", new object[] { }, CancellationToken.None);
            }
        }
    }
}
```

**Reference**

_TwinCAT.Ads Namespace_ [151]

### 6.2.1.1 AdsClient Properties

The AdsClient [154] type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Gets the target \texttt{AmsAddress} of the established ADS connection.</td>
</tr>
<tr>
<td>ClientAddress</td>
<td>Gets the client \texttt{AmsAddress} (Source side).</td>
</tr>
<tr>
<td>DefaultValueEncoding</td>
<td>Gets the default value encoding.</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the identifier.</td>
</tr>
<tr>
<td>IsConnected</td>
<td>Gets a value indicating whether the local ADS port opened successfully.</td>
</tr>
<tr>
<td>IsDisposed</td>
<td>Gets a value indicating whether this instance is disposed.</td>
</tr>
<tr>
<td>IsLocal</td>
<td>Gets a value indicating whether the ADS client is connected to an ADS Server.</td>
</tr>
<tr>
<td>Session</td>
<td>Gets the session that initiated this \texttt{IConnection}.</td>
</tr>
<tr>
<td>SymbolEncoding</td>
<td>Gets the symbol encoding.</td>
</tr>
<tr>
<td>Timeout</td>
<td>Sets the timeout for the ads communication. Unit is in ms.</td>
</tr>
</tbody>
</table>

## Reference

\texttt{AdsClient Class} \[154\]

\texttt{TwinCAT.Ads Namespace} \[151\]

### 6.2.1.1 AdsClient.Address Property

Gets the target \texttt{AmsAddress} of the established ADS connection (Destination side).

**Namespace:** TwinCAT.Ads \[151\]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AmsAddress Address { get; }
```

**Property Value**

Type: \texttt{AmsAddress} \[648\]

The address.

**Implements**

\texttt{IAdsConnection.Address} \[781\]
6.2.1.1.2 AdsClient.ClientAddress Property

Get the client AmsAddress (Source side).

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public AmsAddress ClientAddress { get; }

Property Value

Type: AmsAddress
The client address.

Implements

IAdsConnection.ClientAddress

Reference

AdsClient Class
TwinCAT.Ads Namespace

6.2.1.1.3 AdsClient.DefaultValueEncoding Property

Gets the default value encoding.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public Encoding DefaultValueEncoding { get; }

Property Value

Type: Encoding
The default value encoding.

Implements

IConnection.DefaultValueEncoding

Reference

AdsClient Class
TwinCAT.Ads Namespace
6.2.1.1.4 AdsClient.Id Property

Gets the AdsClient Identifier.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int Id { get; }
```

**Property Value**

Type: Int32
The identifier.

**Implements**

IConnection.Id [76]

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]

6.2.1.1.5 AdsClient.IsConnected Property

Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool IsConnected { get; }
```

**Property Value**

Type: Boolean
true if this instance is connected; otherwise, false.
Implementes

IConnection.IsConnected [76]

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]

6.2.1.1.6 AdsClient.IsDisposed Property

Gets a value indicating whether this instance is disposed.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdaca3e72bc0ea15da1c14

Syntax

C#

public bool IsDisposed { get; }

Property Value

Type: Boolean
ture if this instance is disposed; otherwise, false.

Implementes

IAdsDisposableConnection.IsDisposed [812]

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]

6.2.1.1.7 AdsClient.IsLocal Property

Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdaca3e72bc0ea15da1c14

Syntax

C#

public bool IsLocal { get; }

Property Value

Type: Boolean
ture if this instance is local; otherwise, false.
Implements

IAdsConnection.IsLocal [782]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

6.2.1.1.8 AdsClient.Session Property

Gets the session that initiated this IConnection [74]

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public ISession Session { get; }

Property Value

Type: ISession [88]
The session or NULL

Implements

IConnection.Session [77]

Remarks

The Session can be null on standalone connections.

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

6.2.1.1.9 AdsClient.SymbolEncoding Property

Gets the symbol encoding.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
public Encoding SymbolEncoding { get; }
```

Property Value

Type: Encoding
The symbol encoding.

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]

6.2.1.10 AdsClient.Timeout Property

Sets the timeout for the ads communication. Unit is in ms.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int Timeout { get; set; }
```

Property Value

Type: Int32

Implements

IConnection.Timeout [77]

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]

6.2.1.2 AdsClient Methods

The AdsClient [154] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![icon] AddDeviceNotification(String, Int32, NotificationSettings, Object) [195]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [338] event.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Adds a device notification as an asynchronous operation.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, Int32, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [340] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, Int32, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [340] event.</td>
</tr>
<tr>
<td>CleanupSymbolTable [209]</td>
<td>Clears the internal symbol cache.</td>
</tr>
<tr>
<td>Close [210]</td>
<td>Closes this AdsClient [154]</td>
</tr>
<tr>
<td>Connect(Int32) [211]</td>
<td>Connects to the local target ADS Device.</td>
</tr>
<tr>
<td>Connect(AmsAddress) [211]</td>
<td>Connects the target</td>
</tr>
<tr>
<td>Connect(AmsPort) [212]</td>
<td>Connects to the local target ADS Device.</td>
</tr>
<tr>
<td>Connect(String, Int32) [212]</td>
<td>Connects to the target ADS Device.</td>
</tr>
<tr>
<td>Connect(AmsNetId, Int32) [213]</td>
<td>Connects to the target ADS Device.</td>
</tr>
<tr>
<td>Connect(AmsNetId, AmsPort) [214]</td>
<td>Connects to the target ADS Device.</td>
</tr>
<tr>
<td>ConnectAndWaitAsync [214]</td>
<td>Connects to the target address and waits until the AdsClient [154] is disconnected asynchronously.</td>
</tr>
<tr>
<td>CreateVariableHandle [215]</td>
<td>Determines the Symbol handle by its instance path synchronously.</td>
</tr>
<tr>
<td>CreateVariableHandleAsync [216]</td>
<td>Determines the Symbol handle by its instance path asynchronously.</td>
</tr>
<tr>
<td>DeleteDeviceNotification [216]</td>
<td>Deletes a registered notification.</td>
</tr>
<tr>
<td>DeleteDeviceNotificationAsync [217]</td>
<td>Deletes a registered notification asynchronously.</td>
</tr>
<tr>
<td>DeleteVariableHandle [218]</td>
<td>Releases the specified symbol/variable handle synchronously.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DeleteVariableHandleAsync</td>
<td>Releases the specified symbol/variable handle asynchronously.</td>
</tr>
<tr>
<td>Disconnect</td>
<td>Disconnects this AdsClient from the local ADS router.</td>
</tr>
<tr>
<td>Dispose</td>
<td>Performs application-defined tasks associated with freeing, releasing, or</td>
</tr>
<tr>
<td></td>
<td>resetting unmanaged resources.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Finalizes an instance of the AdsClient class.</td>
</tr>
<tr>
<td></td>
<td>(Overloads Object.Finalize.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String,</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>String, Object.)</td>
<td></td>
</tr>
<tr>
<td>InvokeRpcMethod(String,</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>String, Object, Object.)</td>
<td></td>
</tr>
<tr>
<td>InvokeRpcMethod(String,</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object.)</td>
<td></td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String,</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>String, Object, CancellationToken)</td>
<td></td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String,</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>String, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>InvokeRpcMethodA sync(IRpcCallable Ins tance,</td>
<td>invoke RPC method as an asynchronous operation.</td>
</tr>
<tr>
<td>IRpcMethod, Objec t, .AnyTypeSpecifier</td>
<td></td>
</tr>
<tr>
<td>.AnyTypeSpecifier, CancellationToken)</td>
<td></td>
</tr>
<tr>
<td>Read(UInt32, Memory) [233]</td>
<td></td>
</tr>
<tr>
<td>Read(UInt32, UInt32, Memory) [233]</td>
<td></td>
</tr>
<tr>
<td>ReadAny(UInt32, Type) [236]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, Int32.) [238]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type) [240]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type, Int32.) [241]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32) [235]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, Int32.) [235]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, UInt32) [237]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, UInt32, Int32.) [239]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, CancellationToken) [245]</td>
<td>Reads the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, Int32., CancellationToken)</td>
<td>Reads the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, CancellationToken)</td>
<td>Reads the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>ReadAnyAsync(UInt32, UInt32, Type, Int32, CancellationToken)</code></td>
<td>Reads the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td><code>ReadAnyAsync.T.(UInt32, CancellationToken)</code></td>
<td>Reads data synchronously from an ADS device.</td>
</tr>
<tr>
<td><code>ReadAnyAsync.T.(UInt32, Int32, CancellationToken)</code></td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyAsync.T.(UInt32, UInt32, Int32, CancellationToken)</code></td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyAsync.T. (UInt32, UInt32, Int32, CancellationToken)</code></td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyString(UInt32, Int32, Encoding)</code></td>
<td>Reads a string from the specified symbol/variable.</td>
</tr>
<tr>
<td><code>ReadAnyString(UInt32, UInt32, Int32, Encoding)</code></td>
<td>Reads a string from a specified address.</td>
</tr>
<tr>
<td><code>ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken)</code></td>
<td>Reads a string asynchronously from the specified symbol/variable.</td>
</tr>
<tr>
<td><code>ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken)</code></td>
<td>Reads a string from a specified address asynchronously.</td>
</tr>
<tr>
<td><code>ReadAsync(UInt32, Memory, Void)</code></td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadDataTypeAsync</td>
<td>read data type as an asynchronous operation.</td>
</tr>
<tr>
<td>ReadDeviceInfo</td>
<td>Reads the identification and version number of an ADS server.</td>
</tr>
<tr>
<td>ReadDeviceInfoAsync</td>
<td>Reads the identification and version number of an ADS server.</td>
</tr>
<tr>
<td>ReadState</td>
<td>Reads the ADS status and the device status from an ADS server.</td>
</tr>
<tr>
<td>ReadStateAsync</td>
<td>Reads the ADS status and the device status from an ADS server.</td>
</tr>
<tr>
<td>ReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>ReadSymbolAsync</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>ReadValue(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>ReadValue(String, Type)</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>ReadValue.T(String)</td>
<td>Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>ReadValue.T.(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>ReadValueAsync(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously and returns it as an object. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.</td>
</tr>
<tr>
<td>ReadValueAsync(String, Type, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>ReadValueAsync.TValue(String, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>ReadValueAsync.TValue(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously and returns it as an object.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadWriteAsync(UInt32, Memory, Void, Byte)</td>
<td>Registers for <code>AdsStateChanged</code> events as an asynchronous operation.</td>
</tr>
<tr>
<td>ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte)</td>
<td></td>
</tr>
<tr>
<td>RegisterAdsStateChangedAsync</td>
<td>Registers for an <code>AdsSymbolVersionChanged</code> event as an asynchronous operation.</td>
</tr>
<tr>
<td>RegisterSymbolVersionChangedAsync</td>
<td></td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotification</code> event.</td>
</tr>
<tr>
<td>TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotification</code> event.</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotificationEx</code> event.</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotificationEx</code> event.</td>
</tr>
<tr>
<td>TryCreateVariableHandle</td>
<td>Determines the Symbol handle by its instance path synchronously.</td>
</tr>
<tr>
<td>TryDeleteDeviceNotification(UInt32)</td>
<td>Deletes a registered notification.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryDeleteDeviceNotification(UInt32, Int32)</td>
<td>Deletes a registered notification.</td>
</tr>
<tr>
<td>TryDeleteVariableHandle [285]</td>
<td>Releases the specified symbol/variable handle synchronously.</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object.)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Any TypeSpecifier, AnyTypeSpecifier, Object, Object.)</td>
<td>Invokes the rpc method.</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object.)</td>
<td>Tries the invoke RPC method.</td>
</tr>
<tr>
<td>TryRead(UInt32, Memory, Void)</td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>TryRead(UInt32, UInt32, Memory, Void)</td>
<td></td>
</tr>
<tr>
<td>TryReadDataType</td>
<td></td>
</tr>
<tr>
<td>TryReadState [295]</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.</td>
</tr>
<tr>
<td>TryReadSymbol [296]</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>TryReadValue(ISymbol, Object)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryReadValue(String, Type, Object)</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
<tr>
<td>TryReadValue(String, T.)</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
<tr>
<td>TryReadValue(ISymbol, T.)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, Memory, Void, Byte)</td>
<td></td>
</tr>
<tr>
<td>TryWrite(UInt32, ReadOnlyMemory)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo)</td>
<td></td>
</tr>
<tr>
<td>TryWriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryWriteValue(ISymbol, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>TryWriteValue(String, T.)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryWriteValue(ISymbol, T.)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>UnregisterAdsStateChangedAsync</td>
<td>unregister ads state changed as an asynchronous operation.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>UnregisterSymbolVersionChangedAsync</td>
<td>Unregisters from an AdsSymbolVersionChanged event as an asynchronous operation.</td>
</tr>
<tr>
<td>Write(UInt32, ReadOnlyMemory)</td>
<td>Triggers a 'Write' call to the ADS device at the specified address.</td>
</tr>
<tr>
<td>Write(UInt32, UInt32, ReadOnlyMemory)</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object, Int32.)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)</td>
<td>Writes the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, Int32., CancellationToken)</td>
<td>Writes the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteAnyStringAsyncc(String, String, Int32, Encoding, CancellationToken)</td>
<td>![322] Writes the string (Potentially unsafe!)</td>
</tr>
<tr>
<td>WriteAnyStringAsyncc(UInt32, String, Int32, Encoding, CancellationToken)</td>
<td>![323]</td>
</tr>
<tr>
<td>WriteAsync(UInt32, ReadOnlyMemory, Void)</td>
<td>![324] Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.</td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, CancellationToken)</td>
<td>![325]</td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void)</td>
<td>![326]</td>
</tr>
<tr>
<td>WriteControl(StateInfo)</td>
<td>![327] Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>WriteControl(StateInfo, ReadOnlyMemory)</td>
<td>![328]</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, CancellationToken)</td>
<td>![328] Changes the ADS status and device status of the ADS server asynchronously.</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void)</td>
<td>![329]</td>
</tr>
<tr>
<td>WriteSymbolAsync</td>
<td>![330] Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue(String, Object)</td>
<td>![331] Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue(Symbol, Object)</td>
<td>![332] Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValue.T.(String, T)</td>
<td>![332] Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValue.T. ([ISymbol, T] [333])</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValueAsync(ISymbol, Object, CancellationToken) [335]</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValueAsync.T. (String, T, CancellationToken) [334]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
</tbody>
</table>

### Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollAdsState(IObservable&lt;Unit&gt;) [1061]</td>
<td>Overloaded. Gets an observable sequence of AdsState [626]s via Polling. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>PollAdsStateAsync(IObservable&lt;Unit&gt;, CancellationToken) [1064]</td>
<td>Overloaded. Gets an observable sequence of AdsState [626]s via Polling. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>PollValues(String, Type, Observable&lt;Unit&gt;) [1089]</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan) [1090]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, TimeSpan) [1093]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PollValues(String, Type, IObservable.Unit, Func.Exception, Object)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan, Func.Exception, Object)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, IObservable.Unit, Func.Exception, Object)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, TimeSpan, Func.Exception, Object)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T(String, IObservable.Unit)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T(String, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T(String, Int32, IObservable.Unit, Func.Exception, T)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T(String, Int32, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T(String, Int32, IObservable.Unit, Func.Exception, T)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T(String, Int32, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WhenAdsStateChanges</td>
<td>Gets an observable sequence of AdsState objects. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol)</td>
<td>Overloaded. Gets an observable sequence of Notification objects. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection)</td>
<td>Overloaded. Gets an observable sequence of Notification objects. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol, NotificationSettings)</td>
<td>Overloaded. Gets an observable sequence of SymbolValueNotification objects. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection, NotificationSettings)</td>
<td>Overloaded. Gets an observable sequence of Notification objects. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(String, NotificationSettings)</td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>WhenValueChanged</td>
<td>Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions.)</td>
</tr>
<tr>
<td>WriteValues(String, IObservable)</td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>WriteValues(String, IObservable, Action.Exception)</td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions.)</td>
</tr>
</tbody>
</table>
### AdsClient.AddDeviceNotification Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotification(String, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
</tbody>
</table>

#### Reference

- AdsClient Class [154]
- TwinCAT.Ads Namespace [151]

### AdsClient.AddDeviceNotification Method (String, Int32, NotificationSettings, Object)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public uint AddDeviceNotification(
    string symbolPath,
    int dataSize,
    NotificationSettings settings,
    Object userData
)
```

**Parameters**

- **symbolPath**
  - Type: System.String
  - Symbol / Instance path of the ADS variable.

- **dataSize**
  - Type: System.Int32
  - Maximum amount of data in bytes to receive with this ADS Notification.

- **settings**
  - Type: TwinCAT.Ads.NotificationSettings [979]
  - The settings.
userData

Type: System.Object

This object can be used to store user specific data (tag data)

Return Value

Type: UInt32

The notification handle.

Implements

IAdsNotifications.AddDeviceNotification(String, Int32, NotificationSettings, Object) [844]

Remarks

The dataSize Parameter defines the amount of bytes, that will be attached to the AdsNotification [338] as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [216] should always called when the notification is not used anymore.

Reference

AdsClient Class [154]

AddDeviceNotification Overload [195]

TwinCAT.Ads Namespace [151]

AdsClient.AdsNotification [338]

AdsClient.DeleteDeviceNotification(UInt32) [216]

AddDeviceNotification Overload [844]

AddDeviceNotificationAsync Overload [847]

TryAddDeviceNotification Overload [859]

AdsClient.AddDeviceNotification Method (UInt32, UInt32, Int32, NotificationSettings, Object)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:

5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public uint AddDeviceNotification(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData
)
```
Parameters

- **indexGroup**
  - Type: System.UInt32
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - Type: System.UInt32
  - Contains the index offset number of the requested ADS service.

- **dataSize**
  - Type: System.Int32
  - Maximum amount of data in bytes to receive with this ADS Notification.

- **settings**
  - Type: TwinCAT.Ads.NotificationSettings
  - The Notification settings.

- **userData**
  - Type: System.Object
  - This object can be used to store user specific data (tag data)

Return Value

- Type: UInt32
  - The notification handle.

Implements

- IAdsNotifications.AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object)

Remarks

- The `dataSize` Parameter defines the amount of bytes, that will be attached to the AdsNotification as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) should always called when the notification is not used anymore.

Reference

- AdsClient Class
- AddDeviceNotification Overload
- TwinCAT.Ads Namespace
- AdsClient_DeleteDeviceNotification(UInt32)
- AdsClient.AdvNotification
- AdsClient.AdvNotificationError
- AddDeviceNotification Overload
- TryAddDeviceNotification Overload
- AddDeviceNotificationAsync Overload
6.2.1.2.2 AdsClient.AddDeviceNotificationAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [198]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [338] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [199]</td>
<td>Adds a device notification as an asynchronous operation.</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]

AdsClient.AddDeviceNotificationAsync Method (String, Int32, NotificationSettings, Object, CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [338] event.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultHandle> AddDeviceNotificationAsync(
    string symbolPath,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    CancellationToken cancel
)
```

Parameters

symbolPath
Type: System.String
The symbol/instance path of the ADS variable.

dataSize
Type: System.Int32
Maximum amount of data in bytes to receive with this ADS Notification.

settings
Type: TwinCAT.Ads.NotificationSettings [979]
The notification settings.
userData

Type: System.Object
This object can be used to store user specific data (tag data)

cancel

Type: System.Threading.CancellationToken
The Cancellation token.

Return Value
Type: Task.ResultHandle [1005]
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [1005] type parameter contains the created handle (Handle [1007]) and the ErrorCode [992] after execution.

Implements
IAdsNotifications.AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [847]

Remarks
The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [338] as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotificationAsync(UInt32, CancellationToken) [217] should always be called when the notification is not used anymore.

Reference
AdsClient Class [154]
AddDeviceNotificationAsync Overload [198]
TwinCAT.Ads Namespace [151]
AdsClient.AdsNotification [338]
AdsClient.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [217]
AddDeviceNotification Overload [844]
AddDeviceNotificationAsync Overload [847]
TryAddDeviceNotification Overload [859]

AdsClient.AddDeviceNotificationAsync Method (UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)

Adds a device notification as an asynchronous operation.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public Task<ResultHandle> AddDeviceNotificationAsync(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    CancellationToken cancel
)
```

**Parameters**

- **indexGroup**
  - Type: `System.UInt32`
  - The index group number of the requested ADS service.

- **indexOffset**
  - Type: `System.UInt32`
  - The index offset number of the requested ADS service.

- **dataSize**
  - Type: `System.Int32`
  - Maximum amount of data in bytes to receive with this ADS Notification.

- **settings**
  - Type: `TwinCAT.Ads.NotificationSettings`
  - The notification settings.

- **userData**
  - Type: `System.Object`
  - This object can be used to store user specific data (tag data)

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The Cancellation token.

**Return Value**

Type: `Task<ResultHandle>`

A task that represents the asynchronous 'AddDeviceNotification' operation. The `ResultHandle` type parameter contains the created handle (Handle) and the `ErrorCode` after execution.

**Implements**

`IAdsNotifications.AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)`

**Remarks**

The `dataSize` parameter defines the amount of bytes, that will be attached to the `AdsNotification` as value. Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotificationAsync(UInt32, CancellationToken)` should always be called when the notification is not used anymore.

**Reference**

- `AdsClient Class`
- `AddDeviceNotificationAsync Overload`
- `TwinCAT.Ads Namespace`
- `IAdsNotifications.AdsNotification`
6.2.1.2.3 AdsClient.AddDeviceNotificationEx Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.</td>
</tr>
</tbody>
</table>

### Reference

- AdsClient Class [154]
- TwinCAT.Ads Namespace [151]

### AdsClient.AddDeviceNotificationEx Method (String, NotificationSettings, Object, Type)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public uint AddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type
)
```

**Parameters**

- **symbolPath**
  - Type: `System.String`
  - Symbol/Instance path of the ADS variable.

- **settings**
  - Type: `TwinCAT.Ads.NotificationSettings`
  - The Notification settings.

- **userData**
  - Type: `System.Object`
  - This object can be used to store user specific data (tag data)

- **type**
  - Type: `System.Type`
  - Type of the object stored in the event argument ('AnyType')

**Return Value**

- Type: `UInt32`
  - The notification handle.

**Implements**

- `IAdsNotifications.AddDeviceNotificationEx(String, NotificationSettings, Object, Type)`

**Remarks**

Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotification(UInt32)` should always be called when the notification is not used anymore.

**Reference**

- `AdsClient Class`
- `AddDeviceNotificationEx Overload`
- `TwinCAT.Ads Namespace`
- `AdsClient AdsNotificationEx` (`340`)
- `AdsClient.DeleteDeviceNotification(UInt32)` (`216`)
- `AddDeviceNotificationEx Overload` (`850`)
- `AddDeviceNotificationExAsync Overload` (`855`)
- `TryAddDeviceNotificationEx Overload` (`862`)

**AdsClient.AddDeviceNotificationEx Method (String, NotificationSettings, Object, Type, .Int32.)**

Connects a variable to the ADS client. The ADS client will be notified by the `AdsNotificationEx` event.
Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public uint AddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args
)
```

Parameters

- **symbolPath**: Type: System.String
  Symbol/Instance path of the ADS variable.
- **settings**: Type: TwinCAT.Ads.NotificationSettings
  The Notification settings.
- **userData**: Type: System.Object
  This object can be used to store user specific data (tag data)
- **type**: Type: System.Type
  Type of the object stored in the event argument ('AnyType')
- **args**: Type: System.Int32
  Additional arguments (for 'AnyType')

Return Value

Type: UInt32
The notification handle.

Implements

IAdsNotifications.AddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32.)

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) should always called when the notification is not used anymore.

Reference

AdsClient Class
AddDeviceNotificationEx Overload
TwinCAT.Ads Namespace
AdsClient.AdsNotificationEx
AdsClient.DeleteDeviceNotification(UInt32)
AddDeviceNotificationEx Overload
AddDeviceNotificationExAsync Overload
TryAddDeviceNotificationEx Overload
AdsClient.AddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [340] event.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public uint AddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type
)
```

Parameters

- **indexGroup**
  - Type: System.UInt32
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - Type: System.UInt32
  - Contains the index offset number of the requested ADS service.

- **settings**
  - Type: TwinCAT.Ads.NotificationSettings [979]
  - The Notification settings.

- **userData**
  - Type: System.Object
  - This object can be used to store user specific data (tag data)

- **type**
  - Type: System.Type
  - Type of the object stored in the event argument ('AnyType')

Return Value

Type: UInt32
The notification handle.

Implements

|AdsNotifications.AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [852]

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [216] should always called when the notification is not used anymore.

Reference

- AdsClient Class [154]
- AddDeviceNotificationEx Overload [201]
- TwinCAT.Ads Namespace [151]
- AdsClient.DeleteDeviceNotification(UInt32) [216]
- AdsClient.AdsNotificationEx [340]
AdsClient.AdsNotificationError [339]

AddDeviceNotificationEx Overload [850]

TryAddDeviceNotificationEx Overload [862]

AddDeviceNotificationExAsync Overload [855]

**AdsClient.AddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32.)**

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public uint AddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args
)
```

**Parameters**

- **indexGroup**
  - Type: System.UInt32
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - Type: System.UInt32
  - Contains the index offset number of the requested ADS service.

- **settings**
  - Type: TwinCAT.Ads.NotificationSettings [979]
  - The Notification settings.

- **userData**
  - Type: System.Object
  - This object can be used to store user specific data.

- **type**
  - Type: System.Type
  - Type of the object stored in the event argument.

- **args**
  - Type: System.Int32
  - Additional arguments for 'AnyType' types.

**Return Value**

Type: UInt32

The notification handle.

**Implements**

IAdsNotifications.AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [853]
Remarks

Because notifications allocate TwinCAT system resources, a complementary call to
DeleteDeviceNotification(UInt32) [216] should always called when the notification is not used anymore.

Reference

AdsClient Class [154]

AddDeviceNotificationEx Overload [201]

TwinCAT.Ads Namespace [151]

AdsClient.AdsNotificationEx [340]

AddDeviceNotificationEx Overload [850]

AddDeviceNotificationExAsync Overload [855]

TryAddDeviceNotificationEx Overload [862]

6.2.1.2.4 AdsClient.AddDeviceNotificationExAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, Int32, CancellationToken) [206]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [340] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, Int32, CancellationToken) [208]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [340] event.</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

AdsClient.AddDeviceNotificationExAsync Method (String, NotificationSettings, Object, Type, .Int32., CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [340] event.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd72bc0ea15da1c14
Syntax

C#

```csharp
public Task<ResultHandle> AddDeviceNotificationExAsync(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

Parameters

- **symbolPath**
  Type: `System.String`
  The symbol/instance path of the ADS variable.

- **settings**
  Type: `TwinCAT.Ads.NotificationSettings` *(979)*
  The notification settings.

- **userData**
  Type: `System.Object`
  This object can be used to store user specific data (tag data)

- **type**
  Type: `System.Type`
  Type of the object stored in the event argument ('AnyType')

- **args**
  Type: `System.Int32`
  Additional arguments (for 'AnyType')

- **cancel**
  Type: `System.Threading.CancellationToken`
  The Cancellation token.

Return Value

Type: `Task<ResultHandle>` *(1005)*
A task that represents the asynchronous 'AddDeviceNotification' operation. The `ResultHandle` *(1005)* type parameter contains the created handle (`Handle` *(1007)*) and the `ErrorCode` *(992)* after execution.

Implements

`IAdsNotifications.AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, Int32, CancellationToken)` *(855)*

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotificationAsync(UInt32, CancellationToken)` *(217)* should always be called when the notification is not used anymore.

Reference

- `AdsClient Class` *(154)*
- `AddDeviceNotificationExAsync Overload` *(206)*
- `TwinCAT_Ads Namespace` *(151)*
- `AdsClient.AdsNotificationEx` *(340)*
- `AdsClient.DeleteDeviceNotificationAsync(UInt32, CancellationToken)` *(217)*
- `AddDeviceNotificationEx Overload` *(850)*
- `AddDeviceNotificationExAsync Overload` *(855)*
TryAddDeviceNotificationEx Overload [862]

**AdsClient.AddDeviceNotificationExAsync Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken)**

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [340] event.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Task<ResultHandle> AddDeviceNotificationExAsync(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type anyType,
    int[] args,
    CancellationToken cancel)
```

**Parameters**

- **indexGroup**
  - **Type:** System.UInt32
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - **Type:** System.UInt32
  - Contains the index offset number of the requested ADS service.

- **settings**
  - **Type:** TwinCAT.Ads.NotificationSettings [979]
  - The settings.

- **userData**
  - **Type:** System.Object
  - This object can be used to store user specific data.

- **anyType**
  - **Type:** System.Type
  - Type of the object stored in the event argument, only Primitive 'AnyTypes' allowed.

- **args**
  - **Type:** System.Int32
  - Additional arguments (for 'AnyType')

- **cancel**
  - **Type:** System.Threading.CancellationToken
  - The Cancellation token.

**Return Value**

- **Type:** Task<ResultHandle> [1005]
  - A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [1005] type parameter contains the created handle (Handle [1007]) and the ErrorCode [992] after execution.

**Implements**

- AdsNotifications.AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [856]
Remarks

If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive types (AnyType) are supported by this method. Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotificationAsync(UInt32, CancellationToken)` should always be called when the notification is not used anymore.

Reference

`AdsClient` Class [154]

`AddDeviceNotificationExAsync` Overload [206]

TwinCAT.Ads Namespace [151]

`AdsClient.DeleteDeviceNotificationAsync(UInt32, CancellationToken)` [217]

`AdsClient.AdsNotificationEx` [340]

`AdsClient.AdsNotificationError` [339]

`AddDeviceNotificationEx` Overload [850]

`TryAddDeviceNotificationEx` Overload [862]

`AddDeviceNotificationExAsync` Overload [855]

6.2.1.2.5 AdsClient.CleanupSymbolTable Method

Clears the internal symbol cache.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void CleanupSymbolTable()
```

Implements

`IAdsSymbolicAccess.CleanupSymbolTable` [942]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ObjectDisposedException</code></td>
<td></td>
</tr>
<tr>
<td><code>ClientNotConnectedException</code></td>
<td>[63]</td>
</tr>
</tbody>
</table>

Remarks

Previously stored symbol information is cleared. As a consequence the symbol information must be obtained from the ADS server again if accessed, which which needs an extra ADS round trip.
6.2.1.2.6 AdsClient.Close Method

Closes this AdsClient.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public void Close()

Implements

IConnection.Close

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]

6.2.1.2.7 AdsClient.Connect Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect(Int32)</td>
<td>Connects to the local target ADS Device.</td>
</tr>
<tr>
<td>Connect(AmsAddress)</td>
<td>Connects the target</td>
</tr>
<tr>
<td>Connect(AmsPort)</td>
<td>Connects to the local target ADS Device.</td>
</tr>
<tr>
<td>Connect(String, Int32)</td>
<td>Connects to the target ADS Device.</td>
</tr>
<tr>
<td>Connect(AmsNetId, Int32)</td>
<td>Connects to the target ADS Device.</td>
</tr>
<tr>
<td>Connect(AmsNetId, AmsPort)</td>
<td>Connects to the target ADS Device.</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]
**AdsClient.Connect Method (Int32)**

Connects to the local target ADS Device.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public void Connect(
    int port
)
```

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>port</td>
<td>System.Int32</td>
<td>The port number of the local ADS target device to connect to.</td>
</tr>
</tbody>
</table>

**Implements**

IAdsConnectAddress.Connect(Int32) [763]

**Reference**

AdsClient Class [154]

Connect Overload [210]

TwinCAT.Ads Namespace [151]

---

**AdsClient.Connect Method (AmsAddress)**

Connects the target

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public void Connect(
    AmsAddress address
)
```

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>TwinCAT.Ads.AmsAddress</td>
<td>The address.</td>
</tr>
</tbody>
</table>

**Implements**

IAdsConnectAddress.Connect(AmsAddress) [763]
Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

Connect Overload [210]

TwinCAT.Ads Namespace [151]

**AdsClient.Connect Method (AmsPort)**

Connects to the local target ADS Device.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void Connect(
    AmsPort port
)
```

**Parameters**

- **port**
  - Type: TwinCAT.Ads.AmsPort [693]
  - The port number of the local ADS target device to connect to.

Reference

AdsClient Class [154]

Connect Overload [210]

TwinCAT.Ads Namespace [151]

**AdsClient.Connect Method (String, Int32)**

Connects to the target ADS Device.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void Connect(
    string netId,
    int port
)
```
Parameters

netId
Type: System.String
The AmsNetId of the ADS target device specified as string.

port
Type: System.Int32
The port number of the ADS target device.

Implements

IAdsConnectAddress.Connect(String, Int32)

Reference

AdsClient Class
Connect Overload
TwinCAT.Ads Namespace

AdsClient.Connect Method (AmsNetId, Int32)

Connects to the target ADS Device.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public void Connect(
    AmsNetId netId,
    int port
)

Parameters

netId
Type: TwinCAT.Ads.AmsNetId
The AmsNetId of the target device.

port
Type: System.Int32
The Ams Port number on the target device to connect to.

Implements

IAdsConnectAddress.Connect(AmsNetId, Int32)

Reference

AdsClient Class
Connect Overload
TwinCAT.Ads Namespace
**AdsClient.Connect Method (AmsNetId, AmsPort)**

Connects to the target ADS Device.

**Parameters**

- `netId` Type: `TwinCAT.Ads.AmsNetId` [665]
  The `AmsNetId` [665] of the ADS target device specified as string.

- `port` Type: `TwinCAT.Ads.AmsPort` [693]
  The port number of the ADS target device.

**Reference**

- `AdsClient Class` [154]
- `Connect Overload` [210]
- `TwinCAT.Ads Namespace` [151]

### 6.2.1.2.8 AdsClient.ConnectAndWaitAsync Method

Connects to the target address and waits until the `AdsClient` [154] is disconnected asynchronously.

**Parameters**

- `address` Type: `TwinCAT.Ads.AmsAddress` [648]
  The target address.

- `cancel` Type: `System.Threading.CancellationToken`
  Cancellation Token.

**Return Value**

Type: `Task`

Returns a task object that represents the ConnectAndWaitAsync(AmsAddress, CancellationToken) operation as result.
Remarks

This method is used for scenarios, where the AdsClient disconnects from other code asynchronously. When this method returns, the connection is already terminated and only additional cleanup code should be processed.

Reference

AdsClient Class

TwinCAT.Ads Namespace

6.2.1.2.9 AdsClient.CreateVariableHandle Method

Determines the Symbol handle by its instance path synchronously.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C# public uint CreateVariableHandle(  string symbolPath  )

Parameters

symbolPath Type: System.String SymbolName / InstancePath.

Return Value

Type: UInt32 The symbols/variable handle

Implements

IAdsHandle.CreateVariableHandle(String)

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this CreateVariableHandle(String) is the DeleteVariableHandle(UInt32)

Reference

AdsClient Class

TwinCAT.Ads Namespace

AdsClient.DeleteVariableHandle(UInt32)
AdsClient.CreateVariableHandleAsync(String, CancellationToken)
AdsClient.TryCreateVariableHandle(String, UInt32.)
6.2.1.2.10 AdsClient.CreateVariableHandleAsync Method

Determines the Symbol handle by its instance path asynchronously.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Task<ResultHandle> CreateVariableHandleAsync(
    string symbolPath,
    CancellationToken cancel
)
```

**Parameters**

- `symbolPath`  
  Type: System.String  
  SymbolName / InstancePath.

- `cancel`  
  Type: System.Threading.CancellationToken  
  The cancellation token.

**Return Value**

Type: Task<ResultHandle>  
A task that represents the asynchronous 'CreateVariableHandle' operation. The ResultHandle parameter contains the variable handle (Handle) and the ErrorCode after execution.

**Implements**

IAdsHandle.CreateVariableHandleAsync(String, CancellationToken)

**Remarks**

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this CreateVariableHandleAsync(String, CancellationToken) is the DeleteVariableHandleAsync(UInt32, CancellationToken).

**Reference**

AdsClient Class

TwinCAT.Ads Namespace

AdsClient.DeleteVariableHandleAsync(UInt32, CancellationToken)

AdsClient.TryCreateVariableHandle(String, UInt32.)

AdsClient.CreateVariableHandle(String)

6.2.1.2.11 AdsClient.DeleteDeviceNotification Method

Deletes a registered notification.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
public void DeleteDeviceNotification(
    uint notificationHandle
)
```

Parameters

- **notificationHandle**
  - Type: `System.UInt32`
  - Notification handle.

Implements

- `IAdsNotifications.DeleteDeviceNotification(UInt32)`

Remarks

This is the complementary method to `AddDeviceNotification Overload` overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

- `AdsClient Class` [857]

6.2.1.2.12  AdsClient.DeleteDeviceNotificationAsync Method

Deletes a registered notification asynchronously.

Namespace:  `TwinCAT.Ads` [151]
Assembly:  `TwinCAT.Ads` (in `TwinCAT.Ads.dll`) Version:  5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultAds> DeleteDeviceNotificationAsync(
    uint notificationHandle,
    CancellationToken cancel
)
```

Parameters

- **notificationHandle**
  - Type: `System.UInt32`
  - Notification handle.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.
Return Value

Type: Task.ResultAds [989].
A task that represents the asynchronous 'DeleteDeviceNotification' operation. The ErrorCode [992] property contains the ADS error code after execution.

Implements

IAdsNotifications.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [858]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException [63]</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

This is the complementary method to AddDeviceNotificationAsync Overload [847] overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

AddDeviceNotificationAsync Overload [847]

IAdsNotifications.AdsNotification [866]

TryAddDeviceNotification Overload [859]

AddDeviceNotification Overload [844]

6.2.1.2.13 AdsClient.DeleteVariableHandle Method

Releases the specified symbol/variable handle synchronously.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void DeleteVariableHandle(
    uint variableHandle
)
```

Parameters

variableHandle Type: System.UInt32
Handle of the ADS variable
Return Value
Type: The ADS error code.

Implements
| AdsHandle.DeleteVariableHandle(UInt32) [158]

Remarks
It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this TryDeleteVariableHandle(UInt32) [154] is the TryCreateVariableHandle(String, UInt32) [151].

Reference
AdsClient Class [154]
TwinCAT.Ads Namespace [151]
AdsClient.CreateVariableHandle(String) [215]
AdsClient.TryDeleteVariableHandle(UInt32) [285]
AdsClient.DeleteVariableHandleAsync(UInt32, CancellationToken) [219]

6.2.1.2.14 AdsClient.DeleteVariableHandleAsync Method
Releases the specified symbol/variable handle asynchronously.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
public Task<ResultAds> DeleteVariableHandleAsync(
    uint variableHandle,
    CancellationToken cancel
)

Parameters
variableHandle Type: System.UInt32
Handle of the ADS variable
cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value
Type: Task<ResultAds> [989].
A task that represents the asynchronous ‘DeleteVariableHandle’ operation. The ResultAds [989] parameter contains the ErrorCode [992] after execution.
Implements

IAdsHandle.DeleteVariableHandleAsync(UInt32, CancellationToken) [831]

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this DeleteVariableHandleAsync(UInt32, CancellationToken) is the CreateVariableHandleAsync(String, CancellationToken) [216]

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

AdsClient.CreateVariableHandleAsync(String, CancellationToken) [216]

AdsClient.TryDeleteVariableHandle(UInt32) [285]

AdsClient.DeleteVariableHandle(UInt32) [218]

6.2.1.2.15 AdsClient.Disconnect Method

Disconnects this AdsClient [154] from the local ADS router.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public bool Disconnect()

Return Value

Type: Boolean
ture if disconnected, false otherwise.

Implements

IConnection.Disconnect [79]

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

6.2.1.2.16 AdsClient.Dispose Method

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
6.2.1.2.17 AdsClient.Finalize Method

Finalizes an instance of the AdsClient [154] class.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected override void Finalize()
```

Implements

Object.Finalize.

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvokeRpcMethod(String, String, Object, AnyObjectTypeSpecifier, AnyTypeSpecifier, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

**AdsClient.InvokeRpcMethod Method (String, String, .Object.)**

Invokes the specified RPC Method

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters
)
```

**Parameters**

- **symbolPath**  
  Type: System.String  
  The symbol path.

- **methodName**  
  Type: System.String  
  The method name.

- **inParameters**  
  Type: System.Object  
  The input parameters or NULL

**Return Value**

Type: Object  
The return value of the Method (as object).

**Implements**

IAdsRpcInvoke.InvokeRpcMethod(String, String, Object) [890]

**Remarks**

This method only supports primitive data types as inParameters. Any available outparameters will be ignored. Complex types will fall back to byte[] arrays.
Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```csharp
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;

            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /* {attribute 'TcRpcEnable'}
               METHOD PUBLIC M_Add : INT
               VAR_INPUT
               l1 : INT := 0;
               l2 : INT := 0;
               END_VAR
            */

            short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short) 3, (short) 4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});

            //Browsing RpcMethods
            foreach(IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;
                foreach(IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}
```

Reference

AdsClient Class [▲ 154]
InvokeRpcMethod Overload [▲ 221]
TwinCAT.Ads Namespace [▲ 151]

AdsClient.InvokeRpcMethod Method (String, String, .Object., .Object..)
Invokes the specified RPC Method
Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object[] outParameters
)
```

Parameters

- **symbolPath**: Type: System.String
  The symbol path.
- **methodName**: Type: System.String
  The method name.
- **inParameters**: Type: System.Object
  The input parameters or NULL
- **outParameters**: Type: System.Object
  The output parameters.

Return Value

Type: Object
The return value of the Method (as object).

Implements

IAdsRpcInvoke.InvokeRpcMethod(String, String, Object, Object) [891]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

**Dynamic Tree Mode**

```csharp
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);
```
SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

// Get the Symbols (Dynamic Symbols)
IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

// Call a Method that has the following signature (within MAIN Program)
/* (attribute 'TcRpcEnable')
METHOD PUBLIC M_Add : INT
VAR_INPUT
  i1 : INT := 0;
  i2 : INT := 0;
END_VAR
*/
short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short)3, (short)4});;

// Call a Method that has no parameter and returns VOID
main.InvokeRpcMethod("M_Method1", new object[]{});;

// Browsing RpcMethods
foreach(IRpcMethod method in main.RpcMethods)
{
  string methodName = method.Name;
  foreach(IRpcMethodParameter parameter in method.Parameters)
  {
    string parameterName = parameter.Name;
    string parameterType = parameter.TypeName;
  }
}

Reference

AdsClient Class [154]
InvokeRpcMethod Overload [221]

TwinCAT.Ads Namespace [151]

AdsClient.InvokeRpcMethod Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

public Object InvokeRpcMethod(
  string symbolPath,
  string methodName,
  Object[] inParameters,
  AnyTypeSpecifier[] outSpecifiers,
  AnyTypeSpecifier retSpecifier,
  out Object[] outParameters
)
Parameters

symbolPath
Type: System.String
The symbol path.

methodName
Type: System.String
The method name.

inParameters
Type: System.Object
The parameters.

outSpecifiers
Type: TwinCAT.TypeSystem.AnyTypeSpecifier
The out specifiers (specifying the out types) or NULL.

retSpecifier
Type: TwinCAT.TypeSystem.AnyTypeSpecifier
The ret specifier (specifying the return value) or NULL.

outParameters
Type: System.Object
The out parameters.

Return Value

Type: Object
The return value of the Method (as object).

Implements

IAdsRpcInvoke.InvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object)

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
/* {attribute 'TcRpcEnable'}
METHOD PUBLIC M_Add : INT
VAR_INPUT
   i1 : INT := 0;
i2 : INT := 0;
END_VAR
*/

short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short)3, (short)4});

// Call a Method that has no parameter and returns VOID
main.InvokeRpcMethod("M_Method1", new object{});

// Browsing RpcMethods
foreach(IRpcMethod method in main.RpcMethods)
{
    string methodName = method.Name;
    foreach(IRpcMethodParameter parameter in method.Parameters)
    {
        string parameterName = parameter.Name;
        string parameterType = parameter.TypeName;
    }
}

Reference

AdsClient Class [154]
InvokeRpcMethod Overload [221]
TwinCAT.Ads Namespace [151]

6.2.1.2.19 AdsClient.InvokeRpcMethodAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvokeRpcMethodAsync(String, String, Object, CancellationToken) [228]</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken) [230]</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvokeRpcMethodAsync</td>
<td>invoke RPC method as an asynchronous operation.</td>
</tr>
</tbody>
</table>

### Reference

- **AdsClient Class** [154]
- **TwinCAT.Ads Namespace** [151]

### AdsClient.InvokeRpcMethodAsync Method (String, String, .Object., CancellationToken)

Invokes the specified RPC Method asynchronously

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

#### Syntax

C#

```csharp
public Task<ResultRpcMethod> InvokeRpcMethodAsync(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    CancellationToken cancel
)
```

#### Parameters

- **symbolPath**
  - Type: `System.String`
  - The symbol/Instance path of the symbol.

- **methodName**
  - Type: `System.String`
  - The method name.

- **inParameters**
  - Type: `.System.Object`
  - The parameters.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token

#### Return Value

- Type: `Task<ResultRpcMethod>` [1025]
  - A task that represents the asynchronous 'InvokeRpcMethod' operation. The `ResultRpcMethod` [1025] results contains the return value together with the output parameters.

#### Implements

- `IAdsRpcInvoke.InvokeRpcMethodAsync(String, String, .Object., CancellationToken)` [895]
Remarks

Because this overload doesn't provide any AnyTypeSpecifier \[1633\] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

**Dynamic Tree Mode**

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /* {attribute 'TcRpcEnable'}
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
            END_VAR
            */

            short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short)3, (short)4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[]{});

            // Browsing RpcMethods
            foreach(IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;
                foreach(IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}

Reference

- AdsClient Class [154]
- InvokeRpcMethodAsync Overload [227]
- TwinCAT.Ads Namespace [151]
AdsClient.InvokeRpcMethodAsync Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken)

Invokes the specified RPC Method asynchronously

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Task<ResultRpcMethod> InvokeRpcMethodAsync(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    CancellationToken cancel
)
```

**Parameters**

- **symbolPath**
  - Type: System.String
  - The symbol/Instance path of the symbol.

- **methodName**
  - Type: System.String
  - The method name.

- **inParameters**
  - Type: System.Object
  - The parameters.

- **outSpecifiers**
  - Type: TwinCAT.TypeSystem.AnyTypeSpecifier
  - The out specifiers (specifying the out types) or NULL.

- **retSpecifier**
  - Type: TwinCAT.TypeSystem.AnyTypeSpecifier
  - The ret specifier (specifying the return value) or NULL.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token

**Return Value**

Type: Task<ResultRpcMethod>

A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethod results contains the return value together with the output parameters.

**Remarks**

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set. ReturnValue and the ErrorCode of the ADS communication after execution.

**Implements**

IAdsRpcInvoke.InvokeRpcMethodAsync(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken) [897]

**Examples**

The following sample shows how to call (Remote Procedures / Methods) within the PLC.
Dynamic Tree Mode

class RpcCallVirtualProgram
{
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static void Main(string[] args)
{
  // Get the AdsAddress from command-line arguments
  AmsAddress address = ArgParser.Parse(args);
  using (AdsClient client = new AdsClient())
  {
    //client.Synchronize = false;
    // Connect to the target device
    client.Connect(address);
    SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
    // Get the Symbols (Dynamic Symbols)
    IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program
    // Call a Method that has the following signature (within MAIN Program)
    /*
     * {attribute 'TcRpcEnable'}
     METHOD PUBLIC M_Add : INT
     VAR_INPUT
      i1 : INT := 0;
      i2 : INT := 0;
     END_VAR
     */
    short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short)3, (short)4});
    // Call a Method that has no parameter and returns VOID
    main.InvokeRpcMethod("M_Method1", new object[]{});
    //Browsing RpcMethods
    foreach(IRpcMethod method in main.RpcMethods)
    {
      string methodName = method.Name;
      foreach(IRpcMethodParameter parameter in method.Parameters)
      {
        string parameterName = parameter.Name;
        string parameterType = parameter.TypeName;
      }
    }
  }
}

Reference

AdsClient Class [154]
InvokeRpcMethodAsync Overload [227]
TwinCAT.Ads Namespace [151]

AdsClient.InvokeRpcMethodAsync Method (IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., .AnyTypeSpecifier, CancellationToken)

invoke RPC method as an asynchronous operation.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

```csharp
public Task<ResultRpcMethod> InvokeRpcMethodAsync(
    IRpcCallableInstance symbol,
    IRpcMethod rpcMethod,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpec,
    AnyTypeSpecifier returnSpec,
    CancellationToken cancel)
```

Parameters

symbol Type: TwinCAT.TypeSystem.IRpcCallableInstance
The symbol.

rpcMethod Type: TwinCAT.TypeSystem.IRpcMethod
The RPC method.

inParameters Type: System.Object
The in parameters.

outSpec Type: TwinCAT.TypeSystem.AnyTypeSpecifier
The out spec.

returnSpec Type: TwinCAT.TypeSystem.AnyTypeSpecifier
The return spec.

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultRpcMethod>
Task<ResultRpcMethod>.

Implements

IAdsRpcInvoke.InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td></td>
</tr>
<tr>
<td>ArgumentNullException</td>
<td>symbol</td>
</tr>
<tr>
<td>ArgumentNullException</td>
<td>rpcMethod</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class | 154 |
InvokeRpcMethodAsync Overload | 227 |
TwinCAT.Ads Namespace | 151 |
6.2.1.2.20 AdsClient.Read Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read(UInt32, Memory)</td>
<td>[233]</td>
</tr>
<tr>
<td>Read(UInt32, UInt32, Memory)</td>
<td>[233]</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

AdsClient.Read Method (UInt32, Memory`1)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```c#
public int Read(
    uint variableHandle,
    Memory readBuffer
)
```

Parameters

variableHandle Type: System.UInt32

readBuffer Type: Memory

Return Value

Type: Int32

Reference

AdsClient Class [154]

Read Overload [233]

TwinCAT.Ads Namespace [151]

AdsClient.Read Method (UInt32, UInt32, Memory`1)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public int Read(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer
)
```

Parameters

- **indexGroup**
  Type: `System.UInt32`

- **indexOffset**
  Type: `System.UInt32`

- **readBuffer**
  Type: `Memory`

Return Value

Type: `Int32`

Reference

- AdsClient Class [154]
- Read Overload [233]
- TwinCAT.Ads Namespace [151]

### 6.2.1.2.21 AdsClient.ReadAny Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAny.T.(UInt32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, UInt32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, UInt32, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
</tbody>
</table>
## AdsClient.ReadAny<T>.T. Method (UInt32)

Reads data synchronously from an ADS device and writes it to an object.

**Namespace:** [TwinCAT.Ads](#)

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**
```csharp
public T ReadAny<T>(
    uint variableHandle
)
```

### Parameters

- **variableHandle**
  - Type: System.UInt32
  - Handle of the ADS variable.

### Type Parameters

- **T**
  - The type of the value to read.

### Return Value

- **Type:** T
  - The value of the read symbol.

### Implements

- [IAdsAnyAccess.ReadAny.T.(UInt32)](#)

### Reference

- [AdsClient Class](#)
- [TwinCAT.Ads Namespace](#)

---

### AdsClient.ReadAny<T>.T. Method (UInt32, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.
Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public T ReadAny<T>(
    uint variableHandle,
    int[] args
)
```

Parameters

variableHandle Type: System.UInt32
Handle of the ADS variable.

args Type: System.Int32
Additional arguments.

Type Parameters

T The type of the value to read.

Return Value

Type: T
The value of the read symbol.

Implements

IAdsAnyAccess.ReadAny.T.(UInt32, .Int32.)

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsClient Class

ReadAny Overload

TwinCAT.Ads Namespace

AdsClient.ReadAny Method (UInt32, Type)

Reads data synchronously from an ADS device and writes it to an object.
Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Object ReadAny(
    uint variableHandle,
    Type type
)
```

Parameters

- variableHandle: Type: System.UInt32
  Handle of the ADS variable.
- type: Type: System.Type
  Type of the object to be read.

Return Value

Type: Object
The read object.

Implements

IAdsAnyAccess.ReadAny(UInt32, Type)

Reference

AdsClient Class
ReadAny Overload
TwinCAT.Ads Namespace

AdsClient.ReadAny.T. Method (UInt32, UInt32)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public T ReadAny<T>(
    uint indexGroup,
    uint indexOffset
)
```

Parameters

- indexGroup: Type: System.UInt32
  Index group of the ADS variable.
- indexOffset: Type: System.UInt32
  Index offset of the ADS variable.
Type Parameters

T

The type of the object to be read.

Return Value

Type: T
The read value.

Implements

IAdsAnyAccess.ReadAny.T(UInt32, UInt32) [710]

Reference

AdsClient Class [154]
ReadAny Overload [234]
TwinCAT.Ads Namespace [151]

AdsClient.ReadAny Method (UInt32, Type, Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public Object ReadAny(
    uint variableHandle,
    Type type,
    int[] args
)

Parameters

variableHandle

Type: System.UInt32
Handle of the ADS variable.

type

Type: System.Type
Type of the object to be read.

args

Type: System.Int32
Additional arguments.

Return Value

Type: Object
The read value.

Implements

IAdsAnyAccess.ReadAny(UInt32, Type, Int32.) [710]
Remarks
As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference
- AdsClient Class [154]
- ReadAny Overload [234]
- TwinCAT.Ads Namespace [151]

**AdsClient.ReadAny.T. Method (UInt32, UInt32, .Int32.)**
Reads data synchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public T ReadAny<T>(
    uint indexGroup,
    uint indexOffset,
    int[] args
)
```

**Parameters**
- **indexGroup**
  - Type: System.UInt32
  - Index group of the ADS variable.
- **indexOffset**
  - Type: System.UInt32
  - Index offset of the ADS variable.
- **args**
  - Type: .System.Int32.
  - Additional arguments.

**Type Parameters**
- **T**
  - The type of the object to be read.

**Return Value**
- Type: T
  - The read value.
Implements

IAdsAnyAccess.ReadAny.T(UInt32, UInt32, _Int32.) [711]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395].</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]
ReadAny Overload [234]

TwinCAT.Ads Namespace [151]

AdsClient.ReadAny Method (UInt32, UInt32, Type)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>indexGroup</td>
<td>System.UInt32</td>
<td>Index group of the ADS variable.</td>
</tr>
<tr>
<td>indexOffset</td>
<td>System.UInt32</td>
<td>Index offset of the ADS variable.</td>
</tr>
<tr>
<td>type</td>
<td>System.Type</td>
<td>Type of the object to be read.</td>
</tr>
</tbody>
</table>

Return Value

Type: Object
The read value.
Implements

IAdsAnyAccess.ReadAny(UInt32, UInt32, Type) [712]

Reference

AdsClient Class [154]

ReadAny Overload [234]

TwinCAT.Ads Namespace [151]

**AdsClient.ReadAny Method (UInt32, UInt32, Type, .Int32.)**

Reads data synchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args
)
```

**Parameters**

- `indexGroup` Type: System.UInt32
  Index group of the ADS variable.
- `indexOffset` Type: System.UInt32
  Index offset of the ADS variable.
- `type` Type: System.Type
  Type of the object to be read.
- `args` Type: .System.Int32.
  Additional arguments.

**Return Value**

Type: Object
The read value.

Implements

IAdsAnyAccess.ReadAny(UInt32, UInt32, Type, .Int32.) [713]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>
### Exception

<table>
<thead>
<tr>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientNotConnectedException [63]</td>
</tr>
</tbody>
</table>

### Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395].</td>
</tr>
<tr>
<td>string[]</td>
<td>Array</td>
</tr>
</tbody>
</table>

### Reference

- AdsClient Class [154]
- ReadAny Overload [234]
- TwinCAT.Ads Namespace [151]

### 6.2.1.2.22 AdsClient.ReadAnyAsync Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAnyAsync(UInt32, CancellationToken)</td>
<td>Reads data synchronously from an ADS device.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, CancellationToken)</td>
<td>Reads the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, UInt32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, UInt32, CancellationToken)</td>
<td>Reads the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadAnyAsync.T. (UInt32, UInt32, Int32, CancellationToken) [247]</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [249]</td>
<td>Reads the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, Int32, CancellationToken) [249]</td>
<td>Reads the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

**AdsClient.ReadAnyAsync.T. Method (UInt32, CancellationToken)**

Reads data synchronously from an ADS device.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint variableHandle,
    CancellationToken cancel
)
```

**Parameters**

- **variableHandle**
  - Type: `System.UInt32`
  - The variable/symbol handle.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.

**Type Parameters**

- **T**
  - The Type of the value to be read.

**Return Value**

Type: `Task<ResultValue<T>>`

A task that represents the asynchronous read operation. The `ResultAnyValue` parameter contains the read value (Value [1000]) and the `ErrorCode` after execution.
Implements
IAdsAnyAccess.ReadAnyAsync<T).(UInt32, CancellationToken) [715]

Remarks
As object types only primitive types are supported.

Reference
AdsClient Class [154]
ReadAnyAsync Overload [242]
TwinCAT.Ads Namespace [151]

AdsClient.ReadAnyAsync<T>. Method (UInt32, Int32, CancellationToken)
Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax
C#
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint variableHandle,
    int[] args,
    CancellationToken cancel
)

Parameters
variableHandle Type: System.UInt32
The variable handle.
args Type: System.Int32
Additional arguments.
cancel Type: System.Threading.CancellationToken
The cancellation token.

Type Parameters
T Type of the object to be read

Return Value
Type: Task<ResultValue<T>>.T..
A task that represents the asynchronous read operation. The ResultValue.TValue. [1029] parameter contains the read value (Value [1032]) and the ErrorCode [992] after execution.

Implements
IAdsAnyAccess.ReadAnyAsync<T>(UInt32, Int32, CancellationToken) [715]
Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

ReadAnyAsync Overload [242]

TwinCAT.Ads Namespace [151]

**AdsClient.ReadAnyAsync Method (UInt32, Type, CancellationToken)**

Reads the value of an Anytype (Primitive type) asynchronously.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public Task<ResultAnyValue> ReadAnyAsync(  
    uint variableHandle,  
    Type type,  
    CancellationToken cancel
)
```

**Parameters**

- **variableHandle**  
  Type: System.UInt32  
  The variable handle.

- **type**  
  Type: System.Type  
  The type as AnyType (primitive types).

- **cancel**  
  Type: System.Threading.CancellationToken  
  The cancellation token.

**Return Value**

Type: Task<ResultAnyValue>[998].  
A task that represents the asynchronous ‘ReadState’ operation. The ResultAnyValue [998] parameter contains the value Value [1000] and the ErrorCode [992] of the ADS communication after execution.

**Implements**

IAdsAnyAccess.ReadAnyAsync(UInt32, Type, CancellationToken) [716]
AdsClient.ReadAnyAsync.T. Method (UInt32, UInt32, CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint indexGroup,
    uint indexOffset,
    CancellationToken cancel
)
```

**Parameters**

- **indexGroup**
  - Type: System.UInt32
  - Index group of the ADS variable.

- **indexOffset**
  - Type: System.UInt32
  - Index offset of the ADS variable.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

**Type Parameters**

- **T**

**Return Value**

- **Type:** Task<ResultValue<T>>
  - The asynchronous result.

**Return Value**

- **Type:** Task<ResultValue<T>>
  - A task that represents the asynchronous read operation. The ResultValue.TValue parameter contains the read value (Value) and the ErrorCode parameter contains the error code (Error) after execution.

**Implements**

- IAdsAnyAccess.ReadAnyAsync.T.(UInt32, UInt32, CancellationToken)

**Reference**

- AdsClient Class
- ReadAnyAsync Overload
**TwinCAT.Ads Namespace** [› 151]

**AdsClient.ReadAnyAsync Method (UInt32, Type, .Int32., CancellationToken)**

Reads the value of an Anytype (Primitive type) asynchronously.

**Namespace:** TwinCAT.Ads [› 151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Task<ResultAnyValue> ReadAnyAsync(
    uint variableHandle,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

**Parameters**

- **variableHandle**
  - Type: System.UInt32
  - The variable handle.

- **type**
  - Type: System.Type
  - The type as AnyType (primitive types).

- **args**
  - Type: System.Int32.
  - The type arguments (AnyType)

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

**Return Value**

Type: Task<ResultAnyValue> [› 998].

A task that represents the asynchronous 'ReadState' operation. The ResultAnyValue [› 998] parameter contains the value Value [› 1000] and the ErrorCode [› 992] of the ADS communication after execution.

**Implements**

IAdsAnyAccess.ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [› 718]

**Reference**

- AdsClient Class [› 154]
- ReadAnyAsync Overload [› 242]
- TwinCAT.Ads Namespace [› 151]

**AdsClient.ReadAnyAsync.T. Method (UInt32, UInt32, .Int32., CancellationToken)**

Reads data asynchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads [› 151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint indexGroup,
    uint indexOffset,
    int[] args,
    CancellationToken cancel)
```

Parameters

indexGroup  
Type: System.UInt32  
Index group of the ADS variable.

indexOffset  
Type: System.UInt32  
Index offset of the ADS variable.

args  
Type: System.Int32.  
Additional arguments.

cancel  
Type: System.Threading.CancellationToken  
The cancellation token.

Type Parameters

T  
The type of the result value.

Return Value

Type: Task<ResultValue<T>>.  
A task that represents the asynchronous read operation. The ResultValue.TValue parameter contains the read value (Value) and the ErrorCode after execution.

Implements

IAdsAnyAccess.ReadAnyAsync<T>(UInt32, UInt32, Int32, CancellationToken)

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

ReadAnyAsync Overload [242]

TwinCAT.Ads Namespace [151]
AdsClient.ReadAnyAsync Method (UInt32, UInt32, Type, CancellationToken)

Reads the value of an Anytype (Primitive type) asynchronously.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
public Task<ResultAnyValue> ReadAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Type type,
    CancellationToken cancel
)
```

**Parameters**

- **indexGroup**
  - Type: System.UInt32
  - The index group.

- **indexOffset**
  - Type: System.UInt32
  - The index offset.

- **type**
  - Type: System.Type
  - The type as AnyType (primitive types).

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

**Return Value**

- **Type:** Task<ResultAnyValue>

  A task that represents the asynchronous 'ReadState' operation. The ResultAnyValue parameter contains the value Value and the ErrorCode of the ADS communication after execution.

**Implements**

IAdsAnyAccess.ReadAnyAsync(UInt32, UInt32, Type, CancellationToken)

**Reference**

AdsClient Class

ReadAnyAsync Overload

TwinCAT.Ads Namespace

---

**AdsClient.ReadAnyAsync Method (UInt32, UInt32, Type, .Int32., CancellationToken)**

Reads the value of an Anytype (Primitive type) asynchronously.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
public Task<ResultAnyValue> ReadAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args,
    CancellationToken cancel)
```

Parameters

- **indexGroup**
  - Type: `System.UInt32`
  - The index group.
- **indexOffset**
  - Type: `System.UInt32`
  - The index offset.
- **type**
  - Type: `System.Type`
  - The type as AnyType (primitive types).
- **args**
  - Type: `System.Int32`
  - The type arguments (AnyType)
- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.

Return Value

Type: `Task<ResultAnyValue>`.

A task that represents the asynchronous 'ReadState' operation. The `ResultAnyValue` parameter contains the value `Value` and the `ErrorCode` of the ADS communication after execution.

Implements

`IAdsAnyAccess.ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken)`

Reference

- `AdsClient Class` [154]
- `ReadAnyAsync Overload` [242]
- `TwinCAT.Ads Namespace` [151]

6.2.1.2.23 AdsClient.ReadAnyString Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="Image" alt="ReadAnyString" /> ReadAnyString(UInt32, Int32, Encoding) [251]</td>
<td>Reads a string from the specified symbol/variable.</td>
</tr>
<tr>
<td><img src="Image" alt="ReadAnyString" /> ReadAnyString(UInt32, UInt32, Int32, Encoding) [251]</td>
<td>Reads as string from a specified address.</td>
</tr>
</tbody>
</table>
**AdsClient.ReadAnyString Method (UInt32, Int32, Encoding)**

Reads a string from the specified symbol/variable.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

C#

```csharp
public string ReadAnyString(
    uint variableHandle,
    int len,
    Encoding encoding
)
```

### Parameters

- **variableHandle**
  - Type: System.UInt32
  - The variable handle.

- **len**
  - Type: System.Int32
  - The length.

- **encoding**
  - Type: System.Text.Encoding
  - The encoding.

### Return Value

Type: String

The string value.

### Implements

ADSAnyAccess.ReadAnyString(UInt32, Int32, Encoding) [722]

### Reference

- AdsClient Class [154]
- ReadAnyString Overload [250]
- TwinCAT.Ads Namespace [151]

---

**AdsClient.ReadAnyString Method (UInt32, UInt32, Int32, Encoding)**

Reads a string from a specified address.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

---

**Reference**

- AdsClient Class [154]
- ReadAnyString Overload [250]
- TwinCAT.Ads Namespace [151]
Syntax

C#

```csharp
public string ReadAnyString(
    uint indexGroup,
    uint indexOffset,
    int len,
    Encoding encoding
)
```

Parameters

- `indexGroup` Type: `System.UInt32`  
The index group.
- `indexOffset` Type: `System.UInt32`  
The index offset.
- `len` Type: `System.Int32`  
The string length to be read.
- `encoding` Type: `System.Text.Encoding`  
The encoding.

Return Value

Type: `String`  
`System.String`.

Implements

`IAdsAnyAccess.ReadAnyString(UInt32, UInt32, Int32, Encoding)`

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ObjectDisposedException</code></td>
<td></td>
</tr>
<tr>
<td><code>ClientNotConnectedExcep tion</code></td>
<td>[63]</td>
</tr>
</tbody>
</table>

Reference

- `AdsClient Class` [154]
- `ReadAnyString Overload` [250]
- `TwinCAT.Ads Namespace` [151]
## 6.2.1.2.24 AdsClient.ReadAnyStringAsync Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) (253)</td>
<td>Reads a string asynchronously from the specified symbol/variable.</td>
</tr>
<tr>
<td>ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) (254)</td>
<td>Reads a string from a specified address asynchronously.</td>
</tr>
</tbody>
</table>

### Reference

- AdsClient Class ([154])
- TwinCAT.Ads Namespace ([151])

### AdsClient.ReadAnyStringAsync Method (UInt32, Int32, Encoding, CancellationToken)

Reads a string asynchronously from the specified symbol/variable.

**Namespace:** TwinCAT.Ads ([151])

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public Task<ResultAnyValue> ReadAnyStringAsync( 
    uint variableHandle,
    int len,
    Encoding encoding,
    CancellationToken cancel
)
```

**Parameters**

- `variableHandle` Type: System.UInt32
  - The variable handle.
- `len` Type: System.Int32
  - The length.
- `encoding` Type: System.Text.Encoding
  - The encoding.
- `cancel` Type: System.Threading.CancellationToken
  - The cancellation token.
Return Value

Type: Task<ResultAnyValue> [998].
A task that represents the asynchronous read operation. The ResultAnyValue [998] parameter contains the read string (Value [1000]) and the ErrorCode [992] after execution.

Implements

IAdsAnyAccess.ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [724]

Reference

AdsClient Class [154]

ReadAnyStringAsync Overload [253]

TwinCAT.Ads Namespace [151]

AdsClient.ReadAnyStringAsync Method (UInt32, UInt32, Int32, Encoding, CancellationToken)

Reads a string from a specified address asynchronously.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultAnyValue> ReadAnyStringAsync(
    uint indexGroup,
    uint indexOffset,
    int len,
    Encoding encoding,
    CancellationToken cancel)
```

Parameters

indexGroup
Type: System.UInt32
The index group.

indexOffset
Type: System.UInt32
The index offset.

len
Type: System.Int32
The string length to be read.

encoding
Type: System.Text.Encoding
The encoding.

cancel
Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultAnyValue> [998].
A task that represents the asynchronous read operation. The ResultAnyValue [998] parameter contains the read value (Value [1000]) and the ErrorCode [992] after execution.
Implement

IAdsAnyAccess.ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [725]

Reference

AdsClient Class [154]

ReadAnyStringAsync Overload [253]

TwinCAT.Ads Namespace [151]

6.2.1.2.25 AdsClient.ReadAsync Method

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="255" alt="ReadAsync(UInt32, Memory, Void)" /></td>
<td>AdsClient.ReadAsync(UInt32, Memory, Void) ![255]</td>
</tr>
<tr>
<td><img src="256" alt="ReadAsync(UInt32, UInt32, Memory, Void)" /></td>
<td>AdsClient.ReadAsync(UInt32, UInt32, Memory, Void) ![256]</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

AdsClient.ReadAsync Method (UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public Task<ResultRead> ReadAsync(
    uint variableHandle,
    Memory readBuffer,
    void cancel
)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>variableHandle</td>
<td>System.UInt32</td>
</tr>
<tr>
<td>readBuffer</td>
<td>Memory</td>
</tr>
<tr>
<td>cancel</td>
<td>System.Void</td>
</tr>
</tbody>
</table>
Return Value

Type: Task<ResultRead [1008].

Reference

AdsClient Class [154]
ReadAsync Overload [255]
TwinCAT.Ads Namespace [151]

AdsClient.ReadAsync Method (UInt32, UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public Task<ResultRead> ReadAsync(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void cancel
)

Parameters

indexGroup Type: System.UInt32
indexOffset Type: System.UInt32
readBuffer Type: Memory
cancel Type: System.Void

Return Value

Type: Task<ResultRead [1008].

Reference

AdsClient Class [154]
ReadAsync Overload [255]
TwinCAT.Ads Namespace [151]

6.2.1.2.26 AdsClient.ReadDataType Method

Call this method to obtain information about the specified data type.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
# TwinCAT.Ads Namespaces

## C#

```csharp
public IDataType ReadDataType(
    string typeName
)
```

### Parameters

- **typeName**
  - Type: `System.String`
  - Name of the data type (without namespace)

### Return Value

- Type: `IDataType`
  - An containing the requested type.

### Implements

- `IAdsSymbolicAccess.ReadDataType(String)`

### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ArgumentOutOfRangeException</code></td>
<td>typeName</td>
</tr>
<tr>
<td><code>ObjectDisposedException</code></td>
<td></td>
</tr>
<tr>
<td><code>ClientNotConnectedException</code></td>
<td></td>
</tr>
</tbody>
</table>

### Reference

- `AdsClient Class [154]`
- `TwinCAT.Ads Namespace [151]`
- `IAdsSymbolicAccess.TryReadDataType(String, IDataType.) [952]`
- `IAdsSymbolicAccess.ReadDataTypeAsync(String, CancellationToken) [943]`

### 6.2.1.2.27 AdsClient.ReadDataTypeAsync Method

Read data type as an asynchronous operation.

### Namespace: TwinCAT.Ads [151]

### Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

## Syntax

```csharp
public Task<ResultValue<IDataType>> ReadDataTypeAsync(
    string typeName,
    CancellationToken cancel
)
```
Parameters

typeName                      Type: System.String  
Name of the data type.

cancel                       Type: System.Threading.CancellationToken  
The cancel token.

Return Value

Type: Task.ResultValue[1029]!IDataType[1986].
A task that represents the asynchronous 'ReadDataType' operation. The ResultValue.TValue parameter contains the read value (Value[1032]) and the ErrorCode[992] after execution.

Implements

IAdsSymbolicAccess.ReadDataTypeAsync(String, CancellationToken)[943]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td>typeName</td>
</tr>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>63</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class[154]
TwinCAT.Ads Namespace[151]
IAdsSymbolicAccess.ReadDataType(String)[942]
IAdsSymbolicAccess.TryReadDataType(String, IDataType. )[952]

6.2.1.2.28 AdsClient.ReadDeviceInfo Method

Reads the identification and version number of an ADS server.

Namespace: TwinCAT.Ads[151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffddca3e72bc0ea15da1c14

Syntax

C#  
publicDeviceInfo ReadDeviceInfo()  

Return Value

Type: DeviceInfo[698]  
DeviceInfo struct containing the name of the device and the version information.
Implements

IAdsConnection.ReadDeviceInfo. [795]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>[63]</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

6.2.1.2.29 AdsClient.ReadDeviceInfoAsync Method

Reads the identification and version number of an ADS server.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultDeviceInfo> ReadDeviceInfoAsync(
    CancellationToken cancel
)
```

Parameters

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultDeviceInfo> [1001].

Implements

IAdsConnection.ReadDeviceInfoAsync(CancellationToken). [796]

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]
6.2.1.2.30 AdsClient.ReadState Method

Reads the ADS status and the device status from an ADS server.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public StateInfo ReadState()
```

**Return Value**

Type: StateInfo

The ADS statue and device status.

**Implements**

IAdsStateProvider.ReadState.

**Reference**

AdsClient Class

TwinCAT.Ads Namespace

6.2.1.2.31 AdsClient.ReadStateAsync Method

Reads the ADS status and the device status from an ADS server.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Task<ResultReadDeviceState> ReadStateAsync(
    CancellationToken cancel
)
```

**Parameters**

cancel Type: System.Threading.CancellationToken

The cancellation token

**Return Value**

Type: Task<ResultReadDeviceState>

A task that represents the asynchronous 'ReadState' operation. The ResultReadDeviceState parameter contains the state (State) as long as the ErrorCode of the ADS communication after execution.
6.2.1.2.32 AdsClient.ReadSymbol Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public IAdsSymbol ReadSymbol(
    string name
)

Parameters

name Type: System.String
Name of the symbol.

Return Value

Type: IAdsSymbol
A IAdsSymbol2 containing the requested symbol information or null if symbol could not be found.

Implements

IAdsSymbolicAccess.ReadSymbol(String) [943]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsErrorException [583]</td>
<td>Thrown when the ADS call fails.</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]

6.2.1.2.33 AdsClient.ReadSymbolAsync Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.
TwinCAT.Ads Namespaces

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultValue<IAdsSymbol>> ReadSymbolAsync(string name, CancellationToken cancel)
```

Parameters

- **name**
  - Type: `System.String`
  - Name of the symbol.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancel token.

Return Value

Type: `Task<ResultValue<IAdsSymbol>>`

A task that represents the asynchronous 'ReadSymbolInfo' operation. The `ResultValue.TValue` parameter contains the read value (Value) and the `ErrorCode` after execution.

Implements

- `IAdsSymbolicAccess.ReadSymbolAsync(String, CancellationToken)`

Reference

- AdsClient Class
- TwinCAT.Ads Namespace

6.2.1.2.34 AdsClient.ReadValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReadValue.T(String)</code> [263]</td>
<td>Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td><code>ReadValue(ISymbol)</code> [263]</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td><code>ReadValue.T(ISymbol)</code> [264]</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td><code>ReadValue(String, Type)</code> [265]</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
</tbody>
</table>

Reference

- AdsClient Class
- TwinCAT.Ads Namespace
**AdsClient.ReadValue<T>. Method (String)**

Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public T ReadValue<T>(
    string name
)
```

**Parameters**

- **name**
  - Type: System.String
  - Name of the ADS symbol.

**Type Parameters**

- **T**
  - The value type

**Return Value**

- **Type:** T
  - Value of the symbol

**Implements**

- IAdsSymbolicAccess.ReadValue<T>(String)

**Reference**

- AdsClient Class
- ReadValue Overload
- TwinCAT.Ads Namespace

**AdsClient.ReadValue Method (ISymbol)**

Reads the value of a symbol and returns it as an object.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Object ReadValue(
    ISymbol symbol
)
```
Parameters

symbol Type: TwinCAT.TypeSystem.ISymbol [2176]
The symbol that should be read.

Return Value

Type: Object
The value of the symbol as an object.

Implements

IAdsSymbolicAccess.ReadValue(ISymbol) [946]

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'. Structs are not supported.

Reference

AdsClient Class [154]
ReadWrite Overload [262]
TwinCAT.Ads Namespace [151]

AdsClient.ReadValue<T>. Method (ISymbol)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public T ReadValue<T>(
    ISymbol symbol
)
```

Parameters

symbol Type: TwinCAT.TypeSystem.ISymbol [2176]
The symbol that should be read.

Type Parameters

T The value type.

Return Value

Type: T
The value of the symbol.
Implements
IAdsSymbolicAccess.ReadValue.T.(ISymbol) [946]

Remarks
Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference
AdsClient Class [154]
ReadValue Overload [262]
TwinCAT.Ads Namespace [151]

AdsClient.ReadValue Method (String, Type)

Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Object ReadValue(
    string name,
    Type type
)
```

Parameters

name Type: System.String
      Name of the ADS symbol.

type Type: System.Type
       Managed type of the ADS symbol.

Return Value

Type: Object
Value of the symbol

Implements
IAdsSymbolicAccess.ReadValue(String, Type) [947]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### Exception | Condition
---|---
ClientNotConnectedException [63] | 

### Reference
- AdsClient Class [154]
- ReadValue Overload [262]
- TwinCAT.Ads Namespace [151]

### 6.2.1.2.35 AdsClient.ReadValueAsync Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsClient.ReadValueAsync.TValue.(String, CancellationToken) [266]</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>AdsClient.ReadValueAsync(ISymbol, CancellationToken) [267]</td>
<td>Reads the value of a symbol asynchronously and returns it as an object. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.</td>
</tr>
<tr>
<td>AdsClient.ReadValueAsync.TValue.(ISymbol, CancellationToken) [268]</td>
<td>Reads the value of a symbol asynchronously and returns it as an object.</td>
</tr>
<tr>
<td>AdsClient.ReadValueAsync.String, Type, CancellationToken) [269]</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
</tbody>
</table>

### Reference
- AdsClient Class [154]
- TwinCAT.Ads Namespace [151]

### AdsClient.ReadValueAsync.TValue. Method (String, CancellationToken)

Reads the value of a symbol asynchronously.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public Task<ResultValue<TValue>> ReadValueAsync<TValue>(
    string name,
    CancellationToken cancel
)
```

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>System.String</td>
<td>Name of the ADS symbol.</td>
</tr>
<tr>
<td>cancel</td>
<td>System.Threading.CancellationToken</td>
<td>The cancel token.</td>
</tr>
</tbody>
</table>

Type Parameters

| TValue | The value type.                           |

Return Value

Type: Task<ResultValue[TValue].TValue>
A task that represents the asynchronous read operation. The ResultAnyValue.ErrorCode parameter contains the read value (Value) and the ErrorCode after execution.

Remarks

The parameter type must have the same layout as the ADS symbol.

Reference

<table>
<thead>
<tr>
<th>Reference</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsClient Class [154]</td>
<td></td>
</tr>
<tr>
<td>ReadValueAsync Overload [266]</td>
<td></td>
</tr>
<tr>
<td>TwinCAT.Ads Namespace [151]</td>
<td></td>
</tr>
</tbody>
</table>

**AdsClient.ReadValueAsync Method (ISymbol, CancellationToken)**

Reads the value of a symbol asynchronously and returns it as an object. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultAnyValue> ReadValueAsync(
    ISymbol symbol,
    CancellationToken cancel
)
```

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbol</td>
<td>TwinCAT.TypeSystem.ISymbol [2176]</td>
<td>The symbol that should be read.</td>
</tr>
</tbody>
</table>
cancel  
Type: System.Threading.CancellationToken  
The cancel token.

Return Value
Type: Task<ResultAnyValue>.  
The ResultAnyValue as customized task object.

Implements
IAdsSymbolicAccess.ReadValueAsync(ISymbol, CancellationToken).  
Reference
AdsClient Class.  
ReadValueAsync Overload.  
TwinCAT.Ads Namespace.

AdsClient.ReadValueAsync.TValue. Method (ISymbol, CancellationToken)
Reads the value of a symbol asynchronously and returns it as an object.

Namespace: TwinCAT.Ads.  
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#  
public Task<ResultValue<TValue>> ReadValueAsync<TValue>(  
    ISymbol symbol,  
    CancellationToken cancel  
)

Parameters
symbol  
Type: TwinCAT.TypeSystem.ISymbol.  
The symbol that should be read.
cancel  
Type: System.Threading.CancellationToken.  
The cancel token.

Type Parameters
TValue  
The value type.

Return Value
Type: Task<ResultValue>.  
A task that represents the asynchronous read operation. The ResultAnyValue parameter contains the read value (Value) and the ErrorCode after execution.
TwinCAT.Ads Namespaces

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

AdsClient Class [154]
ReadValueAsync Overload [266]
TwinCAT.Ads Namespace [151]

AdsClient.ReadValueAsync Method (String, Type, CancellationToken)

Reads the value of a symbol asynchronously.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultAnyValue> ReadValueAsync(
    string name,
    Type type,
    CancellationToken cancel
)
```

Parameters

name Type: System.String
Name of the ADS symbol.

type Type: System.Type
Managed type of the ADS symbol.

cancel Type: System.Threading.CancellationToken
The cancel token.

Return Value

Type: Task<ResultAnyValue> [998].
A task that represents the asynchronous read operation. The ResultAnyValue [998] parameter contains the read value (Value) and the ErrorCode [992] after execution.

Implements

IAdsSymbolicAccess.ReadValueAsync(String, Type, CancellationToken) [951]
**Remarks**
The parameter type must have the same layout as the ADS symbol.

**Reference**
- AdsClient Class [154]
- ReadValueAsync Overload [266]
- TwinCAT.Ads Namespace [151]

### 6.2.1.2.36 AdsClient.ReadWrite Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReadWrite(UInt32, Memory, Void)</code></td>
<td>[270]</td>
</tr>
<tr>
<td><code>ReadWrite(UInt32, UInt32, Memory, Void)</code></td>
<td>[271]</td>
</tr>
</tbody>
</table>

**Reference**
- AdsClient Class [154]
- TwinCAT.Ads Namespace [151]

**AdsClient.ReadWrite Method (UInt32, Memory`1, Void)**

*Namespace:* TwinCAT.Ads [151]
*Assembly:* TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public int ReadWrite(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer
)
```

**Parameters**

- **variableHandle** Type: System.UInt32
- **readBuffer** Type: Memory
- **writeBuffer** Type: System.Void
Return Value
Type: Int32

Reference
AdsClient Class [154]
ReadWrite Overload [270]
TwinCAT.Ads Namespace [151]

AdsClient.ReadWrite Method (UInt32, UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public int ReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer
)

Parameters

indexGroup Type: System.UInt32
indexOffset Type: System.UInt32
readBuffer Type: Memory
writeBuffer Type: System.Void

Return Value
Type: Int32

Reference
AdsClient Class [154]
ReadWrite Overload [270]
TwinCAT.Ads Namespace [151]
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>➔</td>
<td>ReadWriteAsync(UInt32, Memory, Void, Byte) [272]</td>
</tr>
<tr>
<td>➔</td>
<td>ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte) [273]</td>
</tr>
</tbody>
</table>

### Reference

- AdsClient Class [154]
- TwinCAT.Ads Namespace [151]

### AdsClient.ReadWriteAsync Method (UInt32, Memory<1>, Void, Byte)

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

#### C#

```csharp
public Task<ResultReadWrite> ReadWriteAsync(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer,
    byte cancel
)
```

**Parameters**

- `variableHandle`: Type: `System.UInt32`
- `readBuffer`: Type: `Memory`
- `writeBuffer`: Type: `System.Void`
- `cancel`: Type: `System.Byte`

**Return Value**

Type: `Task<ResultReadWrite>` [1019].

### Reference

- AdsClient Class [154]
- ReadWriteAsync Overload [272]
**TwinCAT.Ads Namespace**

**AdsClient.ReadWriteAsync Method (UInt32, UInt32, Memory`1, Void, Byte)**

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```
C#
public Task<ResultReadWrite> ReadWriteAsync(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte cancel
)
```

**Parameters**

- **indexGroup**
  
  Type: System.UInt32

- **indexOffset**

  Type: System.UInt32

- **readBuffer**

  Type: Memory

- **writeBuffer**

  Type: System.Void

- **cancel**

  Type: System.Byte

**Return Value**

Type: Task<ResultReadWrite>.

**Reference**

- AdsClient Class
- ReadWriteAsync Overload
- TwinCAT.Ads Namespace

### 6.2.1.2.38 AdsClient.RegisterAdsStateChangedAsync Method

Registers for AdsStateChanged events as an asynchronous operation.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
### Syntax

**C#**

```csharp
public Task<ResultAds> RegisterAdsStateChangedAsync(
    EventHandler<AdsStateChangedEventArgs> handler,
    CancellationToken cancel)
```

### Parameters

- **handler**
  Type: `System.EventHandler<AdsStateChangedEventArgs>`. The handler function to be registered for AdsStateChanged calls.

- **cancel**

### Return Value

Type: `Task<ResultAds>`. A task that represents the asynchronous 'RegisterAdsStateChanged' operation. The `ResultAds` parameter contains the state the `ErrorCode` of the ADS communication after execution.

### Implements

`IAdsStateProvider.RegisterAdsStateChangedAsync(EventHandler<AdsStateChangedEventArgs>, CancellationToken)`

### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

### Reference

- **AdsClient Class**
  [154]

- **TwinCAT.Ads Namespace**
  [151]

### 6.2.1.2.39 AdsClient/RegisterSymbolVersionChangedAsync Method

Registers an `AdsSymbolVersionChanged` event as an asynchronous operation.

**Namespace:** `TwinCAT.Ads` [151]

**Assembly:** `TwinCAT.Ads.dll` (in `TwinCAT.Ads`)

**Version:** 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public Task<ResultAds> RegisterSymbolVersionChangedAsync(
    EventHandler<AdsSymbolVersionChangedEventArgs> handler,
    CancellationToken cancel)
```

### Parameters

- **handler**
cancel

Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task.ResultAds [989].
A task that represents the asynchronous 'RegisterSymbolVersionChanged' operation. The ResultAds [989] parameter contains the value ErrorCode [992] of the ADS communication after execution.

Implements

IAdsSymbolChangedProvider.RegisterSymbolVersionChangedAsync(EventHandler.AdsSymbolVersionChangeEventArgs., CancellationToken) [935]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

6.2.1.2.40 AdsClient.TryAddDeviceNotification Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [338] event.</td>
</tr>
<tr>
<td></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [866] event.</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]
AdsClient.TryAddDeviceNotification Method (String, Int32, NotificationSettings, Object, UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

**Namespace:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public AdsErrorCode TryAddDeviceNotification(
    string symbolPath,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    out uint handle
)
```

**Parameters**

- **symbolPath**
  - Type: System.String
  - The symbol/instance path of the ADS variable.

- **dataSize**
  - Type: System.Int32
  - Maximum amount of data in bytes to receive with this ADS Notification.

- **settings**
  - Type: TwinCAT.Ads.NotificationSettings
  - The notification settings.

- **userData**
  - Type: System.Object
  - This object can be used to store user specific data.

- **handle**
  - Type: System.UInt32
  - The notification handle.

**Return Value**

Type: AdsErrorCode

The ADS ErrorCode.

**Implements**

IAdsNotifications.TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.)

**Remarks**

The `dataSize` parameter defines the amount of bytes, that will be attached to the AdsNotification event as value. Because notifications allocate TwinCAT system resources, a complementary call to `TryDeleteDeviceNotification(UInt32)` should always be called when the notification is not used anymore.

**Reference**

- AdsClient Class
- TryAddDeviceNotification Overload
AdsClient.TryAddDeviceNotification Method (UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AdsErrorCode TryAddDeviceNotification(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    out uint handle
)
```

**Parameters**

- **indexGroup**
  Type: `System.UInt32`
  The index group number of the requested ADS service.

- **indexOffset**
  Type: `System.UInt32`
  The index offset number of the requested ADS service.

- **dataSize**
  Type: `System.Int32`
  Maximum amount of data in bytes to receive with this ADS Notification.

- **settings**
  Type: `TwinCAT.Ads.NotificationSettings` [979]
  The Notification settings.

- **userData**
  Type: `System.Object`
  This object can be used to store user specific data (tag data)

- **handle**
  Type: `System.UInt32`
  The notification handle.

**Return Value**

Type: `AdsErrorCode` [575]
The ADS error code.

**Implements**

`IAdsNotifications.TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)` [861]
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException [63]</td>
<td></td>
</tr>
</tbody>
</table>

### Remarks

The `dataSize` parameter defines the amount of bytes, that will be attached to the `AdsNotification` [338] as value. Because notifications allocate TwinCAT system resources, a complementary call to `TryDeleteDeviceNotification(UInt32) [864]` should always called when the notification is not used anymore.

### Reference

- AdsClient Class [154]
- TryAddDeviceNotification Overload [275]
- TwinCAT.Ads Namespace [151]
- IAdsNotifications.TryDeleteDeviceNotification(UInt32) [864]
- IAdsNotifications.AdsNotification [866]
- IAdsNotifications.AdsNotificationError [867]
- AddDeviceNotification Overload [844]
- TryAddDeviceNotification Overload [859]
- AddDeviceNotificationAsync Overload [847]

### 6.2.1.2.41 AdsClient.TryAddDeviceNotificationEx Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32,, UInt32) [279]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotificationEx[340]</code> event.</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32,, UInt32) [280]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotificationEx[868]</code> event.</td>
</tr>
</tbody>
</table>
AdsClient.TryAddDeviceNotificationEx Method (String, NotificationSettings, Object, Type, .Int32., UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [154] event.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryAddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    out uint handle)
```

Parameters

- **symbolPath**
  Type: `System.String`
  Symbol/Instance path of the ADS variable.

- **settings**
  Type: `TwinCAT.Ads.NotificationSettings [283]`
  The Notification settings.

- **userData**
  Type: `System.Object`
  This object can be used to store user specific data (tag data)

- **type**
  Type: `System.Type`
  Type of the object stored in the event argument (‘AnyType’)

- **args**
  Type: `System.Int32`
  Additional arguments (for ‘AnyType’)

- **handle**
  Type: `System.UInt32`
  The notification handle

Return Value

Type: `AdsErrorCode [154]`

The ADS error code.

Implements

`IAdsNotifications.TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [283]`

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to `TryDeleteDeviceNotification(UInt32) [283]` should always called when the notification is not used anymore.
Reference

AdsClient Class [154]
TryAddDeviceNotificationEx Overload [278]
TwinCAT.Ads Namespace [151]
AdsClient AdsNotificationEx [340]
AdsClient.DeleteDeviceNotification(UInt32) [216]
AddDeviceNotificationEx Overload [850]
AddDeviceNotificationExAsync Overload [855]
TryAddDeviceNotificationEx Overload [862]

**AdsClient.TryAddDeviceNotificationEx** Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AdsErrorCode TryAddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type anyType,
    int[] args,
    out uint handle
)
```

**Parameters**

- **indexGroup**
  Type: `System.UInt32`
  Contains the index group number of the requested ADS service.

- **indexOffset**
  Type: `System.UInt32`
  Contains the index offset number of the requested ADS service.

- **settings**
  Type: `TwinCAT.Ads.NotificationSettings` [979]
  The Notification settings.

- **userData**
  Type: `System.Object`
  This object can be used to store user specific data (tag data)

- **anyType**
  Type: `System.Type`
  Type of the object stored in the event argument ('AnyType')

- **args**
  Type: `.System.Int32.`
  The 'AnyType' arguments.

- **handle**
  Type: `System.UInt32`
  The notification handle.
Return Value

Type: AdsErrorCode [575]
The ADS Error code.

Implements

IAdsNotifications.TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, _Int32_,
UInt32.) [863]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException [63]</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive types (AnyType) are supported by this method. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [857] should always called when the notification is not used anymore.

Reference

AdsClient Class [154]
TryAddDeviceNotificationEx Overload [278]

TwinCAT.Ads Namespace [151]
IAdsNotifications.DeleteDeviceNotification(UInt32) [857]

IAdsNotifications.AdsNotificationEx [868]
IAdsNotifications.AdsNotificationError [867]

AddDeviceNotificationEx Overload [850]
TryAddDeviceNotificationEx Overload [862]
AddDeviceNotificationExAsync Overload [855]

6.2.1.2.42 AdsClient.TryCreateVariableHandle Method

Determines the Symbol handle by its instance path synchronously.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
public AdsErrorCode TryCreateVariableHandle(
    string symbolPath,
    out uint variableHandle
)
```

Parameters

- `symbolPath` Type: `System.String`  
  SymbolName / InstancePath.
- `variableHandle` Type: `System.UInt32`  
  The symbols handle.

Return Value

Type: `AdsErrorCode` [575]  
The ADS error code.

Implements

- `IAdsHandle.TryCreateVariableHandle(String, UInt32.)` [834]

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this `TryCreateVariableHandle(String, UInt32.)` is the `TryDeleteVariableHandle(UInt32)` [285].

Reference

- `AdsClient Class` [154]
- `TwinCAT.Ads Namespace` [151]
- `AdsClient.TryDeleteVariableHandle(UInt32)` [285]
- `AdsClient.CreateVariableHandleAsync(String, CancellationToken)` [216]
- `AdsClient.CreateVariableHandle(String)` [215]

6.2.1.2.43 AdsClient.TryDeleteDeviceNotification Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="283" alt="TryDeleteDeviceNotification(UInt32)" /></td>
<td>Deletes a registered notification.</td>
</tr>
<tr>
<td><img src="284" alt="TryDeleteDeviceNotification(UInt32, Int32)" /></td>
<td>Deletes a registered notification.</td>
</tr>
</tbody>
</table>
AdsClient.TryDeleteDeviceNotification Method (UInt32)

Deletes a registered notification.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public AdsErrorCode TryDeleteDeviceNotification(
    uint notificationHandle
)
```

**Parameters**

- **notificationHandle**
  - Type: System.UInt32
  - Notification handle.

**Return Value**

- Type: AdsErrorCode
  - The ADS error code.

**Implements**

- IAdsNotifications.TryDeleteDeviceNotification(UInt32)

**Remarks**

This is the complementary method to TryAddDeviceNotification Overload overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

**Reference**

- AdsClient Class
- TwinCAT.Ads Namespace
- TryDeleteDeviceNotification Overload
- TwinCAT.Ads Namespace
- AddDeviceNotification Overload
- IAdsNotifications.AdsNotification
- TryAddDeviceNotification Overload
- AddDeviceNotificationAsync Overload
AdsClient.TryDeleteDeviceNotification Method (UInt32, Int32)

Deletes a registered notification.

**Namespace:** TwinCAT.Ads [› 151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AdsErrorCode TryDeleteDeviceNotification(
    uint notificationHandle,
    int timeout
)
```

**Parameters**

- `notificationHandle`  
  Type: System.UInt32  
  Notification handle.

- `timeout`  
  Type: System.Int32  
  The timeout.

**Return Value**

Type: AdsErrorCode [› 575]  
The ADS error code.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException [› 63]</td>
<td></td>
</tr>
</tbody>
</table>

**Remarks**

This is the complementary method to TryAddDeviceNotification Overload [› 859] overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

**Reference**

- AdsClient Class [› 154]
- TryDeleteDeviceNotification Overload [› 282]
- TwinCAT.Ads Namespace [› 151]
- AddDeviceNotification Overload [› 844]
- IAdsNotifications.AdsNotification [› 866]
- TryAddDeviceNotification Overload [› 859]
- AddDeviceNotificationAsync Overload [› 847]
6.2.1.2.44  AdsClient.TryDeleteVariableHandle Method

Releases the specified symbol/variable handle synchronously.

Namespace:  TwinCAT.Ads [151]
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryDeleteVariableHandle(
    uint variableHandle
)
```

Parameters

variableHandle  Type: System.UInt32
Handle of the ADS variable

Return Value

Type: AdsErrorCode [575]
The ADS error code.

Implements

IAdsHandle.TryDeleteVariableHandle(UInt32) [835]

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this TryDeleteVariableHandle(UInt32) is the TryCreateVariableHandle(String, UInt32.) [281]

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]
AdsClient.TryCreateVariableHandle(String, UInt32.) [281]
AdsClient.DeleteVariableHandleAsync(UInt32, CancellationToken) [219]
AdsClient.DeleteVariableHandle(UInt32) [218]

6.2.1.2.45  AdsClient.TryInvokeRpcMethod Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ] TryInvokeRpcMethod(String, String, Object.)</td>
<td>Invokes the specified RPC Method</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]
AdsClient.TryCreateVariableHandle(String, UInt32.) [281]
AdsClient.DeleteVariableHandleAsync(UInt32, CancellationToken) [219]
AdsClient.DeleteVariableHandle(UInt32) [218]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object) [288]</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object) [290]</td>
<td>Invokes the rpc method.</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object) [291]</td>
<td>Tries the invoke RPC method.</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]

AdsClient.TryInvokeRpcMethod Method (String, String, Object, Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```c#
public AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object retValue
)
```

Parameters

- **symbolPath**
  - Type: `System.String`
  - The symbol path.
- **methodName**
  - Type: `System.String`
  - The method name.
inParameters
Type: System.Object.
The parameters.

retValue
Type: System.Object.
The return value of the RPC method as object.

Return Value
Type: AdsErrorCode
The ADS Error Code.

Implements
IAdsRpcInvoke.TryInvokeRpcMethod(String, String, Object, Object) [901]

Remarks
Because this overload doesn't provide any AnyTypeSpecifier specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples
The following sample shows how to call (Remote Procedures / Methods) within the PLC.

dynamic Tree Mode
class RpcCallIVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program
            // Call a Method that has the following signature (within MAIN Program)
            /* {attribute 'TcRpcEnable'}
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
                i1 : INT := 0;
                i2 : INT := 0;
            END_VAR
            */
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});
            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});
            //Browsing RpcMethods
            foreach(IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;
            }
        }
    }
}
foreach(IRpcMethodParameter parameter in method.Parameters)
{
    string parameterName = parameter.Name;
    string parameterType = parameter.TypeName;
    }
}}

Reference

AdsClient Class [154]

TryInvokeRpcMethod Overload [285]

TwinCAT.Ads Namespace [151]

AdsClient.TryInvokeRpcMethod Method (String, String, .Object., .Object.., Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object[] outParameters,
    out Object retValue)

Parameters

symbolPath Type: System.String
The symbol path.

methodName Type: System.String
The method name.

inParameters Type: System.Object
The parameters.

outParameters Type: System.Object
The out parameters.

retValue Type: System.Object
The return value of the RPC method as object.

Return Value

Type: AdsErrorCode [575]
The ADS Error Code.

Implements

IAdsRpcInvoke.TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [903]
Remarks

Because this overload doesn't provide any AnyTypeSpecifier[1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program
            // Call a Method that has the following signature (within MAIN Program)
            /*
             * {attribute 'TcRpcEnable'}
             * METHOD PUBLIC M_Add : INT
             * VAR_INPUT
             *     i1 : INT := 0;
             *     i2 : INT := 0;
             * END_VAR
             *
             * short result = (short)main.InvokeRpcMethod("M_Add", new object[] {{short) 3, (short) 4}});
             * // Call a Method that has no parameter and returns VOID
             * main.InvokeRpcMethod("M_Method1", new object[] {});
             * //Browsing RpcMethods
             * foreach(IRpcMethod method in main.RpcMethods)
             * {
             *     string methodName = method.Name;
             *     foreach(IRpcMethodParameter parameter in method.Parameters)
             *     {
             *         string parameterName = parameter.Name;
             *         string parameterType = parameter.TypeName;
             *     }
             * }
            */
    }
}

Reference

AdsClient Class[154]
TryInvokeRpcMethod Overload[285]
TwinCAT.Ads Namespace[151]
**AdsClient.TryInvokeRpcMethod Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.)**

Invokes the rpc method.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd6ca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters,
    out Object retValue
)
```

**Parameters**

- `symbolPath` Type: System.String
  The symbol.
- `methodName` Type: System.String
  Name of the method.
- `inParameters` Type: System.Object
  The parameters.
- `outSpecifiers` Type: TwinCAT.TypeSystem.AnyTypeSpecifier
  The out specifiers (specifying the out types) or NULL.
- `retSpecifier` Type: TwinCAT.TypeSystem.AnyTypeSpecifier
  The ret specifier (specifying the return value) or NULL.
- `outParameters` Type: System.Object
  The out parameters.
- `retValue` Type: System.Object
  The return value of the RPC method.

**Return Value**

Type: AdsErrorCode

AdsErrorCode.

**Implements**

IAdsRpcInvoke.TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.. Object.)

**Remarks**

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.
Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

### Dynamic Tree Mode

```csharp
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /* {attribute 'TcRpcEnable'}
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
                i1 : INT := 0;
                i2 : INT := 0;
            END_VAR
            */
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short)3, (short)4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});

            // Browsing RpcMethods
            foreach (IRpcMethod method in main.RpcMethods)
            {
                //string methodName = method.Name;
                //foreach (IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}
```

### Reference

- AdsClient Class [154]
- TryInvokeRpcMethod Overload [285]
- TwinCAT.Ads Namespace [151]

**AdsClient.TryInvokeRpcMethod Method**

`IRpcCallableInstance, IRpcMethod, .Object, .AnyTypeSpecifier, AnyTypeSpecifier, .Object.., Object.)`

Tries the invoke RPC method.
Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryInvokeRpcMethod(
    IRpcCallableInstance symbol,
    IRpcMethod rpcMethod,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpec,
    AnyTypeSpecifier returnSpec,
    out Object[] outParameters,
    out Object returnValue
)
```

Parameters

- **symbol**
  - Type: TwinCAT.TypeSystem.IRpcCallableInstance
  - The symbol.

- **rpcMethod**
  - Type: TwinCAT.TypeSystem.IRpcMethod
  - The RPC method.

- **inParameters**
  - Type: System.Object
  - The in parameters.

- **outSpec**
  - Type: TwinCAT.TypeSystem.AnyTypeSpecifier
  - The out spec.

- **returnSpec**
  - Type: TwinCAT.TypeSystem.AnyTypeSpecifier
  - The return spec.

- **outParameters**
  - Type: System.Object
  - The out parameters.

- **returnValue**
  - Type: System.Object
  - The return value.

Return Value

Type: AdsErrorCode

AdSErrorCode.

Implements

IAdsRpcInvoke.TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object) [906]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>[63]</td>
</tr>
<tr>
<td>ArgumentNullException</td>
<td>symbol</td>
</tr>
<tr>
<td>ArgumentNullException</td>
<td>rpcMethod</td>
</tr>
</tbody>
</table>
6.2.1.2.46 AdsClient.TryRead Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>☀ TryRead(UInt32, Memory, Void)</td>
<td>[293]</td>
</tr>
<tr>
<td>☀ TryRead(UInt32, UInt32, Memory, Void)</td>
<td>[294]</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

AdsClient.TryRead Method (UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryRead(
    uint variableHandle,
    Memory readBuffer,
    void readBytes
)
```

Parameters

- variableHandle Type: System.UInt32
- readBuffer Type: Memory
- readBytes Type: System.Void

Return Value

Type: AdsErrorCode [575]
AdsClient.TryRead Method (UInt32, UInt32, Memory`1, Void)

Namespace:  TwinCAT.Ads
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```c#
public AdsErrorCode TryRead(
    uint indexGroup,
    uint indexOffset,
    Memory memory,
    void readBytes
)
```

Parameters

- **indexGroup**  Type: System.UInt32
- **indexOffset** Type: System.UInt32
- **memory** Type: Memory
- **readBytes** Type: System.Void

Return Value

Type: AdsErrorCode

Reference

AdsClient Class  [154]
TryRead Overload  [293]
TwinCAT.Ads Namespace  [151]

6.2.1.2.47 AdsClient.TryReadDataType Method

Call this method to obtain information about the specified data type.

Namespace:  TwinCAT.Ads
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public AdsErrorCode TryReadDataType(
    string typeName,
    out IDataType dataType
)
```

**Parameters**

- `typeName` Type: `System.String`
  Name of the symbol.
- `dataType` Type: `TwinCAT.TypeSystem.IDataType`  
  The symbol.

**Return Value**

Type: `AdsErrorCode`  
A `IDataType` containing the requested symbol information or null if symbol could not be found.

**Implements**

`IAdsSymbolicAccess.TryReadDataType(String, IDataType.)`  

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td><code>typeName</code></td>
</tr>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td><code>63</code></td>
</tr>
</tbody>
</table>

**Reference**

- `AdsClient Class`  
- `TwinCAT.Ads Namespace`  
- `IAdsSymbolicAccess.ReadDataType(String)`  

**6.2.1.2.48 AdsClient.TryReadState Method**

Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an `AdsErrorCode` is returned. If the return value is equal to `AdsErrorCode.NoError` the call was successful.

**Namespace:** `TwinCAT.Ads`  
**Assembly:** `TwinCAT.Ads (in TwinCAT.Ads.dll)`  
Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
public AdsErrorCode TryReadState(
    out StateInfo stateInfo
)
```

Parameters

`stateInfo` Type: `TwinCAT.Ads.StateInfo` [1041]. The ADS statue and device status.

Return Value

Type: `AdsErrorCode` [575]. `AdsErrorCode` [575] of the ADS read state call. Check for `NoError` [575] to see if call was successful.

Implements

`IAdsStateProvider.TryReadState(StateInfo.)` [932]

Reference

`AdsClient Class` [154]

`TwinCAT.Ads Namespace` [151]

6.2.1.2.49 AdsClient.TryReadSymbol Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

Namespace: `TwinCAT.Ads` [151]
Assembly: `TwinCAT.Ads (in TwinCAT.Ads.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryReadSymbol(
    string name,
    out IAdsSymbol symbol
)
```

Parameters

`name` Type: `System.String` Name of the symbol.

`symbol` Type: `TwinCAT.Ads.TypeSystem.IAdsSymbol` [1379]. The symbol.

Return Value

Type: `AdsErrorCode` [575]. A `IAdsSymbol` [1379] containing the requested symbol information or null if symbol could not be found.
Implements

IAdsSymbolicAccess.TryReadSymbol(String, IAdsSymbol) [» 953]

Reference

AdsClient Class [» 154]

TwinCAT.Ads Namespace [» 151]

6.2.1.2.50 AdsClient.TryReadValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryReadValue&lt;T&gt;(String, T.) [» 297]</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
<tr>
<td>TryReadValue(ISymbol, Object.) [» 298]</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>TryReadValue&lt;T&gt;(ISymbol, T.) [» 299]</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>TryReadValue(String, Type, Object.) [» 300]</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [» 154]

TwinCAT.Ads Namespace [» 151]

 AdsClient.TryReadValue<T>. Method (String, T.)

Reads the value of a symbol and returns the value as object.

Namespace: TwinCAT.Ads [» 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryReadValue<T>(
    string name,
    out T value
)
```

Parameters

name Type: System.String
   Name of the ADS symbol.

value Type: T
    The read value of the Symbol.
Type Parameters

T  The value type.

Return Value

Type: AdsErrorCode [575]
The AdsErrorCode [575].

Implements

IAdsSymbolicAccess.TryReadValue(T,(String, T.) [954]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

The parameter type must have the same layout as the ADS symbol.

Reference

AdsClient Class [154]
TryReadValue Overload [297]
TwinCAT.Ads Namespace [151]

AdsClient.TryReadValue Method (ISymbol, Object.)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public AdsErrorCode TryReadValue(
    ISymbol symbol,
    out Object value
)

Parameters

symbol  Type: TwinCAT.TypeSystem.ISymbol [2176]
The symbol that should be read.

value   Type: System.Object.
The value.
Return Value

Type: AdsErrorCode [575]
The ADS Error Code

Implements

IAdsSymbolicAccess.TryReadValue(ISymbol, Object.) [954]

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'. Structs are not supported.

Reference

AdsClient Class [154]
TryReadValue Overload [297]
TwinCAT.Ads Namespace [151]

AdsClient.TryReadValue.T. Method (ISymbol, T.)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

    public AdsErrorCode TryReadValue<T>(
        ISymbol symbol,
        out T value
    )

Parameters

symbol Type: TwinCAT.TypeSystem.ISymbol [2176]
The symbol that should be read.

value Type: T.
The value.

Type Parameters

T The value type.

Return Value

Type: AdsErrorCode [575]
The ADS Error Code

Implements

IAdsSymbolicAccess.TryReadValue.T.ISymbol, T.) [955]
Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

AdsClient Class [154]

TryReadValue Overload [297]

TwinCAT.Ads Namespace [151]

AdsClient.TryReadValue Method (String, Type, Object.)

Reads the value of a symbol and returns the value as object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryReadValue(
    string name,
    Type type,
    out Object value
)
```

Parameters

- **name**
  - Type: System.String
  - Name of the ADS symbol.

- **type**
  - Type: System.Type
  - Managed type of the ADS symbol.

- **value**
  - Type: System.Object
  - The read value of the Symbol.

Return Value

Type: AdsErrorCode [575]
The AdsErrorCode [575].

Implements

IAdsSymbolicAccess.TryReadValue(String, Type, Object.) [956]

Remarks

The parameter type must have the same layout as the ADS symbol.

Reference

AdsClient Class [154]

TryReadValue Overload [297]
6.2.1.2.51 AdsClient.TryReadWrite Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryReadWrite(UInt32, Memory, Void, Byte)</td>
<td>[301]</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, UInt32, Memory, Void, Byte)</td>
<td>[302]</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

AdsClient.TryReadWrite Method (UInt32, Memory`1, Void, Byte)

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd2ca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryReadWrite(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer,
    byte readBytes
)
```

Parameters

- variableHandle: Type: System.UInt32
- readBuffer: Type: Memory
- writeBuffer: Type: System.Void
- readBytes: Type: System.Byte

Return Value

Type: AdsErrorCode [575]

Reference

AdsClient Class [154]
TryReadWrite Overload [301]

TwinCAT.Ads Namespace [151]

AdsClient.TryReadWrite Method (UInt32, UInt32, Memory`1, Void, Byte)

Namespace:  TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public AdsErrorCode TryReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte readBytes
)

Parameters

indexGroup Type: System.UInt32
indexOffset Type: System.UInt32
readBuffer Type: Memory
writeBuffer Type: System.Void
readBytes Type: System.Byte

Return Value

Type: AdsErrorCode [575]

Reference

AdsClient Class [154]
TryReadWrite Overload [301]
TwinCAT.Ads Namespace [151]

6.2.1.2.52 AdsClient.TryResurrect Method

Resurrects the connection

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#
public bool TryResurrect (  
    out AdsException error
)

Parameters

error          Type: TwinCAT.AdsException [57].
The error.

Return Value

Type: Boolean
true if resurrection was accepted, false otherwise.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]

6.2.1.2.53 AdsClient.TryWrite Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[TryWrite(UInt32, ReadOnlyMemory) [303]]</td>
<td></td>
</tr>
<tr>
<td>[TryWrite(UInt32, UInt32, ReadOnlyMemory) [304]]</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]

AdsClient.TryWrite Method (UInt32, ReadOnlyMemory`1)

Namespace:  TwinCAT.Ads [151]
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public AdsErrorCode TryWrite(
    uint variableHandle,
    ReadOnlyMemory writeBuffer
)
```

Parameters

variableHandle Type: System.UInt32

writeBuffer Type: ReadOnlyMemory

Return Value

Type: AdsErrorCode

Reference

AdsClient Class [154]
TryWrite Overload [303]
TwinCAT.Ads Namespace [151]

AdsClient.TryWrite Method (UInt32, UInt32, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryWrite(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer
)
```

Parameters

indexGroup Type: System.UInt32

indexOffset Type: System.UInt32

writeBuffer Type: ReadOnlyMemory

Return Value

Type: AdsErrorCode

Reference

AdsClient Class [154]
6.2.1.2.54 AdsClient.TryWriteControl Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![TryWriteControl(StateInfo) ![305]]</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>![TryWriteControl(StateInfo, ReadOnlyMemory) ![306]]</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsClient Class ![154]

TwinCAT.Ads Namespace ![151]

AdsClient.TryWriteControl Method (StateInfo)

Changes the ADS status and the device status of an ADS server.

Namespace: TwinCAT.Ads ![151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryWriteControl(
    StateInfo stateInfo
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stateInfo</td>
<td>TwinCAT.Ads.StateInfo ![1041]</td>
</tr>
</tbody>
</table>

New ADS status and device status.

Return Value

Type: AdsErrorCode ![575]
AdsErrorCode.

Implements

IAdsStateControl.TryWriteControl(StateInfo) ![917]
### AdsClient.TryWriteControl Method (StateInfo, ReadOnlyMemory`1)

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public AdsErrorCode TryWriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer
)
```

**Parameters**

- `stateInfo`: Type: TwinCAT.Ads.StateInfo
- `writeBuffer`: Type: ReadOnlyMemory

**Return Value**

Type: AdsErrorCode

**Reference**

- AdsClient Class [154]
- TryWriteControl Overload [305]
- TwinCAT.Ads Namespace [151]

### 6.2.1.2.55 AdsClient.TryWriteValue Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryWriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryWriteValue(T, (String, T))</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryWriteValue(ISymbol, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryWriteValue.T. (ISymbol, T)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
</tbody>
</table>

**Reference**

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

**AdsClient.TryWriteValue Method (String, Object)**

Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public AdsErrorCode TryWriteValue(
    string name,
    Object value
)
```

**Parameters**

- **name**  
  Type: System.String  
  Name of the ADS symbol.

- **value**  
  Type: System.Object  
  Object holding the value to be written to the ADS symbol

**Return Value**

Type: AdsErrorCode [575]  
AdsErrorCode.

**Implements**

IAdsSymbolicAccess.TryWriteValue(String, Object) [957]

**Reference**

AdsClient Class [154]

TryWriteValue Overload [306]

TwinCAT.Ads Namespace [151]

**AdsClient.TryWriteValue.T. Method (String, T)**

Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.
Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

class Ads

public AdsErrorCode TryWriteValue<T>(
    string name,
    T value
)

Parameters

name Type: System.String
Name of the ADS symbol.
value Type: T
Object holding the value to be written to the ADS symbol

Type Parameters

T The value type.

Return Value

Type: AdsErrorCode

Implements

IAdsSymbolicAccess.TryWriteValue<T>(String, T)

Reference

AdsClient Class

TryWriteValue Overload

TwinCAT.Ads Namespace

AdsClient.TryWriteValue Method (ISymbol, Object)

Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

class Ads

public AdsErrorCode TryWriteValue(ISymbol symbol,
    Object val
)
Parameters

symbol
  Type: TwinCAT.TypeSystem.ISymbol [2176]
  The symbol the value is written to.

val
  Type: System.Object
  The value to write.

Return Value

Type: AdsErrorCode [575]
AdsErrorCode.

Implements

IAdsSymbolicAccess.TryWriteValue(ISymbol, Object) [959]

Reference

AdsClient Class [154]
TryWriteValue Overload [306]
TwinCAT.Ads Namespace [151]

AdsClient.TryWriteValue<T>. Method (ISymbol, T)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryWriteValue<T>(
    ISymbol symbol,
    T val
)
```

Parameters

symbol
  Type: TwinCAT.TypeSystem.ISymbol [2176]
  The symbol the value is written to.

val
  Type: T
  The value to write.

Type Parameters

T
  The value type.

Return Value

Type: AdsErrorCode [575]
AdsErrorCode.
6.2.1.2.56 AdsClient.UnregisterAdsStateChangedAsync Method

Unregister ads state changed as an asynchronous operation.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultAds> UnregisterAdsStateChangedAsync(
    EventHandler<AdsStateChangedEventArgs> handler,
    CancellationToken cancel)
```

Parameters

- **handler**
  Type: `System.EventHandler<AdsStateChangedEventArgs>`
  The handler function to be unregistered.

- **cancel**
  Type: `System.Threading.CancellationToken`
  The cancellation token.

Return Value

Type: `Task<ResultAds>`
A task that represents the asynchronous 'UnregisterAdsStateChanged' operation. The `ResultAds` parameter contains the state the `ErrorCode` of the ADS communication after execution.

Implements

- `IAdsStateProvider.UnregisterAdsStateChangedAsync(EventHandler<AdsStateChangedEventArgs>, CancellationToken)`

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ObjectDisposedException</code></td>
<td></td>
</tr>
</tbody>
</table>

Reference

- AdsClient Class [154]
- TwinCAT.Ads Namespace [151]

6.2.1.2.57 AdsClient.UnregisterSymbolVersionChangedAsync Method

Unregisters from an AdsSymbolVersionChanged event as an asynchronous operation.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultAds> UnregisterSymbolVersionChangedAsync(
    EventHandler<AdsSymbolVersionChangedEventArgs> handler,
    CancellationToken cancel)
```

Parameters

- **handler**
  Type: System.EventHandler<AdsSymbolVersionChangedEventArgs>
  The handler function to unregister.

- **cancel**
  Type: System.Threading.CancellationToken
  The cancellation token.

Return Value

Type: Task<ResultAds>
A task that represents the asynchronous 'UnregisterSymbolVersionChangedAsync' operation. The ResultAds parameter contains the value ErrorCode of the ADS communication after execution.

Implements


Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsClient Class
TwinCAT.Ads Namespace

6.2.1.2.58 AdsClient.Write Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write(UInt32, ReadOnlyMemory)</td>
<td>[312]</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Write(UInt32, UInt32) [312]</td>
<td>Triggers a 'Write' call to the ADS device at the specified address.</td>
</tr>
<tr>
<td>Write(UInt32, UInt32, ReadOnlyMemory) [313]</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

**AdsClient.Write Method (UInt32, ReadOnlyMemory`1)**

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void Write(
    uint variableHandle,
    ReadOnlyMemory writeBuffer
)
```

**Parameters**

- `variableHandle` Type: System.UInt32
- `writeBuffer` Type: ReadOnlyMemory

Reference

AdsClient Class [154]

Write Overload [311]

TwinCAT.Ads Namespace [151]

**AdsClient.Write Method (UInt32, UInt32)**

Triggers a 'Write' call to the ADS device at the specified address.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public void Write(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer
)
```

Parameters

- `indexGroup` Type: `System.UInt32`
  The index group.
- `indexOffset` Type: `System.UInt32`
  The index offset.
- `writeBuffer` Type: `ReadOnlyMemory`

Implements

`IAdsReadWrite2.Write(UInt32, UInt32)`

Reference

- AdsClient Class [154]
- Write Overload [311]
- TwinCAT.Ads Namespace [151]

AdsClient.Write Method (UInt32, UInt32, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void Write(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer
)
```

Parameters

- `indexGroup` Type: `System.UInt32`
- `indexOffset` Type: `System.UInt32`
- `writeBuffer` Type: `ReadOnlyMemory`

Reference

- AdsClient Class [154]
- Write Overload [311]
- TwinCAT.Ads Namespace [151]
### 6.2.1.2.59 AdsClient.WriteAny Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteAny(UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object, Int32)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAny(UInt32, UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAny(UInt32, UInt32, Object, Int32)</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
</tbody>
</table>

**Reference**

*AdsClient Class [154]*

*TwinCAT.Ads Namespace [151]*

### AdsClient.WriteAny Method (UInt32, Object)

写的对象同步地写入 ADS 设备。

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public void WriteAny(  
    uint variableHandle,  
    Object value  
)
```

**Parameters**

- **variableHandle**  
  Type: System.UInt32  
  Handle of the ADS variable.

- **value**  
  Type: System.Object  
  Object to write to the ADS device.

**Implements**

*IAdsAnyAccess.WriteAny(UInt32, Object) [726]*

**Reference**

*AdsClient Class [154]*
AdsClient.WriteAny Method (UInt32, Object, Int32.)

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void WriteAny(
    uint variableHandle,
    Object value,
    int[] args
)
```

Parameters

- **variableHandle**: Type: System.UInt32
  Handle of the ADS variable.
- **value**: Type: System.Object
  Object to write to the ADS device.
- **args**: Type: System.Int32.
  Additional arguments.

Implements

IAdsAnyAccess.WriteAny(UInt32, Object, Int32.)

Remarks

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395].</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

- AdsClient Class
- WriteAny Overload
- TwinCAT.Ads Namespace

AdsClient.WriteAny Method (UInt32, UInt32, Object)

Writes an object synchronously to an ADS device.
### AdsClient.WriteAny Method (UInt32, UInt32, Object, .Int32.)

Writes an object synchronously to an ADS device.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd2ca3e72bc0ea15da1c14

**Syntax**

```csharp
public void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args
)
```

**Parameters**

- **indexGroup**
  - Type: System.UInt32
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - Type: System.UInt32
  - Contains the index offset number of the requested ADS service.

- **value**
  - Type: System.Object
  - Object to write to the ADS device.

- **args**
  - Type: System.Int32
  - Arguments for the method.

**Implements**

IAdsAnyAccess.WriteAny(UInt32, UInt32, Object)
value

Type: System.Object
Object to write to the ADS device.

args

Type: System.Int32
Additional arguments.

**Implements**

IAdsAnyAccess.WriteAny(UInt32, UInt32, Object, Int32.) [728]

**Remarks**

If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

**Remarks**

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395]</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

AdsClient Class [154]

WriteAny Overload [314]

TwinCAT.Ads Namespace [151]

**6.2.1.2.60 AdsClient.WriteAnyAsync Method**

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteAnyAsync(UInt32, Object, CancellationToken) [318]</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, Int32, CancellationToken) [319]</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [320]</td>
<td>Write the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, CancellationToken)</td>
<td>Write the value of an Anytype (Primitive type) asynchronously.</td>
</tr>
</tbody>
</table>

**Reference**

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

**AdsClient.WriteAnyAsync Method (UInt32, Object, CancellationToken)**

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```
public Task<ResultWrite> WriteAnyAsync(
    uint variableHandle,
    Object value,
    CancellationToken cancel)
```

**Parameters**

- **variableHandle**
  - Type: System.UInt32
  - Handle of the ADS variable.

- **value**
  - Type: System.Object
  - Object to write to the ADS device.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

**Return Value**

Type: Task<ResultWrite> [1032].

A task that represents the asynchronous task operation. The result parameter ResultWrite [1032] of the write operation contains the ErrorCode [992].

**Implements**

| AdsAnyAccess.WriteAnyAsync(UInt32, Object, CancellationToken) [730] |  |

**Remarks**

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395].</td>
</tr>
</tbody>
</table>
Type of value Parameter | Necessary Arguments (args)
--- | ---
string[] | Array

Reference

AdsClient Class [154]

WriteAnyAsync Overload [317]

TwinCAT.Ads Namespace [151]

**AdsClient.WriteAnyAsync Method (UInt32, Object, Int32., CancellationToken)**

Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public Task<ResultWrite> WriteAnyAsync(
    uint variableHandle,
    Object value,
    int[] args,
    CancellationToken cancel)
```

**Parameters**

variableHandle

Type: System.UInt32  
Handle of the ADS variable.

value

Type: System.Object  
Object to write to the ADS device.

args

Type: System.Int32  
Additional arguments.

cancel

Type: System.Threading.CancellationToken  
The cancellation token.

**Return Value**

Type: Task<ResultWrite> [1032].  
Task<ResultWrite>.

**Return Value**

Type: Task<ResultWrite> [1032].  
A task that represents the asynchronous task operation. The result parameter ResultWrite [1032] of the write operation contains the ErrorCode [992].

**Implements**

IAdsAnyAccess.WriteAnyAsync(UInt32, Object, Int32., CancellationToken) [731]
Remarks

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395].</td>
</tr>
<tr>
<td>string[]</td>
<td>Array</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

WriteAnyAsync Overload [317]

TwinCAT.Ads Namespace [151]

**AdsClient.WriteAnyAsync Method (UInt32, UInt32, Object, CancellationToken)**

Write the value of an Anytype (Primitive type) asynchronously.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Task<ResultWrite> WriteAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Object value,
    CancellationToken cancel
)
```

**Parameters**

- **indexGroup**
  - Type: System.UInt32
  - The index group.

- **indexOffset**
  - Type: System.UInt32
  - The index offset.

- **value**
  - Type: System.Object
  - The value.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

**Return Value**

Type: Task<ResultWrite> [1032].

A task that represents the asynchronous 'ReadState' operation. The ResultWrite [1032] parameter contains the value the ErrorCode [992] of the ADS communication after execution.

**Implements**

IAdsAnyAccess.WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [732]
AdsClient.WriteAnyAsync Method (UInt32, UInt32, Object, .Int32., CancellationToken)

Write the value of an Anytype (Primitive type) asynchronously.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultWrite> WriteAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args,
    CancellationToken cancel)
```

Parameters

- **indexGroup**
  Type: System.UInt32
  The index group.

- **indexOffset**
  Type: System.UInt32
  The index offset.

- **value**
  Type: System.Object
  The value.

- **args**
  Type: System.Int32
  The type arguments (AnyType)

- **cancel**
  Type: System.Threading.CancellationToken
  The cancellation token.

Return Value

Type: Task<ResultWrite>
A task that represents the asynchronous 'ReadState' operation. The ResultWrite parameter contains the value the ErrorCode of the ADS communication after execution.

Implements

- IAdsAnyAccess.WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken)

Reference

- AdsClient Class
- WriteAnyAsync Overload
- TwinCAT.Ads Namespace
### AdsClient.WriteAnyStringAsync Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>WriteAnyStringAsync(String, String, Int32, Encoding, CancellationToken)</code></td>
<td>Writes the string (Potentially unsafe!)</td>
</tr>
<tr>
<td><code>WriteAnyStringAsync(UInt32, String, Int32, Encoding, CancellationToken)</code></td>
<td>Writes the string (Potentially unsafe!)</td>
</tr>
</tbody>
</table>

#### Reference

- **AdsClient Class** [154]
- **TwinCAT.Ads Namespace** [151]

**AdsClient.WriteAnyStringAsync Method (String, String, Int32, Encoding, CancellationToken)**

Writes the string (Potentially unsafe!)

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Task<ResultWrite> WriteAnyStringAsync(
    string symbolPath,
    string value,
    int length,
    Encoding encoding,
    CancellationToken cancel
)
```

**Parameters**

- **symbolPath**
  - Type: `System.String`
  - The symbol path.
- **value**
  - Type: `System.String`
  - The value.
- **length**
  - Type: `System.Int32`
  - The length of the string to write
- **encoding**
  - Type: `System.Text.Encoding`
  - The encoding.
- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.
Return Value

Type: Task<ResultWrite>.
Task<ResultWrite>.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>[63]</td>
</tr>
</tbody>
</table>

Remarks

ATTENTION: Potentially this method is unsafe because following data can be overwritten after the string symbol. Please be sure to specify the string length lower than the string size reserved within the process image! The String is written with the specified encoding.

Reference

AdsClient Class [154]

WriteAnyStringAsync Overload [322]

TwinCAT.Ads Namespace [151]

AdsClient.WriteAnyStringAsync Method (UInt32, String, Int32, Encoding, CancellationToken)

Writes the string (Potentially unsafe!)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultWrite> WriteAnyStringAsync(
    uint variableHandle,
    string value,
    int length,
    Encoding encoding,
    CancellationToken cancel
)
```

Parameters

| variableHandle | Type: System.UInt32  
The variable handle. |
|----------------|----------------------|
| value          | Type: System.String  
The value.          |
| length         | Type: System.Int32   
The length of the string to write |
| encoding       | Type: System.Text.Encoding  
The encoding. |
cancel

Type: System.Threading.CancellationToken
The cancellation token.

**Return Value**

Type: Task<ResultWrite>.
Task<ResultWrite>.

**Remarks**

ATTENTION: Potentially this method is unsafe because following data can be overwritten after the string symbol. Please be sure to specify the string length lower than the string size reserved within the process image! The String is written with the specified encoding.

**Reference**

AdsClient Class [154]

WriteAnyStringAsync Overload [322]

TwinCAT.Ads Namespace [151]

6.2.1.2.62 AdsClient.WriteAsync Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="324" alt="WriteAsync(UInt32, ReadOnlyMemory, Void)" /></td>
<td>Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.</td>
</tr>
<tr>
<td><img src="325" alt="WriteAsync(UInt32, UInt32, CancellationToken)" /></td>
<td>Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.</td>
</tr>
<tr>
<td><img src="326" alt="WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void)" /></td>
<td>Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.</td>
</tr>
</tbody>
</table>

**Reference**

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

**AdsClient.WriteAsync Method (UInt32, ReadOnlyMemory`1, Void)**

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Shaf00b9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public Task<ResultWrite> WriteAsync(
    uint variableHandle,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

Parameters

| variableHandle               | Type: System.UInt32 |
| writeBuffer                  | Type: ReadOnlyMemory |
| cancel                        | Type: System.Void    |

Return Value

Type: Task<ResultWrite>

Reference

AdsClient Class [154]
WriteAsync Overload [324]
TwinCAT.Ads Namespace [151]

**AdsClient.WriteAsync Method (UInt32, UInt32, CancellationToken)**

Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultWrite> WriteAsync(
    uint indexGroup,
    uint indexOffset,
    CancellationToken cancel
)
```

Parameters

| indexGroup     | Type: System.UInt32 |
| indexOffset    | Type: System.UInt32 |
| cancel         | Type: System.Threading.CancellationToken |

The index group.
The index offset.
The cancellation token.
Return Value

Type: Task<ResultWrite>.
A task that represents the asynchronous 'ReadWrite' operation. The ResultWrite parameter contains the ErrorCode after execution.

Reference

AdsClient Class
WriteAsync Overload
TwinCAT.Ads Namespace

AdsClient.WriteAsync Method (UInt32, UInt32, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultWrite> WriteAsync(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

Parameters

- indexGroup Type: System.UInt32
- indexOffset Type: System.UInt32
- writeBuffer Type: ReadOnlyMemory
- cancel Type: System.Void

Return Value

Type: Task<ResultWrite>.

Reference

AdsClient Class
WriteAsync Overload
TwinCAT.Ads Namespace
6.2.1.2.63 AdsClient.WriteControl Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>WriteControl(StateInfo, ReadOnlyMemory)</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

AdsClient.WriteControl Method (StateInfo)

Changes the ADS status and the device status of an ADS server.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void WriteControl(
    StateInfo stateInfo
)
```

Parameters

stateInfo Type: TwinCAT.Ads.StateInfo [1041]
New ADS status and device status.

Implements

IAdsStateControl.WriteControl(StateInfo) [918]

Reference

AdsClient Class [154]

WriteControl Overload [327]

TwinCAT.Ads Namespace [151]
**AdsClient.WriteControl Method (StateInfo, ReadOnlyMemory)\(^{1}\)**

**Namespace:** [TwinCAT.Ads](#151)  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll)  
**Version:** 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public void WriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer
)
```

### Parameters

- `stateInfo` Type: [TwinCAT.Ads.StateInfo](#1041)
- `writeBuffer` Type: [ReadOnlyMemory](#)

### Reference

- [AdsClient Class](#154)
- [WriteControl Overload](#327)
- [TwinCAT.Ads Namespace](#151)

### 6.2.1.2.64 AdsClient.WriteControlAsync Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteControlAsync(AdsState, UInt16, CancellationToken) [328]</td>
<td>Changes the ADS status and device status of the ADS server asynchronously.</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [329]</td>
<td></td>
</tr>
</tbody>
</table>

### Reference

- [AdsClient Class](#154)
- [TwinCAT.Ads Namespace](#151)

### AdsClient.WriteControlAsync Method (AdsState, UInt16, CancellationToken)

Changes the ADS status and device status of the ADS server asynchronously.
Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```c#
public Task<ResultAds> WriteControlAsync(
    AdsState adsState,
    ushort deviceState,
    CancellationToken cancel
)
```

Parameters

- **adsState**
  - Type: TwinCAT.Ads.AdsState
  - The ADS state.
- **deviceState**
  - Type: System.UInt16
  - The device state.
- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

Return Value

Type: Task.ResultAds

A task that represents the asynchronous 'WriteControl' operation. The ResultAds parameter contains the state the ErrorCode of the ADS communication after execution.

Implements

-IAdsStateControl.WriteControlAsync(AdsState, UInt16, CancellationToken)

Reference

AdsClient Class

WriteControlAsync Overload

TwinCAT.Ads Namespace

AdsClient.WriteControlAsync Method (AdsState, UInt16, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```c#
public Task<ResultAds> WriteControlAsync(
    AdsState adsState,
    ushort deviceState,
    ReadOnlyMemory writeData,
    void cancel
)
```
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>adsState</td>
<td>TwinCAT.Ads.AdsState</td>
</tr>
<tr>
<td>deviceState</td>
<td>System.UInt16</td>
</tr>
<tr>
<td>writeData</td>
<td>ReadOnlyMemory</td>
</tr>
<tr>
<td>cancel</td>
<td>System.Void</td>
</tr>
</tbody>
</table>

Return Value

Type: Task.ResultAds

Reference

AdsClient Class

WriteControlAsync Overload

TwinCAT.Ads Namespace

6.2.1.2.65 AdsClient.WriteByteSymbolAsync Method

Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultWrite> WriteSymbolAsync(
    string name,
    Object value,
    CancellationToken cancel
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>System.String</td>
</tr>
<tr>
<td>value</td>
<td>System.Object</td>
</tr>
<tr>
<td>cancel</td>
<td>System.Threading.CancellationToken</td>
</tr>
</tbody>
</table>

Return Value

Type: Task.ResultWrite

A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite parameter contains the ErrorCode after execution.
6.2.1.2.66 AdsClient.WriteValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue.T.(String, T)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue(ISymbol, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValue.T.(ISymbol, T)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
</tbody>
</table>

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]

AdsClient.WriteValue Method (String, Object)

Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void WriteValue(
    string name,
    Object value
)
```

Parameters

- **name**: Type: System.String
  Name of the ADS symbol.
- **value**: Type: System.Object
  Object holding the value to be written to the ADS symbol.
Implements

IAdsSymbolicAccess.WriteValue(String, Object) [961]

Reference

AdsClient Class [154]

WriteValue Overload [331]

TwinCAT.Ads Namespace [151]

**AdsClient.WriteValue.T. Method (String, T)**

Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public void WriteValue<T>(
    string name,
    T value
)
```

**Parameters**

- **name**  
  Type: System.String  
  Name of the ADS symbol.

- **value**  
  Type: T  
  Object holding the value to be written to the ADS symbol

**Type Parameters**

- **T**  
  the value type.

**Implements**

IAdsSymbolicAccess.WriteValue.T.(String, T) [961]

**Reference**

AdsClient Class [154]

WriteValue Overload [331]

TwinCAT.Ads Namespace [151]

**AdsClient.WriteValue Method (ISymbol, Object)**

Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void WriteValue(
    ISymbol symbol,
    Object val
)
```

**Parameters**

- **symbol**
  - Type: TwinCAT.TypeSystem.ISymbol
  - The symbol the value is written to.

- **val**
  - Type: System.Object
  - The value to write.

**Implements**

- IAdsSymbolicAccess.WriteValue(ISymbol, Object) [962]

**Reference**

- AdsClient Class [154]
- WriteValue Overload [331]

**AdsClient.WriteValue.T. Method (ISymbol, T)**

Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void WriteValue<T>(
    ISymbol symbol,
    T val
)
```

**Parameters**

- **symbol**
  - Type: TwinCAT.TypeSystem.ISymbol
  - The symbol the value is written to.

- **val**
  - Type: T
  - The value to write.
Type Parameters
T The value type.

Implements
IAdsSymbolicAccess.WriteValue.T.ISymbol, T [963]

Reference
AdsClient Class [154]
WriteValue Overload [331]
TwinCAT.Ads Namespace [151]

6.2.1.2.67 AdsClient.WriteValueAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValueAsync.T. (String, T, CancellationToken) [334]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValueAsync(ISymbol, Object, CancellationToken) [335]</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValueAsync.T. (ISymbol, T, CancellationToken) [336]</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
</tbody>
</table>

Reference
AdsClient Class [154]
TwinCAT.Ads Namespace [151]

AdsClient.WriteValueAsync.T. Method (String, T, CancellationToken)

Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public Task<ResultWrite> WriteValueAsync<T>(
    string name,
    T value,
    CancellationToken cancel
)
```

Parameters

- name
  - Type: System.String
  - Name of the ADS symbol.
- value
  - Type: T
  - Object holding the value to be written to the ADS symbol.
- cancel
  - Type: System.Threading.CancellationToken
  - The cancel token.

Type Parameters

- T
  - The value type.

Return Value

- Type: Task<ResultWrite>
- A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite parameter contains the ErrorCode after execution.

Implements

- AdsSymbolicAccess.WriteValueAsync<T>(String, T, CancellationToken) [964]

Reference

- AdsClient Class [154]
- WriteValueAsync Overload [334]
- TwinCAT.Ads Namespace [151]

**AdsClient.WriteValueAsync Method (ISymbol, Object, CancellationToken)**

Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0e15da1c14

Syntax

C#

```csharp
public Task<ResultWrite> WriteValueAsync(
    ISymbol symbol,
    Object val,
    CancellationToken cancel
)
```
Parameters

symbol
Type: TwinCAT.TypeSystem.ISymbol
The symbol the value is written to.

val
Type: System.Object
The value to write.

cancel
Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultWrite>
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite parameter contains the ErrorCode after execution.

Implements

IAdsSymbolicAccess.WriteValueAsync(ISymbol, Object, CancellationToken)

Reference

AdsClient Class
WriteValueAsync Overload
TwinCAT.Ads Namespace

AdsClient.WriteValueAsync<T>. Method (ISymbol, T, CancellationToken)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public Task<ResultWrite> WriteValueAsync<T>(
    ISymbol symbol,
    T val,
    CancellationToken cancel
)

Parameters

symbol
Type: TwinCAT.TypeSystem.ISymbol
The symbol the value is written to.

val
Type: T
The value to write.

cancel
Type: System.Threading.CancellationToken
The cancellation token.
Type Parameters

T  The value type.

Return Value

Type: Task<ResultWrite[1032]>
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite[1032] parameter contains the ErrorCode[992] after execution.

Implements

IAdsSymbolicAccess.WriteValueAsync.T.ISymbol, T, CancellationToken)[966]

Reference

AdsClient Class[154]
WriteValueAsync Overload[334]
TwinCAT.Ads Namespace[151]

6.2.1.3 AdsClient Events

The AdsClient[154] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsNotification</td>
<td>Occurs when the ADS device sends a notification to the client.</td>
</tr>
<tr>
<td>AdsNotificationError</td>
<td>Occurs when an exception has occurred during notification management.</td>
</tr>
<tr>
<td>AdsNotificationEx</td>
<td>Occurs when the ADS device sends a notification to the client.</td>
</tr>
<tr>
<td>AdsStateChanged</td>
<td>Occurs when the ADS state changes.</td>
</tr>
<tr>
<td>AdsSumNotification</td>
<td>Occurs when Notifications are send (bundled notifications)</td>
</tr>
<tr>
<td>AdsSymbolVersionChanged</td>
<td>Occurs when the symbol version has been changed changes.</td>
</tr>
<tr>
<td>ConnectionStateChanged</td>
<td>Occurs when the connection state has been changed.</td>
</tr>
<tr>
<td>RouterStateChanged</td>
<td>Occurs when the state of the local Router has changed.</td>
</tr>
</tbody>
</table>
6.2.1.3.1 AdsClient.AdsNotification Event

Occurs when the ADS device sends a notification to the client.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# public event EventHandler<AdsNotificationEventArgs> AdsNotification

**Value**

Type: System.EventHandler<AdsNotificationEventArgs>

**Implements**

IAdsNotifications.AdsNotification

**Remarks**

The Event Argument contains the raw data value of the notification, not marshalled to .NET types.

**Examples**

Example of receiving AdsNotification events.

**Trigger on changed values by ADS Notifications**

```csharp
private async Task RegisterNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;
    using (AdsClient client = new AdsClient())
    {
        // Add the Notification event handler
        client.AdsNotification += Client_AdsNotification2;
        // Connect to target
        client.Connect(AmsNetId.Local, 851);
        uint notificationHandle = 0;
        // Notification to a DINT Type (UINT32)
        // Check for change every 200 ms
        byte[] notificationBuffer = new byte[sizeof(UInt32)];
        int size = sizeof(UInt32);
        ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", size, new NotificationSettings(AdsTransModeOnChange, 200, 0), null, cancel);
        if (result.Succeeded)
        {
            notificationHandle = result.Handle;
            await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
            // Unregister the Event / Handle
            ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
        }
    }
}
```
public event EventHandler<AdsNotificationErrorEventArgs> AdsNotificationError

Value

Type: System.EventHandler<AdsNotificationErrorEventArgs> [595].

Implements

IAdsNotifications.AdsNotificationError [867]

Remarks

The occurrence of this event can have two different reasons:

1. Indicates an internal error occurred during Notification management.
2. The registered notification becomes invalid on the server, eg. after a PLC Download / Online Change. If the ADS Server detects that the (still registered) Notification Sender is getting invalid, it sends an error notification so that the client will be informed about detached notifications. The event arguments contains the AdsInvalidNotificationException [591] which describes the invalid notification handle by its Handle [593] property.

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]

TwinCAT.Ads Namespace [151]

6.2.1.3.2 AdsClient.AdsNotificationError Event

Occurs when a exception has occurred during notification management.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public event EventHandler<AdsNotificationErrorEventArgs> AdsNotificationError

Value

Type: System.EventHandler<AdsNotificationErrorEventArgs> [595].

Implements

IAdsNotifications.AdsNotificationError [867]

Remarks

The occurrence of this event can have two different reasons:

1. Indicates an internal error occurred during Notification management.
2. The registered notification becomes invalid on the server, eg. after a PLC Download / Online Change. If the ADS Server detects that the (still registered) Notification Sender is getting invalid, it sends an error notification so that the client will be informed about detached notifications. The event arguments contains the AdsInvalidNotificationException [591] which describes the invalid notification handle by its Handle [593] property.

Reference

AdsClient Class [154]
TwinCAT.Ads Namespace [151]
TwinCAT.Ads.AdsInvalidNotificationException [591]
6.2.1.3.3 AdsClient.AdsNotificationEx Event

Occurs when the ADS devices sends a notification to the client.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public event EventHandler<AdsNotificationExEventArgs> AdsNotificationEx
```

**Value**

Type: System.EventHandler<AdsNotificationExEventArgs>

**Implements**

IAdsNotifications.AdsNotificationEx

**Remarks**

The Notification event arguments marshals the data value automatically to the specified .NET Type with ANY_TYPE marshellers.

**Examples**

Example of receiving AdsNotificationEx events.

**Trigger on changed values by ADS Notifications**

CancellationToken cancel = CancellationToken.None;

```csharp
using (AdsClient client = new AdsClient())
{
    client.AdsNotificationEx += Client_AdsNotificationEx;
    client.Connect(AmsNetId.Local, 851);
    // Add UDINT
    ResultHandle resultHandle = await client.AddDeviceNotificationExAsync("MAIN.udint", new NotificationSettings(AdsTransModeOnChange, 200, 200), null, typeof(uint),null, cancel);
    await Task.Delay(5000, cancel); // Wait ....
    ResultAds resultHandleDelete = await client.DeleteDeviceNotificationAsync(resultHandle.Handle, cancel); // Unregister Event
}
```

**Reference**

AdsClient Class

TwinCAT.Ads Namespace

6.2.1.3.4 AdsClient.AdsStateChanged Event

Occurs when the ADS state changes.
**Namespace**: TwinCAT.Ads

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public event EventHandler<AdsStateChangedEventArgs> AdsStateChanged
```

**Value**

Type: System.EventHandler<AdsStateChangedEventArgs>.

**Implements**

IAdsStateProvider.AdsStateChanged

### Remarks

This works only for ports that support Notifications (e.g. Port 851 but not Port 10000). In case of

**Reference**

AdsClient Class

TwinCAT.Ads Namespace

#### 6.2.1.3.5 AdsClient.AdsSumNotification Event

Occurs when Notifications are send (bundled notifications)

**Namespace**: TwinCAT.Ads

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public event EventHandler<AdsSumNotificationEventArgs> AdsSumNotification
```

**Value**

Type: System.EventHandler<AdsSumNotificationEventArgs>.

**Implements**

IAdsNotifications.AdsSumNotification

**Remarks**

As an optimization, this event receives all ADS Notifications that occurred at one point in time together. As consequence, the overhead of handler code is reduced, what can be important if notifications are triggered in a high frequency and the event has to be synchronized to the UI thread context. Because multiple notifications are bound together, less thread synchronization is necessary. The AdsNotification and
AdsNotificationEx [340] events shouldn't be used when SumNotifications are registered, because they have an performance side effect to this AdsSumNotification event. The full performance is reached only, when all notifications are handled on this event.

Examples

Example of receiving AdsSumNotification events.

Trigger on changed values by ADS Notifications

```csharp
private async Task RegisterSumNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;

    using (AdsClient client = new AdsClient())
    {
        // Add the Notification event handler
        client.AdsSumNotification += Client_SumNotification;

        // Connect to target
        client.Connect(AmsNetId.Local, 851);
        uint notificationHandle = 0;

        // Notification to a DINT Type (UINT32)
        // Check for change every 200 ms
        byte[] notificationBuffer = new byte[sizeof(UInt32)];
        ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", sizeof(UInt32), new NotificationSettings(AdsTransMode.OnChange, 200, 0), null, cancel);

        if (result.Succeeded)
        {
            notificationHandle = result.Handle;
            await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
            // Unregister the Event / Handle
            ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
        }
        client.AdsNotification -= Client_AdsNotification2;
    }
}

private void Client_SumNotification(object sender, AdsSumNotificationEventArgs e)
{
    // Timestamp of the Notification List
    DateTimeOffset dateTime = e.TimeStamp;

    // List of Raw ADS Notifications
    IList<Notification> notifications = e.Notifications;

    foreach(Notification notification in notifications)
    {
        // Notifications can be handled more efficiently, because they occur togeterh
        // handler and can be transformed/synchronized in one step compared to AdsClient.AdsNotification events.
    }
}
```

Reference

AdsClient Class [154]

TwinCAT.Ads Namespace [151]

AdsClient.AdsNotification [338]

6.2.1.3.6 AdsClient.AdsSymbolVersionChanged Event

Occurs when the symbol version has been changed changes.
**Namespace:** TwinCAT.Ads (➤ 151)

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public event EventHandler<AdsSymbolVersionChangedEventArgs> AdsSymbolVersionChanged
```

**Value**

Type: `System.EventHandler<AdsSymbolVersionChangedEventArgs>` (➤ 637)

**Implements**

IAdsSymbolChangedProvider.AdsSymbolVersionChanged (➤ 937)

**Remarks**

This is the case when the connected ADS server restarts. This invalidates all actual opened symbol handles. The SymbolVersion counter doesn't trigger, when an online change is made on the PLC (ports 801, ..., 851 ...)

### Reference

- AdsClient Class (➤ 154)
- TwinCAT.Ads Namespace (➤ 151)

### 6.2.1.3.7 AdsClient.ConnectionStateChanged Event

Occurs when the connection state has been changed.

**Namespace:** TwinCAT.Ads (➤ 151)

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public event EventHandler<ConnectionStateChangedEventArgs> ConnectionStateChanged
```

**Value**

Type: `System.EventHandler<ConnectionStateChangedEventArgs>` (➤ 68)

**Implements**

IConnectionStateProvider.ConnectionStateChanged (➤ 86)

### Reference

- AdsClient Class (➤ 154)
- TwinCAT.Ads Namespace (➤ 151)
6.2.1.3.8 AdsClient.RouterStateChanged Event

Occurs when the state of the local Router has changed.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public event EventHandler<AmsRouterNotificationEventArgs> RouterStateChanged
```

**Value**

Type: `System.EventHandler<AmsRouterNotificationEventArgs>`

**Implements**

`IRouterNotificationProvider.RouterStateChanged`

**Reference**

AdsClient Class

TwinCAT.Ads Namespace

6.2.2 AdsClientSettings Class

Settings object for the `AdsClient` class.

**Inheritance Hierarchy**

System.Object

TwinCAT.Ads.AdsClientSettings

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public class AdsClientSettings
```

The `AdsClientSettings` type exposes the following members.

**Constructors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>☁️ AdsClientSettings</td>
<td>Creates a Default settings AdsClientSettings object with custom timeout.</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompatibilityDefault</td>
<td>Compatibility settings object</td>
</tr>
<tr>
<td>Default</td>
<td>Gets the default settings (Default interceptors, Timeout 5000 ms)</td>
</tr>
<tr>
<td>FastWriteThrough</td>
<td>Gets a Settings object that configures the AdsClient for FastWriteThrough</td>
</tr>
<tr>
<td>Timeout</td>
<td>The communication Timeout that is set initially on the AdsClient</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
</tbody>
</table>

### Remarks

This AdsClientSettings object is used to initialize the AdsClient with application appropriate settings. Several predefined application dependant settings are available as static properties:

- Default
- FastWriteThrough
- CompatibilityDefault

### Reference

TwinCAT.Ads Namespace

### AdsClientSettings Constructor

Creates a Default settings AdsClientSettings object with custom timeout.

**Namespace:** TwinCAT.Ads
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

```csharp
public AdsClientSettings(
    int timeout
)
```

Parameters

timeout Type: System.Int32
The timeout of the AdsClient [154] in milliseconds.

Reference

AdsClientSettings Class [344]
TwinCAT.Ads Namespace [151]

6.2.2.2 AdsClientSettings Properties

The AdsClientSettings [344] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![CompatibilityDefault] [346]</td>
<td>Compatibility settings object</td>
</tr>
<tr>
<td>![Default] [347]</td>
<td>Gets the default settings (Default interceptors, Timeout 5000 ms)</td>
</tr>
<tr>
<td>![FastWriteThrough] [347]</td>
<td>Gets a Settings object that configures the AdsClient for FastWriteThrough</td>
</tr>
<tr>
<td>![Timeout] [348]</td>
<td>The communication Timeout that is set initially on the AdsClient [154]</td>
</tr>
</tbody>
</table>

Reference

AdsClientSettings Class [344]
TwinCAT.Ads Namespace [151]

6.2.2.2.1 AdsClientSettings.CompatibilityDefault Property

Compatibility settings object

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static AdsClientSettings CompatibilityDefault { get; }
```
Property Value

Type: `AdsClientSettings`[344]
The settings object.

Remarks

The compatibility settings initialize the AdsClient the same way as it is done in earlier versions of the TwinCAT.Ads.dll (earlier than Version 4.2)

- **All**[1053]
  - No FailFastHandlerInterceptor active.
  - Default communication timeout 5000ms.
  - Synchronized Notifications.

Reference

`AdsClientSettings Class`[344]

TwinCAT.Ads Namespace[151]

6.2.2.2.2 AdsClientSettings.Default Property

Gets the default settings (Default Interceptors, Timeout 5000 ms)

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

**C#**

```csharp
public static AdsClientSettings Default { get; }
```

Property Value

Type: `AdsClientSettings`[344]
The default.

Remarks

Creates an settings object, with specification for **All**[1053] and FailFastHandlerInterceptor.

- **All**[1053]
  - FailFastHandlerInterceptor is active.
  - Default communication timeout 5000ms.
  - Not synchronized Notifications.

Reference

`AdsClientSettings Class`[344]

TwinCAT.Ads Namespace[151]

6.2.2.2.3 AdsClientSettings.FastWriteThrough Property

Gets a Settings object that configures the AdsClient for FastWriteThrough
Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static AdsClientSettings FastWriteThrough { get; }
```

Property Value

Type: AdsClientSettings

Client settings for a fast write through (with 200 ms Timeout).

Remarks

The settings typically can be used for polling clients, where the "FailFast" feature will be bypassed. That means, that communication fails doesn't trigger the FailFast interceptor and every Request will go out via ADS. This has the Drawback that communication Timeouts are longer and subsequent timeouts block the ADS mailbox (with the danger of overflows). So use this setting with care for specific purposes and should not be used for standard communication.

- No FailFastHandlerInterceptor active.
- Default communicationtimeout 200ms.
- Not synchronized Notifications.

Reference

AdsClientSettings Class

TwinCAT.Ads Namespace

6.2.2.4 AdsClientSettings.Timeout Property

The communication Timeout that is set initially on the AdsClient

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int Timeout { get; }
```

Property Value

Type: Int32

The timeout.

Reference

AdsClientSettings Class

TwinCAT.Ads Namespace
6.2.2.3 AdsClientSettings Methods

The AdsClientSettings [344] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
</tbody>
</table>

Reference

AdsClientSettings Class [344]

TwinCAT.Ads Namespace [151]

6.2.3 AdsCommandId Enumeration

AdsCommandId Enumeration

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public enum AdsCommandId
```

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>None / Uninitialized</td>
</tr>
<tr>
<td>Invalid</td>
<td>0</td>
<td>Invalid</td>
</tr>
<tr>
<td>ReadDeviceInfo</td>
<td>1</td>
<td>ReadDeviceInfo command</td>
</tr>
<tr>
<td>Read</td>
<td>2</td>
<td>Read Command</td>
</tr>
<tr>
<td>Write</td>
<td>3</td>
<td>Write Command</td>
</tr>
<tr>
<td>ReadState</td>
<td>4</td>
<td>ReadState Command</td>
</tr>
<tr>
<td>WriteControl</td>
<td>5</td>
<td>WriteControl Command</td>
</tr>
<tr>
<td>AddNotification</td>
<td>6</td>
<td>AddNotification Command</td>
</tr>
<tr>
<td>DeleteNotification</td>
<td>7</td>
<td>DeleteNotification Command</td>
</tr>
<tr>
<td>Notification</td>
<td>8</td>
<td>Notification event.</td>
</tr>
<tr>
<td>ReadWrite</td>
<td>9</td>
<td>ReadWrite Command</td>
</tr>
</tbody>
</table>
6.2.4 AdsCommunicationStatistics Class

ADS Communication statistics

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.AdsCommunicationStatistics

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public class AdsCommunicationStatistics

The AdsCommunicationStatistics type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessWaitTime</td>
<td>Gets the wait time for the next access (Resurrection time) if in Lost.</td>
</tr>
<tr>
<td>ConnectionActiveSince</td>
<td>Gets the UTC time of the last connection activation.</td>
</tr>
<tr>
<td>ConnectionEstablishedAt</td>
<td>Gets the UTC time when the current connection was established.</td>
</tr>
<tr>
<td>ConnectionLostCount</td>
<td>Gets the connection lost count.</td>
</tr>
<tr>
<td>ConnectionLostTime</td>
<td>Gets the UTC connection lost time.</td>
</tr>
<tr>
<td>ConnectionResurrections</td>
<td>Gets the number of resurrections on the AdsConnection.</td>
</tr>
<tr>
<td>ErrorsSinceLastSucceeded</td>
<td>Gets the error count since last access (UTC).</td>
</tr>
<tr>
<td>LastSucceededAccess</td>
<td>Gets the UTC time of the last succeeded access.</td>
</tr>
<tr>
<td>Resurrections</td>
<td>Gets the number of Resurrections of this Session.</td>
</tr>
<tr>
<td>SessionEstablishedAt</td>
<td>Gets the UTC time when the session was established.</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### TC1000 Version: 1.1

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TotalCycles</td>
<td>Gets the total cycles.</td>
</tr>
<tr>
<td>TotalErrors</td>
<td>Gets the total error count.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Remarks

These statistics can be used for communication diagnosis. They contain Error/Succeed counts as well as Resurrection infos.

### Reference

TwinCAT.Ads Namespace [151]

#### 6.2.4.1 AdsCommunicationStatistics Properties

The AdsCommunicationStatistics [350] type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessWaitTime</td>
<td>Gets the wait time for the next access (Resurrection time) if in Lost.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from AdsConnection.)</td>
</tr>
<tr>
<td>ConnectionActiveSince</td>
<td>Gets the UTC time of the last connection activation.</td>
</tr>
<tr>
<td>ConnectionEstablishedAt</td>
<td>Gets the UTC time when the current connection was established.</td>
</tr>
<tr>
<td>ConnectionLostCount</td>
<td>Gets the connection lost count.</td>
</tr>
<tr>
<td>ConnectionLostTime</td>
<td>Gets the UTC connection lost time.</td>
</tr>
<tr>
<td>ConnectionResurrections</td>
<td>Gets the number of resurrections on the AdsConnection.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>ErrorsSinceLastSucceeded</td>
<td>Gets the error count since last access (UTC)</td>
</tr>
<tr>
<td>LastSucceededAccess</td>
<td>Gets the UTC time of the last succeeded access.</td>
</tr>
<tr>
<td>Resurrections</td>
<td>Gets the number of Resurrections of this Session.</td>
</tr>
<tr>
<td>SessionEstablishedAt</td>
<td>Gets the UTC time when the session was established.</td>
</tr>
<tr>
<td>TotalCycles</td>
<td>Gets the total cycles.</td>
</tr>
<tr>
<td>TotalErrors</td>
<td>Gets the total error count.</td>
</tr>
</tbody>
</table>

**Reference**

*AdsCommunicationStatistics Class [350]*  
TwinCAT.Ads Namespace [151]

### 6.2.4.1.1 AdsCommunicationStatistics.AccessWaitTime Property

Gets the wait time for the next access (Resurrection time) if in Lost [67].

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public TimeSpan AccessWaitTime { get; }
```

**Property Value**

Type: TimeSpan  
The wait time if in Lost [67] otherwise **TimeSpan.Zero**.

**Reference**

*AdsCommunicationStatistics Class [350]*  
TwinCAT.Ads Namespace [151]

### 6.2.4.1.2 AdsCommunicationStatistics.ConnectionActiveSince Property

Gets the UTC time of the last conenction activation.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
6.2.4.1.3 AdsCommunicationStatistics.ConnectionEstablishedAt Property

Gets the UTC time when the current connection was established.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Nullable<DateTimeOffset> ConnectionEstablishedAt { get; }
```

**Property Value**

Type: Nullable DateTimeOffset.
The connection established at.

**Reference**

AdsCommunicationStatistics Class [350]

TwinCAT.Ads Namespace [151]

6.2.4.1.4 AdsCommunicationStatistics.ConnectionLostCount Property

Gets the connection lost count.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int ConnectionLostCount { get; }
```

**Property Value**

Type: Int32
The connection lost count.
6.2.4.1.5 AdsCommunicationStatistics.ConnectionLostTime Property

Gets the UTC connection lost time.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

public Nullable<DateTimeOffset> ConnectionLostTime { get; }

**Property Value**

Type: Nullable DateTimeOffset
The connection lost time.

**Reference**

AdsCommunicationStatistics Class [350]
TwinCAT.Ads Namespace [151]

6.2.4.1.6 AdsCommunicationStatistics.ConnectionResurrections Property

Gets the number of resurrections on the AdsConnection [357]

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

public int ConnectionResurrections { get; }

**Property Value**

Type: Int32
The resurrections.

**Reference**

AdsCommunicationStatistics Class [350]
TwinCAT.Ads Namespace [151]

6.2.4.1.7 AdsCommunicationStatistics.ErrorsSinceLastSucceeded Property

Gets the error count since last access (UTC)
### AdsCommunicationStatistics.ErrorsSinceLastSucceeded Property

Gets the error count since last access.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**C#**
```csharp
public int ErrorsSinceLastSucceeded { get; }
```

**Property Value**

Type: `Int32`

The error count since last access.

**Reference**

AdsCommunicationStatistics Class [350]

TwinCAT.Ads Namespace [151]

### AdsCommunicationStatistics.LastSucceededAccess Property

Gets the UTC time of the last succeeded access.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**C#**
```csharp
public Nullable<DateTimeOffset> LastSucceededAccess { get; }
```

**Property Value**

Type: `Nullable DateTimeOffset`

The last succeeded access.

**Reference**

AdsCommunicationStatistics Class [350]

TwinCAT.Ads Namespace [151]

### AdsCommunicationStatistics.Resurrections Property

Gets the number of Resurrections of this Session.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**C#**
```csharp
public int Resurrections { get; }
```
6.2.4.1.10 AdsCommunicationStatistics.SessionEstablishedAt Property

Gets the UTC time when the session was established.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public DateTimeOffset SessionEstablishedAt { get; }

Property Value

Type: DateTimeOffset
The session established at.

Reference

AdsCommunicationStatistics Class [350]
TwinCAT.Ads Namespace [151]

6.2.4.1.11 AdsCommunicationStatistics.TotalCycles Property

 Gets the total cycles.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public int TotalCycles { get; }

Property Value

Type: Int32
The total cycles.

Reference

AdsCommunicationStatistics Class [350]
TwinCAT.Ads Namespace [151]
6.2.4.1.12 AdsCommunicationStatistics.TotalErrors Property

Gets the total error count.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```c#
public int TotalErrors { get; }
```

**Property Value**

Type: Int32
The total error count.

**Reference**

AdsCommunicationStatistics Class [350]

TwinCAT.Ads Namespace [151]

6.2.4.2 AdsCommunicationStatistics Methods

The AdsCommunicationStatistics [350] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from System.Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>System.Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from System.Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from System.Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current System.Object. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>System.Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>System.Object.)</td>
</tr>
</tbody>
</table>

**Reference**

AdsCommunicationStatistics Class [350]

TwinCAT.Ads Namespace [151]

6.2.5 AdsConnection Class

ADS Connection class
Inheritance Hierarchy

System.Object
   TwinCAT.Ads.AdsConnection

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public sealed class AdsConnection : IAdsConnection,
   IConnection, IConnectionStateProvider, IAdsNotifications, IAdsSymbolicAccess, IAdsAnyAccess,
   IAdsHandle, IAdsReadWrite2, IAdsReadWrite, IAdsStateProvider, IAdsStateControl,
   IAdsSymbolChangedProvider, IAdsRpcInvoke, IAdsReadWriteTimeoutAccess, IAdsStateControlTimeout, I
Disposable

The AdsConnection type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessWaitTime</td>
<td>Gets the access wait time.</td>
</tr>
<tr>
<td>ActiveSince</td>
<td>Gets the UTC time when the last active/resurrected Connection was established</td>
</tr>
<tr>
<td>Address</td>
<td>Gets the AmsAddress of the ADS server.</td>
</tr>
<tr>
<td>ClientAddress</td>
<td>Get the AmsAddress of the ADS client.</td>
</tr>
<tr>
<td>ConnectionEstablishedAt</td>
<td>Gets the UTC time when the Connection was originally established.</td>
</tr>
<tr>
<td>ConnectionLostCount</td>
<td>Gets the connection lost count.</td>
</tr>
<tr>
<td>ConnectionLostTime</td>
<td>Gets the connection lost time.</td>
</tr>
<tr>
<td>ConnectionState</td>
<td>Gets the current Connection state of the AdsConnection</td>
</tr>
<tr>
<td>DefaultValueEncoding</td>
<td>Gets the default value encoding.</td>
</tr>
<tr>
<td>Disposed</td>
<td>Gets a value indicating whether this AdsConnection is disposed.</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the AdsConnection identifier.</td>
</tr>
<tr>
<td>IsActive</td>
<td>Gets a value indicating whether communication is in active state</td>
</tr>
<tr>
<td>IsConnected</td>
<td>Gets a value indicating whether the local ADS port was opened successfully.</td>
</tr>
<tr>
<td>IsLocal</td>
<td>Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsLost</td>
<td>Gets a value indicating whether the communication is in lost / open state</td>
</tr>
<tr>
<td>IsReconnecting</td>
<td>Gets a value indicating whether communication is ready for reconnecting</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of this AdsConnection.</td>
</tr>
<tr>
<td>ResurrectingTries</td>
<td>Gets the number of tries to resurrect the AdsConnection.</td>
</tr>
<tr>
<td>Resurrections</td>
<td>Gets the number of succeeded connection resurrections.</td>
</tr>
<tr>
<td>Session</td>
<td>Gets the Session object of the AdsConnection object.</td>
</tr>
<tr>
<td>State</td>
<td>Gets the current ConnectionState.</td>
</tr>
<tr>
<td>Timeout</td>
<td>Gets the timeout (in milliseconds)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotification(String, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32.) [409]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [566] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [411]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [566] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32.) [412]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, Int32., CancellationToken) [413]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [566] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, Int32., CancellationToken) [415]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [566] event.</td>
</tr>
<tr>
<td>CleanupSymbolTable [416]</td>
<td>Clears the internal symbol cache.</td>
</tr>
<tr>
<td>Close [417]</td>
<td>Closes the AdsConnection</td>
</tr>
<tr>
<td>Connect [417]</td>
<td>(Re)Connects the IConnection [74] when disconnected.</td>
</tr>
<tr>
<td>CreateSymbolLoader [417]</td>
<td>Creates a new instance of the Symbol loader [1383] with the specified mode.</td>
</tr>
<tr>
<td>CreateVariableHandle [429]</td>
<td>Generates a unique handle for an ADS variable.</td>
</tr>
<tr>
<td>CreateVariableHandleAsync [429]</td>
<td>Determines the Symbol handle by its instance path asynchronously.</td>
</tr>
<tr>
<td>DeleteDeviceNotification [430]</td>
<td>Deletes an existing notification.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>DeleteDeviceNotificationAsync</td>
<td>Deletes a registered notification asynchronously.</td>
</tr>
<tr>
<td>DeleteVariableHandle</td>
<td>Releases the handle of a ADS variable again.</td>
</tr>
<tr>
<td>DeleteVariableHandleAsync</td>
<td>Releases the handle of a ADS variable again (asynchronously)</td>
</tr>
<tr>
<td>Disconnect</td>
<td>Disconnects this IConnection.</td>
</tr>
<tr>
<td>Dispose</td>
<td>Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object.)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object.)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object.)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, String, Object, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>Read(UInt32, Memory)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>Read(UInt32, UInt32, Memory)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>Read(UInt32, UInt32, Memory, Void)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, Int32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type, Int32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, Int32, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, Int32, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, Int32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, CancellationToken)</td>
<td>Reads data synchronously from an ADS device.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Int32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Int32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Int32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Int32, Encoding)</td>
<td>Reads a string from the specified symbol/variable.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Int32, Encoding, CancellationToken)</td>
<td>Reads a string asynchronously from the specified symbol/variable</td>
</tr>
<tr>
<td>ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken)</td>
<td>Reads any string as an asynchronous operation.</td>
</tr>
<tr>
<td>ReadAsync(UInt32, Memory, Void)</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadAsync(UInt32, UInt32, Memory, Void)</td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>ReadDataType</td>
<td>read data type as an asynchronous operation.</td>
</tr>
<tr>
<td>ReadDeviceInfo</td>
<td>Reads the identification and version number of an ADS server.</td>
</tr>
<tr>
<td>ReadDeviceInfoAsync</td>
<td>Reads the identification and version number of an ADS server.</td>
</tr>
<tr>
<td>ReadState</td>
<td>Reads the ADS status and the device status from an ADS server.</td>
</tr>
<tr>
<td>ReadStateAsync</td>
<td>Reads the ADS State asynchronously</td>
</tr>
<tr>
<td>ReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>ReadSymbolAsync</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>ReadValue(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.</td>
</tr>
<tr>
<td>ReadValue(String, Type)</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>ReadValue.T.(String)</td>
<td>Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>ReadValue.T.(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>ReadValueAsync(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.</td>
</tr>
<tr>
<td>ReadValueAsync(String, Type, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>ReadValueAsync.T.(String, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>ReadValueAsync.T.(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously and returns it as an object.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>ReadWrite(UInt32, Memory, Void)</code></td>
<td>Registers for AdsStateChanged events as an asynchronous operation.</td>
</tr>
<tr>
<td><code>ReadWrite(UInt32, UInt32, Memory, Void)</code></td>
<td>Registers the symbol version changed asynchronously.</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td><code>TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td><code>TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32., UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32, UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>TryCreateVariableHandle</td>
<td>Read (determine) the Symbol handle by its name/path</td>
</tr>
<tr>
<td>TryDeleteDeviceNotification</td>
<td>Deletes a registered notification.</td>
</tr>
<tr>
<td>TryDeleteVariableHandle</td>
<td>Releases the specified symbol/variable handle synchronously.</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object)</td>
<td>Invokes the specified RPC Method.</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, Object)</td>
<td>Invokes the specified RPC Method.</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object.)</td>
<td>Invokes the rpc method.</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object.)</td>
<td>Invokes the rpc method.</td>
</tr>
<tr>
<td>TryRead(UInt32, Memory, Void)</td>
<td></td>
</tr>
<tr>
<td>TryRead(UInt32, UInt32, Memory, Void)</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryRead(UInt32, UInt32, Memory, Void, Byte)</td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>TryReadDataType</td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>TryReadState(StateInfo.)</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.</td>
</tr>
<tr>
<td>TryReadState(UInt32, StateInfo.)</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.</td>
</tr>
<tr>
<td>TryReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>TryReadValue(ISymbol, Object.)</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryReadValue(String, Type, Object.)</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
<tr>
<td>TryReadValue.T. (String, T.)</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
<tr>
<td>TryReadValue.T. (ISymbol, T.)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, Memory, Void, Byte)</td>
<td></td>
</tr>
<tr>
<td>TryReadWrite(UInt32, UInt32, Memory, Void, Byte)</td>
<td></td>
</tr>
<tr>
<td>TryReadWrite(UInt32, UInt32, Memory, Void, Byte, ReadOnlyMemory)</td>
<td></td>
</tr>
<tr>
<td>TryWrite(UInt32, ReadOnlyMemory)</td>
<td></td>
</tr>
<tr>
<td>TryWrite(UInt32, UInt32, ReadOnlyMemory)</td>
<td></td>
</tr>
<tr>
<td>TryWrite(UInt32, UInt32, ReadOnlyMemory, Void)</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, Int32)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo,ReadOnlyMemory)</td>
<td></td>
</tr>
<tr>
<td>TryWriteControl(StateInfo,ReadOnlyMemory,Void)</td>
<td></td>
</tr>
<tr>
<td>TryWriteControl(StateInfo,ReadOnlyMemory,Void)</td>
<td></td>
</tr>
<tr>
<td>TryWriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter</td>
</tr>
<tr>
<td></td>
<td>type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryWriteValue(ISymbol, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32,</td>
</tr>
<tr>
<td></td>
<td>Int32, Bool etc.) are supported. Array and structures are not supported.</td>
</tr>
<tr>
<td></td>
<td>If a string is passed as parameter, the method attempts to parse the string</td>
</tr>
<tr>
<td></td>
<td>according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>TryWriteValue.T.ISymbol, T</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter</td>
</tr>
<tr>
<td></td>
<td>type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>UnregisterAdsStateChangedAsync</td>
<td>Registers for AdsStateChanged events as an asynchronous operation.</td>
</tr>
<tr>
<td>UnregisterSymbolVersionChangedAsync</td>
<td>Unregisters the symbol version changed asynchronous.</td>
</tr>
<tr>
<td>Write(UInt32,ReadOnlyMemory)</td>
<td>Trigger Client Method/Command.</td>
</tr>
<tr>
<td>Write(UInt32, UInt32)</td>
<td>Trigger Client Method/Command.</td>
</tr>
<tr>
<td>Write(UInt32, UInt32, Int32)</td>
<td>Trigger Client Method/Command.</td>
</tr>
<tr>
<td>Write(UInt32, UInt32, ReadOnlyMemory)</td>
<td>Trigger Client Method/Command.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Write(UInt32, UInt32, ReadOnlyMemory, Void)</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object, Int32)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAny(UInt32, UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAny(UInt32, UInt32, Object, Int32)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, Int32, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, Int32, CancellationToken)</td>
<td>Writes any as an asynchronous operation.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32,UInt32, Object, Int32, CancellationToken)</td>
<td>Writes any string as an asynchronous operation.</td>
</tr>
<tr>
<td>WriteAnyStringAsync(String, String, Int32, Encoding, CancellationToken)</td>
<td>writes any string as an asynchronous operation.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteAnyStringAsync(UInt32, String, Int32, Encoding, CancellationToken)</td>
<td>write any string as an asynchronous operation.</td>
</tr>
<tr>
<td>WriteAsync(UInt32, ReadOnlyMemory, Void)</td>
<td>Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.</td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, CancellationToken)</td>
<td></td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void)</td>
<td></td>
</tr>
<tr>
<td>WriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>WriteControl(StateInfo, Int32)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>WriteControl(StateInfo, ReadOnlyMemory)</td>
<td></td>
</tr>
<tr>
<td>WriteControl(StateInfo, ReadOnlyMemory, Void)</td>
<td></td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, CancellationToken)</td>
<td>Changes the ADS status and device status of the ADS server asynchronously.</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void)</td>
<td></td>
</tr>
<tr>
<td>WriteSymbolAsync</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue(String, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValue(ISymbol, Object)</td>
<td></td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WriteValue.T.(String, T)</strong></td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td><strong>WriteValue.T.(ISymbol, T)</strong></td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td><strong>WriteValueAsync(ISymbol, Object, CancellationToken)</strong></td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td><strong>WriteValueAsync.T.(String, T, CancellationToken)</strong></td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td><strong>WriteValueAsync.T.(ISymbol, T, CancellationToken)</strong></td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AdsNotification</strong></td>
<td>Occurs when the ADS device sends a notification to the client.</td>
</tr>
<tr>
<td><strong>AdsNotificationError</strong></td>
<td>Occurs when a exception has occurred during notification management.</td>
</tr>
<tr>
<td><strong>AdsNotificationEx</strong></td>
<td>Occurs when the ADS device sends a notification to the client.</td>
</tr>
<tr>
<td><strong>AdsStateChanged</strong></td>
<td>Occurs when ADS State has been changed.</td>
</tr>
<tr>
<td><strong>AdsSumNotification</strong></td>
<td>Occurs when Notifications are send (bundled notifications).</td>
</tr>
<tr>
<td><strong>AdsSymbolVersionChanged</strong></td>
<td>Occurs when the symbol version has been changed.</td>
</tr>
<tr>
<td><strong>ConnectionStateChanged</strong></td>
<td>Occurs when connection status of the AdsConnection has been changed.</td>
</tr>
<tr>
<td><strong>RouterStateChanged</strong></td>
<td>Occurs when [router state changed].</td>
</tr>
</tbody>
</table>
### Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PollAdsState(I Observable&lt;Unit.&gt;)</code></td>
<td>Overloaded. Gets an observable sequence of <code>AdsState</code> via Polling. (Defined by <code>AdsClientExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollAdsState(TimeSpan)</code></td>
<td>Overloaded. Gets an observable sequence of <code>AdsState</code> via Polling. (Defined by <code>AdsClientExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollAdsStateAsync(I Observable&lt;Unit., CancellationToken&gt;)</code></td>
<td>Overloaded. Gets an observable sequence of <code>AdsState</code> via Polling. (Defined by <code>AdsClientExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollAdsStateAsync(TimeSpan, CancellationToken)</code></td>
<td>Overloaded. Gets an observable sequence of <code>AdsState</code> via Polling. (Defined by <code>AdsClientExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues(String, Type, I Observable&lt;Unit.&gt;)</code></td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues(String, Type, TimeSpan)</code></td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues(String, Type, .Int32., TimeSpan)</code></td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues(String, Type, I Observable&lt;Unit., Func.Exception, Object.&gt;)</code></td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues(String, Type, TimeSpan, Func.Exception, Object.&gt;)</code></td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues(String, Type, .Int32., I Observable&lt;Unit., Func.Exception, Object.&gt;)</code></td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues(String, Type, Int32., TimeSpan, Func.Exception, Object.&gt;)</code></td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues(String, Type, .Int32., I Observable&lt;Unit., Func.Exception, Object.&gt;)</code></td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues.(String, I Observable&lt;Unit.&gt;)</code></td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollValues.T.(String, TimeSpan) [1084]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, I.Observable.Unit, Func.Exception, T.) [1087]</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, TimeSpan, Func.Exception, T.) [1088]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, .Int32., I.Observable.Unit.) [1085]</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, .Int32., TimeSpan) [1086]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, .Int32., I.Observable.Unit, Func.Exception, T.) [1091]</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, .Int32., TimeSpan, Func.Exception, T.) [1092]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>WhenAdsStateChanges [1066]</td>
<td>Gets an observable sequence of AdsState [626]s. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol) [1068]</td>
<td>Overloaded. Gets an observable sequence of Notification [974]s. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection) [1069]</td>
<td>Overloaded. Gets an observable sequence of Notification [974] objects. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol, NotificationSettings) [1070]</td>
<td>Overloaded. Gets an observable sequence of SymbolValueNotification [1104]s. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection, NotificationSettings) [1071]</td>
<td>Overloaded. Gets an observable sequence of Notification [974] objects. (Defined by AdsClientExtensions [1056].)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### AdsConnection Properties

The `AdsConnection` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>WhenNotification(String, Type, NotificationSettings)</code></td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>WhenNotification.T.(String, NotificationSettings)</code></td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>WhenSymbolVersionChanges()</code></td>
<td>Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by <code>AdsClientExtensions</code>.)</td>
</tr>
<tr>
<td><code>WhenSymbolVersionChanges(IScheduler)</code></td>
<td>Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by <code>AdsClientExtensions</code>.)</td>
</tr>
<tr>
<td><code>WhenValueChanged()</code></td>
<td>Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by <code>ValueSymbolExtensions</code>.)</td>
</tr>
<tr>
<td><code>WriteValues.T.(String, IObservable.T.)</code></td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>WriteValues.T.(String, IObservable.T., Action.Exception.)</code></td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
</tbody>
</table>

### Remarks

The ADS Connection class represents an ADS Point-to-Point Connection between client and server. It is established by using the Connect method of the `AdsSession` object. An ADS Connection can have different `ConnectionStates`, which represent the state of the logical ADS connection.

### Reference

- TwinCAT.Ads Namespace
- TwinCAT.Ads.AdsSession
- TwinCAT.Ads.IAdsConnection
- System.IDisposable

#### 6.2.5.1 AdsConnection Properties

The `AdsConnection` type exposes the following members.
## TwinCAT.Ads Namespaces

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveSince [376]</td>
<td>Gets the UTC time when the last active/resurrected Connection was established</td>
</tr>
<tr>
<td>Address [377]</td>
<td>Gets the AmsAddress [648] of the ADS server.</td>
</tr>
<tr>
<td>ConnectionEstablishedAt [378]</td>
<td>Gets the UTC time when the Connection was originally established.</td>
</tr>
<tr>
<td>ConnectionLostCount [378]</td>
<td>Gets the connection lost count.</td>
</tr>
<tr>
<td>ConnectionLostTime [378]</td>
<td>Gets the connection lost time.</td>
</tr>
<tr>
<td>ConnectionState [379]</td>
<td>Gets the current Connection state of the AdsConnection [357]</td>
</tr>
<tr>
<td>DefaultValueEncoding [380]</td>
<td>Gets the default value encoding.</td>
</tr>
<tr>
<td>Disposed [381]</td>
<td>Gets a value indicating whether this AdsConnection [357] is disposed.</td>
</tr>
<tr>
<td>Id [381]</td>
<td>Gets the AdsConnection [357] identifier.</td>
</tr>
<tr>
<td>IsActive [382]</td>
<td>Gets a value indicating whether communication is in active state</td>
</tr>
<tr>
<td>IsConnected [382]</td>
<td>Gets a value indicating whether the local ADS port was opened successfully.</td>
</tr>
<tr>
<td>IsLocal [383]</td>
<td>Gets a value indicating whether the ADS client is connected to a ADS Server</td>
</tr>
<tr>
<td>IsLost [383]</td>
<td>Gets a value indicating whether the communication is in lost / open state</td>
</tr>
<tr>
<td>IsReconnecting [384]</td>
<td>Gets a value indicating whether communication is ready for reconnecting</td>
</tr>
<tr>
<td>Name [384]</td>
<td>Gets the name of this AdsConnection [357].</td>
</tr>
<tr>
<td>ResurrectingTries [384]</td>
<td>Gets the number of tries to resurrect the AdsConnection [357].</td>
</tr>
<tr>
<td>Resurrections [384]</td>
<td>Gets the number of succeeded connection resurrections.</td>
</tr>
<tr>
<td>Timeout [386]</td>
<td>Gets the timeout (in milliseconds)</td>
</tr>
</tbody>
</table>
6.2.5.1.1 AdsConnection.AccessWaitTime Property

Gets the access wait time.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public TimeSpan AccessWaitTime { get; }
```

**Property Value**

**Type:** TimeSpan

The access wait time.

**Remarks**

Gets the Wait Time until the next communication try will be done. This time is calculated as follows:

\[\text{ResurrectionTime} - (\text{DateTime.Now} - \text{ConnectionLostTime})\]

**Reference**

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

AdsConnection.ConnectionLostTime [378]

SessionSettings.ResurrectionTime [1039]

6.2.5.1.2 AdsConnection.ActiveSince Property

Gets the UTC time when the last active/resurrected Connection was established

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Nullable<DateTimeOffset> ActiveSince { get; }
```

**Property Value**

**Type:** Nullable DateTimeOffset

The active since.
6.2.5.1.3  AdsConnection.Address Property

Gets the AmsAddress [648] of the ADS server.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public AmsAddress Address { get; }

Property Value

Type: AmsAddress [648]
The server address.

Implements

IAdsConnection.Address [781]

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

6.2.5.1.4  AdsConnection.ClientAddress Property

Get the AmsAddress [648] of the ADS client.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public AmsAddress ClientAddress { get; }

Property Value

Type: AmsAddress [648]
The client address.

Implements

IAdsConnection.ClientAddress [781]
6.2.5.1.5  AdsConnection.ConnectionEstablishedAt Property

Gets the UTC time when the Connection was originally established.

**Namespace:**  TwinCAT.Ads

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public Nullable<DateTimeOffset> ConnectionEstablishedAt { get; }
```

**Property Value**

Type: Nullable DateTimeOffset

The connection established at.

**Reference**

AdsConnection Class [ 357 ]

TwinCAT.Ads Namespace [ 151 ]

6.2.5.1.6  AdsConnection.ConnectionLostCount Property

Gets the connection lost count.

**Namespace:**  TwinCAT.Ads

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public int ConnectionLostCount { get; }
```

**Property Value**

Type: Int32

The connection lost count.

**Reference**

AdsConnection Class [ 357 ]

TwinCAT.Ads Namespace [ 151 ]

6.2.5.1.7  AdsConnection.ConnectionLostTime Property

Gets the connection lost time.
**Property Value**

**Type:** Nullable\(\)\ DosDateTime\(\)\ .
The connection lost time.

**Reference**

AdsConnection Class \[357\]

TwinCAT.Ads Namespace \[151\]

**6.2.5.1.8 AdsConnection\(\)\ .ConnectionState Property**

Gets the current Connection state of the AdsConnection \[357\]

**Namespace:** TwinCAT.Ads \[151\]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

public ConnectionState ConnectionState { get; }  

**Property Value**

**Type:** ConnectionState \[67\]
The state of the connection.

**Implements**

IConnectionStateProvider\(\)\ .ConnectionState \[85\]

**Remarks**

The Connection state changes only if the IConnection \[74\] is established / shut down or active communication is triggered by the User of the IConnection \[74\] object.

**Examples**

The following sample shows how to keep the ConnectionState updated by triggering ADS Communication.

**Trigger ConnectionState changes in WPF Applications**

```csharp
private DispatcherTimer _timer = null;
private AdsSession _session = null;
//private AdsConnection _connection = null;

private void Window_Loaded(object sender, RoutedEventArgs e) {

```
```csharp
_session = new AdsSession(AmsNetId.Local, 10000);
IConnection connection = _session.Connect();
tbConnectionState.Text = connection.ConnectionState.ToString();
_session.ConnectionStateChanged += _session_ConnectionStateChanged;
_timer = new DispatcherTimer();
_timer.Interval = TimeSpan.FromMilliseconds(200);
_timer.Tick += TimerOnTick;
_timer.Start();
}

private void Window_Unloaded(object sender, RoutedEventArgs e)
{
    _timer.Stop();
    _session.Dispose();
}

private void _session_ConnectionStateChanged(object sender, TwinCAT.ConnectionStateChangedEventArgs e)
{
    // ConnectionStateChanged will be triggered by communication Invokes
    tbConnectionState.Text = e.NewState.ToString();
}

private void TimerOnTick(object sender, EventArgs eventArgs)
{
    // The Timer Event will occur here in the UI thread because its an DispatcherTimer event!
    // An active ADS request will trigger Connection State periodically!
    StateInfo stateInfo;
    if (_session.Connection.TryReadState(out stateInfo) == AdsErrorCode.NoError)
    {
        tbAdsState.Text = stateInfo.AdsState.ToString();
    }
    else
    {
        tbAdsState.Text = "Invalid";
    }
}
```

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]
AdsConnection.ConnectionStateChanged [570]

6.2.5.1.9 AdsConnection.DefaultValueEncoding Property

Gets the default value encoding.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Encoding DefaultValueEncoding { get; }
```

Property Value

Type: Encoding
The default value encoding.
6.2.5.1.10  AdsConnection.Disposed Property

Gets a value indicating whether this AdsConnection is disposed.

Namespace:  TwinCAT.Ads
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool Disposed { get; }

Property Value

Type:  Boolean
true if disposed; otherwise, false.

Reference

AdsConnection Class[357]
TwinCAT.Ads Namespace[151]

6.2.5.1.11  AdsConnection.Id Property

Gets the AdsConnection identifier.

Namespace:  TwinCAT.Ads
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public int Id { get; }

Property Value

Type:  Int32
The identifier.

Reference

AdsConnection Class[357]
TwinCAT.Ads Namespace[151]
6.2.5.1.12 AdsConnection.IsActive Property

Gets a value indicating whether communication is in active state

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public bool IsActive { get; }
```

**Property Value**

Type: Boolean  
true if this instance is active; otherwise, false.

**Reference**

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

6.2.5.1.13 AdsConnection.IsConnected Property

Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public bool IsConnected { get; }
```

**Property Value**

Type: Boolean  
true if this instance is connected; otherwise, false.

**Implements**

IConnection.IsConnected [76]

**Reference**

AdsConnection Class [357]
6.2.5.1.14 AdsConnection.IsLocal Property

Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.

**Namespace:** TwinCAT.Ads [▸ 151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool IsLocal { get; }
```

**Property Value**

Type: Boolean
true if this instance is local; otherwise, false.

**Implements**

IAdsConnection.IsLocal [▸ 782]

**Reference**

AdsConnection Class [▸ 357]
TwinCAT.Ads Namespace [▸ 151]

6.2.5.1.15 AdsConnection.IsLost Property

Gets a value indicating whether the communication is in lost / open state

**Namespace:** TwinCAT.Ads [▸ 151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool IsLost { get; }
```

**Property Value**

Type: Boolean
true if this instance is lost; otherwise, false.

**Reference**

AdsConnection Class [▸ 357]
TwinCAT.Ads Namespace [▸ 151]
6.2.5.1.16 AdsConnection.IsReconnecting Property

Gets a value indicating whether communication is ready for reconnecting

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public bool IsReconnecting { get; }
```

**Property Value**

Type: **Boolean**

ture if this instance is reconnecting; otherwise, false.

**Reference**

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

---

6.2.5.1.17 AdsConnection.Name Property

Gets the name of this AdsConnection [357].

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public string Name { get; }
```

**Property Value**

Type: **String**

The name.

**Reference**

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

---

6.2.5.1.18 AdsConnection.ResurrectingTries Property

Gets the number of tries to resurrect the AdsConnection [357].

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public int ResurrectingTries { get; }
```

Property Value

Type: Int32
The number of tried resurrections of the [Connection](#74).

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

6.2.5.1.19 AdsConnection.Resurrections Property

Gets the number of succeeded connection resurrections.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int Resurrections { get; }
```

Property Value

Type: Int32
The resurrection count.

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

6.2.5.1.20 AdsConnection.Session Property

Gets the Session object of the AdsConnection [357] object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public ISession Session { get; }
```

Property Value

Type: ISession [88]
The client.
Implements

IConnection.Session

Reference

AdsConnection Class

TwinCAT.Ads Namespace

6.2.5.1.21 AdsConnection.State Property

Gets the current ConnectionState

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdcca3e72bc0ea15da1c14

Syntax

C#

public ConnectionState State { get; }

Property Value

Type: ConnectionState
The state.

Reference

AdsConnection Class

TwinCAT.Ads Namespace

6.2.5.1.22 AdsConnection.Timeout Property

Gets the timeout (in milliseconds)

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdcca3e72bc0ea15da1c14

Syntax

C#

public int Timeout { get; set; }

Property Value

Type: Int32
The timeout.

Implements

IConnection.Timeout
6.2.5.2 AdsConnection Methods

The AdsConnection type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotification(String, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type)</strong></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx[566] event.</td>
</tr>
<tr>
<td><strong>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32.)</strong></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.</td>
</tr>
<tr>
<td><strong>AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, Int32., CancellationToken)</strong></td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx[566] event.</td>
</tr>
<tr>
<td><strong>AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, Int32., CancellationToken)</strong></td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx[566] event.</td>
</tr>
<tr>
<td><strong>CleanupSymbolTable</strong></td>
<td>Clears the internal symbol cache.</td>
</tr>
<tr>
<td><strong>Close</strong></td>
<td>Closes the AdsConnection[357]</td>
</tr>
<tr>
<td><strong>Connect</strong></td>
<td>(Re)Connects the IConnection[74] when disconnected.</td>
</tr>
<tr>
<td><strong>CreateSymbolLoader</strong></td>
<td>Creates a new instance of the Symbol loader[1383] with the specified mode.</td>
</tr>
<tr>
<td><strong>CreateVariableHandle</strong></td>
<td>Generates a unique handle for an ADS variable.</td>
</tr>
<tr>
<td><strong>CreateVariableHandleAsync</strong></td>
<td>Determines the Symbol handle by its instance path asynchronously.</td>
</tr>
<tr>
<td><strong>DeleteDeviceNotification</strong></td>
<td>Deletes an existing notification.</td>
</tr>
<tr>
<td><strong>DeleteDeviceNotificationAsync</strong></td>
<td>Deletes a registered notification asynchronously.</td>
</tr>
<tr>
<td><strong>DeleteVariableHandle</strong></td>
<td>Releases the handle of a ADS variable again.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DeleteVariableHandleAsync</td>
<td>Releases the handle of a ADS variable again (asynchronously)</td>
</tr>
<tr>
<td>Disconnect</td>
<td>Disconnects this IConnection</td>
</tr>
<tr>
<td>Dispose</td>
<td>Performs application-defined tasks associated with freeing, releasing, or</td>
</tr>
<tr>
<td></td>
<td>resetting unmanaged resources.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object, Any TypeSpecifier, AnyTypeSpecifier, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, String, Object, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, String, Object, Any TypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>Read(UInt32, Memory)</code></td>
<td>[447] Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>Read(UInt32, UInt32, Memory)</code></td>
<td>[448] Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>Read(UInt32, UInt32, Memory, Void)</code></td>
<td>[448] Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny(UInt32, Type)</code></td>
<td>[451] Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny(UInt32, Type, Int32.)</code></td>
<td>[453] Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny(UInt32, UInt32, Type)</code></td>
<td>[455] Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny(UInt32, UInt32, Type, Int32.)</code></td>
<td>[456] Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny(UInt32, UInt32, Type, Int32., Int32)</code></td>
<td>[457] Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny.T.(UInt32)</code></td>
<td>[450] Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny.T. (UInt32, Int32.)</code></td>
<td>[450] Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny.T.(UInt32, UInt32, Int32.)</code></td>
<td>[452] Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyAsync(UInt 32, Type, CancellationToken)</code></td>
<td>[461] Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyAsync(UInt 32, Type, _Int32., CancellationToken)</code></td>
<td>[463] Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyAsync(UInt 32, UInt32, Type, CancellationToken)</code></td>
<td>[465] Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, Int32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, CancellationToken)</td>
<td>Reads data synchronously from an ADS device.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Int32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Int32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Int32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Int32, Encoding)</td>
<td>Reads a string from the specified symbol/variable.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Int32, Encoding)</td>
<td>Reads as string from a specified address.</td>
</tr>
<tr>
<td>ReadAnyAsyncAsync(UInt32, Int32, Encoding, CancellationToken)</td>
<td>Reads a string asynchronously from the specified symbol/variable</td>
</tr>
<tr>
<td>ReadAnyAsyncAsync(UInt32, UInt32, Int32, Encoding, CancellationToken)</td>
<td>reads any string as an asynchronous operation.</td>
</tr>
<tr>
<td>ReadDataType</td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadDataTypeAsync</td>
<td>read data type as an asynchronous operation.</td>
</tr>
<tr>
<td>ReadDeviceInfo</td>
<td>Reads the identification and version number of an ADS server.</td>
</tr>
<tr>
<td>ReadDeviceInfoAsync</td>
<td>Reads the identification and version number of an ADS server.</td>
</tr>
<tr>
<td>ReadState</td>
<td>Reads the ADS status and the device status from an ADS server.</td>
</tr>
<tr>
<td>ReadState(Int32)</td>
<td>Reads the ADS status and the device status from an ADS server.</td>
</tr>
<tr>
<td>ReadStateAsync</td>
<td>Read the ADS State asynchronously</td>
</tr>
<tr>
<td>ReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>ReadSymbolAsync</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>ReadValue(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.</td>
</tr>
<tr>
<td>ReadValue(String, Type)</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>ReadValue.T.(String)</td>
<td>Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>ReadValue.T.(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>ReadValueAsync(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.</td>
</tr>
<tr>
<td>ReadValueAsync(String, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>ReadValueAsync.T.(String, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>ReadValueAsync.T.(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously and returns it as an object.</td>
</tr>
<tr>
<td>ReadWrite(UInt32, Memory, Void)</td>
<td></td>
</tr>
<tr>
<td>ReadWrite(UInt32, UInt32, Memory, Void)</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadWrite(UInt32, UInt32, Memory, Void, Byte)</td>
<td>Registers for AdsStateChanged [567] events as an asynchronous operation.</td>
</tr>
<tr>
<td>ReadWriteAsync(UInt32, Memory, Void, Byte)</td>
<td>Registers the symbol version changed asynchronously.</td>
</tr>
<tr>
<td>RegisterSymbolVersionChangedAsync</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [564] event.</td>
</tr>
<tr>
<td>TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [564] event.</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32., UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32., UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [566] event.</td>
</tr>
<tr>
<td>TryCreateVariableHandle</td>
<td>Read (determine) the Symbol handle by its name/path</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryDeleteDeviceNotification [500]</td>
<td>Deletes a registered notification.</td>
</tr>
<tr>
<td>TryDeleteVariableHandle [501]</td>
<td>Releases the specified symbol/variable handle synchronously.</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object) [503]</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object) [505]</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object) [506]</td>
<td>Invokes the rpc method.</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object) [508]</td>
<td>Invokes the rpc method.</td>
</tr>
<tr>
<td>TryRead(UInt32, Memory, Void) [511]</td>
<td></td>
</tr>
<tr>
<td>TryRead(UInt32, UInt32, Memory, Void) [511]</td>
<td></td>
</tr>
<tr>
<td>TryRead(UInt32, UInt32, Memory, Void, Byte) [512]</td>
<td></td>
</tr>
<tr>
<td>TryReadDataType [513]</td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>TryReadState(StateInfo) [514]</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryReadState(Int32, StateInfo.)</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.</td>
</tr>
<tr>
<td>TryReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>TryReadValue(ISymbol, Object.)</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryReadValue(String, Type, Object.)</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
<tr>
<td>TryReadValue(T, ISymbol, T)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, Memory, Void, Byte)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>TryWrite(UInt32, ReadOnlyMemory)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, ReadOnlyMemory) [528]</td>
<td></td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, ReadOnlyMemory, Void) [528]</td>
<td></td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, ReadOnlyMemory, Void) [528]</td>
<td></td>
</tr>
<tr>
<td>TryWriteValue(String, Object) [529]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryWriteValue(ISymbol, Object) [531]</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>TryWriteValue.T(String, T) [530]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryWriteValue.T.ISymbol, T [532]</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>UnregisterSymbolVersionChangedAsync [533]</td>
<td>Unregisters the symbol version changed asynchronous.</td>
</tr>
<tr>
<td>Write(UInt32, ReadOnlyMemory) [535]</td>
<td></td>
</tr>
<tr>
<td>Write(UInt32, UInt32) [535]</td>
<td>Trigger Client Method/Command.</td>
</tr>
<tr>
<td>Write(UInt32, UInt32, Int32) [536]</td>
<td>Trigger Client Method/Command.</td>
</tr>
<tr>
<td>Write(UInt32, UInt32, ReadOnlyMemory) [537]</td>
<td></td>
</tr>
<tr>
<td>Write(UInt32, UInt32, ReadOnlyMemory, Void) [537]</td>
<td></td>
</tr>
<tr>
<td>Write(UInt32, Object) [538]</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object, Int32)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. [p. 539]</td>
</tr>
<tr>
<td>WriteAny(UInt32, UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device. [p. 540]</td>
</tr>
<tr>
<td>WriteAny(UInt32, UInt32, Object, Int32)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. [p. 541]</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. [p. 542]</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, Int32, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. [p. 543]</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. [p. 545]</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, Int32, CancellationToken)</td>
<td>write any as an asynchronous operation. [p. 545]</td>
</tr>
<tr>
<td>WriteAnyStringAsync(String, String, Int32, Encoding, CancellationToken)</td>
<td>write any string as an asynchronous operation. [p. 547]</td>
</tr>
<tr>
<td>WriteAnyStringAsync(UInt32, String, Int32, Encoding, CancellationToken)</td>
<td>write any string as an asynchronous operation. [p. 548]</td>
</tr>
<tr>
<td>WriteAsync(UInt32, ReadOnlyMemory, Void)</td>
<td>write any as an asynchronous operation. [p. 549]</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, CancellationToken)</td>
<td>Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.</td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>WriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>WriteControl(StateInfo, Int32)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>WriteControl(StateInfo, ReadOnlyMemory)</td>
<td>Changes the ADS status and device status of the ADS server asynchronously.</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, CancellationToken)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteSymbolAsync</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue(String, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValue(ISymbol, T)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
</tbody>
</table>
### TwinCAT Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValueAsync(ISymbol, Object, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValueAsync&lt;T&gt;(String, T, CancellationToken)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValueAsync&lt;T&gt;(ISymbol, T, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
</tbody>
</table>

### Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollAdsState(IObservable&lt;Unit.)</td>
<td>Overloaded. Gets an observable sequence of AdsState [626]s via Polling. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>PollAdsStateAsync(IObservable&lt;Unit., CancellationToken)</td>
<td>Overloaded. Gets an observable sequence of AdsState [626]s via Polling. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>PollValues(String, Type, IObserverable.Unit.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, TimeSpan)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan, Func.Exception, Object.)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [[1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, IObservable.Unit, Func.Exception, Object.)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [[1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, TimeSpan, Func.Exception, Object.)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [[1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, IObservable.Unit.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [[1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [[1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, Int32, IObservable.Unit, Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [[1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, Int32, TimeSpan, Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [[1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, Int32, IObservable.Unit.)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [[1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, Int32, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [[1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, Int32, IObservable.Unit, Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [[1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, Int32, TimeSpan, Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [[1075].)</td>
</tr>
<tr>
<td>WhenAdsStateChanges</td>
<td>Gets an observable sequence of AdsState [[626].s. (Defined by AdsClientExtensions [[1056].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WhenNotification(ISymbol)</td>
<td>Overloaded. Gets an observable sequence of <strong>Notification</strong> objects. (Defined by <strong>AdsClientExtensions</strong>.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection)</td>
<td>Overloaded. Gets an observable sequence of <strong>Notification</strong> objects. (Defined by <strong>AdsClientExtensions</strong>.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol, NotificationSettings)</td>
<td>Overloaded. Gets an observable sequence of <strong>SymbolValueNotification</strong>s. (Defined by <strong>AdsClientExtensions</strong>.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection, NotificationSettings)</td>
<td>Overloaded. Gets an observable sequence of <strong>Notification</strong> objects. (Defined by <strong>AdsClientExtensions</strong>.)</td>
</tr>
<tr>
<td>WhenNotification&lt;String, Type, NotificationSettings&gt;</td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by <strong>AnyTypeExtensions</strong>.)</td>
</tr>
<tr>
<td>WhenValueChanged</td>
<td>Observable sequence of <strong>Values</strong> driven by ADS Notifications on the specified symbol. (Defined by <strong>ValueSymbolExtensions</strong>.)</td>
</tr>
<tr>
<td>WriteValues&lt;T&gt;(String, IObservable&lt;T&gt;)</td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by <strong>AnyTypeExtensions</strong>.)</td>
</tr>
<tr>
<td>WriteValues&lt;T&gt;(String, IObservable&lt;T&gt;, Action.Exception)</td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by <strong>AnyTypeExtensions</strong>.)</td>
</tr>
</tbody>
</table>

**Reference**

**AdsConnection Class** [357]

**TwinCAT.Ads Namespace** [151]
### 6.2.5.2.1 AdsConnection.AddDeviceNotification Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://www.beckhoff.com" alt="AddDeviceNotification(String, Int32, NotificationSettings, Object)" /> ![402]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td><img src="https://www.beckhoff.com" alt="AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object)" /> ![403]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
</tbody>
</table>

#### Reference

- AdsConnection Class [357]
- TwinCAT.Ads Namespace [151]

#### AdsConnection.AddDeviceNotification Method (String, Int32, NotificationSettings, Object)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public uint AddDeviceNotification(
    string variableName,
    int dataSize,
    NotificationSettings settings,
    Object userData
)
```

#### Parameters

- **variableName**
  - Type: `System.String`
  - Name of the variable.

- **dataSize**
  - Type: `System.Int32`
  - Maximum amount of data in bytes to receive with this ADS Notification.

- **settings**
  - Type: `TwinCAT.Ads.NotificationSettings` [979]
  - The settings.

- **userData**
  - Type: `System.Object`
  - This object can be used to store user specific data (tag data)

#### Return Value

- Type: `UInt32`
  - The notification handle.
Implements

IAdsNotifications.AddDeviceNotification(String, Int32, NotificationSettings, Object) [844]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>[63]</td>
</tr>
</tbody>
</table>

Remarks

The dataSize Parameter defines the amount of bytes, that will be attached to the AdsNotification [564] as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [430] should always called when the notification is not used anymore.

Reference

AdsConnection Class [357]
AddDeviceNotification Overload [402]
TwinCAT.Ads Namespace [151]
AdsConnection.AdsNotification [564]
AdsConnection.DeleteDeviceNotification(UInt32) [430]
AddDeviceNotification Overload [844]
AddDeviceNotificationAsync Overload [847]
TryAddDeviceNotification Overload [859]

**AdsConnection.AddDeviceNotification Method (UInt32, UInt32, Int32, NotificationSettings, Object)**

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public uint AddDeviceNotification(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData
)
```
Parameters

- indexGroup: Type: `System.UInt32`
  Contains the index group number of the requested ADS service.

- indexOffset: Type: `System.UInt32`
  Contains the index offset number of the requested ADS service.

- dataSize: Type: `System.Int32`
  Maximum amount of data in bytes to receive with this ADS Notification.

- settings: Type: `TwinCAT.Ads.NotificationSettings`[979]
  The Notification settings.

- userData: Type: `System.Object`
  This object can be used to store user specific data (tag data)

Return Value

Type: `UInt32`
The notification handle.

Implements

- `IAdsNotifications.AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object)`[845]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>[63]</td>
</tr>
</tbody>
</table>

Remarks

The `dataSize` Parameter defines the amount of bytes, that will be attached to the `AdsNotification`[564] as value. Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotification(UInt32)`[430] should always called when the notification is not used anymore.

Reference

- AdsConnection Class [357]
- AddDeviceNotification Overload [402]
- TwinCAT.Ads Namespace [151]
- AdsConnection.DeleteDeviceNotification(UInt32) [430]
- AdsConnection.AdsNotification[564]
- AdsConnection.AdsNotificationError[565]
- AddDeviceNotification Overload [844]
- TryAddDeviceNotification Overload [859]
- AddDeviceNotificationAsync Overload [847]
## 6.2.5.2.2 AdsConnection.AddDeviceNotificationAsync Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event.</td>
</tr>
</tbody>
</table>

### Reference

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]
userData

Type: System.Object
This object can be used to store user specific data (tag data)

cancel

Type: System.Threading.CancellationToken
The Cancellation token.

Return Value

Type: Task.ResultHandle
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle type parameter contains the created handle (Handle) and the ErrorCode after execution.

Implements

IAdsNotifications.AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken)

Remarks

The dataSize Parameter defines the amount of bytes, that will be attached to the AdsNotification as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotificationAsync(UInt32, CancellationToken) should always be called when the notification is not used anymore.

Reference

AdsConnection Class
AddDeviceNotificationAsync Overload
TwinCAT.Ads Namespace
IAdsNotifications.AdsNotification
IAdsNotifications.DeleteDeviceNotificationAsync(UInt32, CancellationToken)
AddDeviceNotification Overload
AddDeviceNotificationAsync Overload
TryAddDeviceNotification Overload

AdsConnection.AddDeviceNotificationAsync Method (UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
## Syntax

**C#**

```csharp
public Task<ResultHandle> AddDeviceNotificationAsync(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    CancellationToken cancel
)
```

### Parameters

- **indexGroup**
  - Type: `System.UInt32`
  - The index group number of the requested ADS service.

- **indexOffset**
  - Type: `System.UInt32`
  - The index offset number of the requested ADS service.

- **dataSize**
  - Type: `System.Int32`
  - Maximum amount of data in bytes to receive with this ADS Notification.

- **settings**
  - Type: `TwinCAT.Ads.NotificationSettings`
  - The notification settings.

- **userData**
  - Type: `System.Object`
  - This object can be used to store user specific data (tag data)

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The Cancellation token.

### Return Value

- Type: `Task<ResultHandle>`
  - A task that represents the asynchronous 'AddDeviceNotification' operation. The `ResultHandle` type parameter contains the created handle (Handle) and the `ErrorCode` after execution.

### Implements

- `IAdsNotifications.AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)`

### Remarks

The `dataSize` parameter defines the amount of bytes, that will be attached to the `AdsNotification` as value. Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotificationAsync(UInt32, CancellationToken)` should always be called when the notification is not used anymore.

### Reference

- `AdsConnection Class`
- `AddDeviceNotificationAsync Overload`
- `TwinCAT.Ads Namespace`
- `AdsConnection.AdsNotification`
AdsConnection.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [431]

AddDeviceNotification Overload [844]
AddDeviceNotificationAsync Overload [847]
TryAddDeviceNotification Overload [859]

### AdsConnection.AddDeviceNotificationEx Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [566] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [566] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [566] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.</td>
</tr>
</tbody>
</table>

**Reference**

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

**AdsConnection.AddDeviceNotificationEx Method (String, NotificationSettings, Object, Type)**

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [566] event.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public uint AddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type
)
```

Parameters

- **symbolPath**
  - Type: `System.String`
  - Symbol/Instance path of the ADS variable.

- **settings**
  - Type: `TwinCAT.Ads.NotificationSettings` [\[.979\]}
  - The Notification settings.

- **userData**
  - Type: `System.Object`
  - This object can be used to store user specific data (tag data).

- **type**
  - Type: `System.Type`
  - Type of the object stored in the event argument ('AnyType')

Return Value

- Type: `UInt32`
- The notification handle.

Implements

- `IAdsNotifications.AddDeviceNotificationEx(String, NotificationSettings, Object, Type)` [\[.850\]}

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotification(UInt32)` [\[.430\] should always called when the notification is not used anymore.

Reference

- `AdsConnection Class` [\[.357\]}
- `AddDeviceNotificationEx Overload` [\[.408\]}
- `TwinCAT.Ads Namespace` [\[.151\]}
- `AdsConnection.AdsNotificationEx` [\[.566\]}
- `AdsConnection.DeleteDeviceNotification(UInt32)` [\[.430\]}
- `AddDeviceNotificationEx Overload` [\[.850\]}
- `AddDeviceNotificationExAsync Overload` [\[.855\]}
- `TryAddDeviceNotificationEx Overload` [\[.862\]}

**AdsConnection.AddDeviceNotificationEx Method (String, NotificationSettings, Object, Type, .Int32.)**

Connects a variable to the ADS client. The ADS client will be notified by the `AdsNotificationEx` [\[.566\] event.
Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public uint AddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args
)
```

Parameters

- **symbolPath**
  - Type: `System.String`
  - Symbol/Instance path of the ADS variable.

- **settings**
  - Type: `TwinCAT.Ads.NotificationSettings`[
  - The Notification settings.

- **userData**
  - Type: `System.Object`
  - This object can be used to store user specific data (tag data)

- **type**
  - Type: `System.Type`
  - Type of the object stored in the event argument ('AnyType')

- **args**
  - Type: `.System.Int32`
  - Additional arguments (for 'AnyType')

Return Value

Type: `UInt32`

The notification handle.

Implements

`IAdsNotifications.AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.)`[

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotification(UInt32)`[

Reference

- AdsConnection Class[
- TwinCAT.Ads Namespace[
- AdsConnection.AdsNotificationEx[
- AdsConnection.DeleteDeviceNotification(UInt32)[
- AddDeviceNotificationEx Overload[
- AddDeviceNotificationEx Async Overload[
- TryAddDeviceNotificationEx Overload[
**AdsConnection.AddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type)**

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [566] event.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public uint AddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type
)
```

**Parameters**

- **indexGroup**
  - Type: System.UInt32
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - Type: System.UInt32
  - Contains the index offset number of the requested ADS service.

- **settings**
  - Type: TwinCAT.Ads.NotificationSettings [979]
  - The Notification settings.

- **userData**
  - Type: System.Object
  - This object can be used to store user specific data (tag data)

- **type**
  - Type: System.Type
  - Type of the object stored in the event argument ('AnyType')

**Return Value**

- Type: UInt32
  - The notification handle.

**Implements**

IAdsNotifications.AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [852]

**Remarks**

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [430] should always be called when the notification is not used anymore.

**Reference**

AdsConnection Class [357]

AddDeviceNotificationEx Overload [408]

TwinCAT.Ads Namespace [151]

AdsConnection.DeleteDeviceNotification(UInt32) [430]

AdsConnection.AdsNotificationEx [566]
AdsConnection.AddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type, Int32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b436095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public uint AddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args
)
```

Parameters

- **indexGroup**
  Type: System.UInt32
  Contains the index group number of the requested ADS service.

- **indexOffset**
  Type: System.UInt32
  Contains the index offset number of the requested ADS service.

- **settings**
  Type: TwinCAT.Ads.NotificationSettings
  The Notification settings.

- **userData**
  Type: System.Object
  This object can be used to store user specific data.

- **type**
  Type: System.Type
  Type of the object stored in the event argument.

- **args**
  Type: System.Int32.
  Additional arguments for 'AnyType' types.

Return Value

Type: UInt32
The notification handle.

Implements

IAdsNotifications.AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32.)
Remarks

Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotification(UInt32)` should always be called when the notification is not used anymore.

Reference

AdsConnection Class [357]
AddDeviceNotificationEx Overload [408]
TwinCAT.Ads Namespace [151]
AdsConnection.AdsNotificationEx [566]
AddDeviceNotificationEx Overload [850]
AddDeviceNotificationExAsync Overload [855]
TryAddDeviceNotificationEx Overload [862]

6.2.5.2.4 AdsConnection.AddDeviceNotificationExAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="413" alt="AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, Int32, CancellationToken)" /></td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td><img src="415" alt="AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, Int32, CancellationToken)" /></td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

AdsConnection.AddDeviceNotificationExAsync Method (String, NotificationSettings, Object, Type, _Int32_, CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx event.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public Task<ResultHandle> AddDeviceNotificationExAsync(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

Parameters

- **symbolPath**: Type: `System.String`
The symbol/instance path of the ADS variable.

- **settings**: Type: `TwinCAT.Ads.NotificationSettings`[979]
The notification settings.

- **userData**: Type: `System.Object`
This object can be used to store user specific data (tag data)

- **type**: Type: `System.Type`
Type of the object stored in the event argument ('AnyType')

- **args**: Type: `System.Int32`
Additional arguments (for 'AnyType')

- **cancel**: Type: `System.Threading.CancellationToken`
The Cancellation token.

Return Value

Type: `Task<ResultHandle>`[1005]
A task that represents the asynchronous 'AddDeviceNotification' operation. The `ResultHandle`[1005] type parameter contains the created handle (`Handle`[1007]) and the `ErrorCode`[992] after execution.

Implements

- `IAdsNotifications.AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, Int32., CancellationToken)`[855]

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotificationAsync(UInt32, CancellationToken)`[431] should always be called when the notification is not used anymore.

Reference

- `AdsConnection Class`[357]
- `AddDeviceNotificationExAsync Overload`[413]
- `TwinCAT.Ads Namespace`[151]
- `AdsConnection.AdsNotificationEx`[566]
- `AdsConnection.DeleteDeviceNotificationAsync(UInt32, CancellationToken)`[431]
- `AddDeviceNotificationEx Overload`[431]
- `AddDeviceNotificationExAsync Overload`[855]
TryAddDeviceNotificationEx Overload [862]

**AdsConnection.AddDeviceNotificationExAsync Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken)**

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [566] event.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Task<ResultHandle> AddDeviceNotificationExAsync(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    CancellationToken cancel)
```

**Parameters**

- **indexGroup**  
  Type: System.UInt32  
  Contains the index group number of the requested ADS service.

- **indexOffset**  
  Type: System.UInt32  
  Contains the index offset number of the requested ADS service.

- **settings**  
  Type: TwinCAT.Ads.NotificationSettings [979]  
  The settings.

- **userData**  
  Type: System.Object  
  This object can be used to store user specific data.

- **type**  
  Type: System.Type  
  Type of the object stored in the event argument, only Primitive 'AnyTypes' allowed.

- **args**  
  Type: .System.Int32.  
  Additional arguments (for 'AnyType')

- **cancel**  
  Type: System.Threading.CancellationToken  
  The Cancellation token.

**Return Value**

Type: Task<ResultHandle> [1005].  
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [1005] type parameter contains the created handle (Handle) and the ErrorCode [992] after execution.

**Implements**

|AdsNotifications.AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [856]|
Remarks

If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive types (AnyType) are supported by this method. Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotificationAsync(UInt32, CancellationToken)` should always called when the notification is not used anymore.

Reference

AddDeviceNotificationExAsync Overload [▸ 413]

TwinCAT.Ads Namespace [▸ 151]

AdsConnection.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▸ 431]

AdsConnection.AdsNotificationEx [▸ 566]

AdsConnection.AdsNotificationError [▸ 565]

AddDeviceNotificationEx Overload [▸ 850]

TryAddDeviceNotificationEx Overload [▸ 862]

AddDeviceNotificationExAsync Overload [▸ 855]

6.2.5.2.5 AdsConnection.CleanupSymbolTable Method

Clears the internal symbol cache.

Namespace: TwinCAT.Ads [▸ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void CleanupSymbolTable()
```

Implements

IAdsSymbolicAccess.CleanupSymbolTable [▸ 942]

Remarks

Previously stored symbol information is cleared. As a consequence the symbol information must be obtained from the ADS server again if accessed, which which needs an extra ADS round trip.

Reference

AddDeviceNotificationExAsync Overload [▸ 413]

TwinCAT.Ads Namespace [▸ 151]
6.2.5.2.6 AdsConnection.Close Method

Closes the AdsConnection [151]

Namespace: TwinCAT.Ads [151]

Syntax

C#

```csharp
public void Close()
```

Implements

IConnection.Close [78]

Reference

AdsConnection Class [151]
TwinCAT.Ads Namespace [151]

6.2.5.2.7 AdsConnection.Connect Method

(Re)Connects the IConnection [74] when disconnected.

Namespace: TwinCAT.Ads [151]

Syntax

C#

```csharp
public bool Connect()
```

Return Value

Type: Boolean
true if the AdsConnection [151] is reconnected, false otherwise.

Implements

IConnection.Connect [78]

Reference

AdsConnection Class [151]
TwinCAT.Ads Namespace [151]

6.2.5.2.8 AdsConnection.CreateSymbolLoader Method

Creates a new instance of the Symbol loader [138] with the specified mode.
Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.dll

Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public IAdsSymbolLoader CreateSymbolLoader(IAds SymbolLoader, ISymbolLoaderSettings settings)

Parameters

- **session**
  - Type: **TwinCAT.ISession**
  - The session (for session orientated loads/symbols). Can be NULL if not present.

- **settings**
  - Type: **TwinCAT.ISymbolLoaderSettings**
  - The settings.

Return Value

- Type: **IAdsSymbolLoader**
  - The interface of the Symbol loader.

Exceptions

- **ObjectDisposedException**
- **ObjectDisposedException**

Remarks

The Symbol Loader (V2) supports the following modes. The flat mode organizes the Symbols in a flat list. At the beginning this List caches only the root symbol objects, which can be enumerated. To access the sub elements like structure fields or array elements use the SubSymbols collection. The property get accessor generates the subsymbols lazy on the fly (performance optimized) and stores them internally as weak reference (memory optimized). This mode is available in all .NET versions. The virtual tree mode organizes the Symbols hierarchically with parent-child relationships. That eases the access to the hierarchical structure but needs slightly more preprocessing of the data. This mode is available in all .NET Versions. The Dynamic tree mode organizes the Symbols hierarchically and (dynamically) creates struct members, array elements and enum fields on the fly. 'Dynamically' means here not only lazy creation like in Flat, but furthermore real creation of type safe .NET complex types/instances as representatives of the TwinCAT Symbol objects/types. This feature is only available on platforms that support the Dynamic Language Runtime (DLR); actually all .NET Framework Version larger than 4.0. Virtual instances means, that all Symbols are ordered within a tree structure. For that symbol nodes that are not located on a fixed address, a Virtual Symbol will be created. Setting the virtualInstance parameter to 'false' means, that the located symbols will be returned in a flattened list.

Examples

The following sample shows how to create a dynamic version of the SymbolLoader V2. The dynamic symbol loader makes use of the Dynamic Language Runtime (DLR) of the .NET Framework. That means Structures, Arrays and Enumeration types and instances are generated 'on-the-fly' during symbol Browsing. These created dynamic objects are a one to one representation of the Symbol Server target objects (e.g. the IEC61131 types on the PLC). Dynamic language features are only available from .NET4 upwards.
Dynamic Tree Mode

namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT;
    using TwinCAT.Ads;
    using TwinCAT.Ads.TypeSystem;
    using TwinCAT.Ads.ValueAccess;
    using TwinCAT.TypeSystem.
    using TwinCAT.TypeSystem.Generic;
    using TwinCAT.ValueAccess;

class SymbolBrowserProgramV2DynamicTree
{
    #region CODE_SAMPLE_SIMPLEDYNAMIC
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static async void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        CancellationTokenSource cancelSource = new CancellationTokenSource();
        CancellationToken cancel = cancelSource.Token;

        using (AdsClient client = new AdsClient())
        {
            // Connect to the target device
            client.Connect(address);

            // Usage of "dynamic" Type and Symbols (>= .NET4 only)
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
            IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);

            // Set the Default setting for Notifications
            dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransModeOnChange, 200, 2000);

            // Get the Symbols (Dynamic Symbols)
            var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);

            dynamic dynamicSymbols = resultSymbols.Symbols;
            dynamic adsPort = dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.AdsPort;

            // Access Main Symbol with Dynamic Language Runtime support (DLR)
            // Dynamically created property "Main"
            dynamic symMain = dynamicSymbols.Main;

            // Main is an 'VirtualSymbol' / Organizational unit that doesn't have a value
            // Calling ReadValue is not allowed
            //bool test = symMain.HasValue;
            //dynamic invalid = symMain.ReadValue();

            //Reading TaskInfo Value
            // With calling ReadValueAsync() a 'snapshot' of the Symbols Instance is taken (reading async)
            ResultReadValueAccess resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo.
            ReadValueAsync(cancel);

            dynamic vTaskInfoArray = resultRead.Value;
            dynamic symTaskInfo1 = dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo[1];

            // Getting CycleCount Symbol
            dynamic symCycleCount = symTaskInfo1.CycleCount;
    
    
}
// Take Snapshot value of the ApplicationInfo struct
resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.ReadValueAsync(cancel);
dynamic vAppInfo = resultRead.Value;

// Get the UTC Timestamp of the snapshot
DateTimeOffset timeStamp2 = vAppInfo.TimeStamp;

// Access the ProjectName of the ApplicationInfo Snapshot (type-safe!)
string projectNameValue = vAppInfo.ProjectName;

// Reading the CycleCount Value
resultRead = await symTaskInfo1.CycleCount.ReadValueAsync(cancel); // Taking a Value Snapshot
int cycleCountValue = (int)resultRead.Value;

// Registering for dynamic "ValueChanged" events for the Values
// Using Default Notification settings
symCycleCount.ValueChanged += new EventHandler<ValueChangedEventArgs>(cycleCount_ValueChanged);

// Override default notification settings
symTaskInfo1.NotificationSettings = new NotificationSettings(AdsTransMode.Cyclic, 500, 0);

// Register for ValueChanged event.
symTaskInfo1.ValueChanged += new EventHandler<ValueChangedEventArgs>(taskInfo1ValueChanged);

Thread.Sleep(10000); // Sleep main thread for 10 Seconds
}

Console.WriteLine("CycleCount Changed events received: {0}", _cycleCountEvents);
Console.WriteLine("TaskInfo1 Value Changed events received: {0}", _taskInfo1Events);

Console.WriteLine("Press [Enter] for leave:");
Console.ReadLine();

static object _notificationSynchronizer = new object();
static int _cycleCountEvents = 0;

/// <summary>
/// Handler function for the CycleCount ValueChanged event.
/// </summary>
/// <param name="sender">The sender.</param>
/// <param name="e">The e.</param>
static void cycleCount_ValueChanged(object sender, ValueChangedEventArgs e)
{
    lock (_notificationSynchronizer)
    {
        Interlocked.Increment(ref _cycleCountEvents);
        // val is a type safe value of int!
        dynamic val = e.Value;
        uint intVal = val;

        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert UTC to local time
        Console.WriteLine("CycleCount changed to: {0}, TimeStamp: {1}", intVal, changedTime.ToString("HH:mm:ss:fff");
    }
}

static int _taskInfo1Events = 0;

/// <summary>
/// Handler function for the TaskInfo ValueChanged event.
/// </summary>
/// <param name="sender">The sender.</param>
/// <param name="e">The e.</param>
static void taskInfo1ValueChanged(object sender, ValueChangedEventArgs e)
{
    lock (_notificationSynchronizer)
    {
        Interlocked.Increment(ref _taskInfo1Events);
        dynamic val = e.Value;
        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert to local time

        // Val is a during Runtime created struct type and contains
        // the same Properties as related PLC object.
        int cycleTime = val.CycleTime;
        Console.WriteLine("TaskInfo1Value changed TimeStamp: {0}", changedTime.ToString("HH:mm:ss:ff"));
    }
}
The following sample shows how to create a static (non dynamic) version of the SymbolLoader V2. The static symbol loader in version 2 is a nearly code compatible version of the Dynamic Loader, only the dynamic creation of objects is not available. The reason for supporting this mode is that .NET Framework Versions lower than Version 4.0 (CLR2) doesn't support the Dynamic Language Runtime (DLR). The SymbolLoader V2 static object is supported from .NET 2.0 on.

Virtual Tree Mode

```csharp
using System;
using System.Threading;
using System.Diagnostics;
using TwinCAT;
using TwinCAT.Ads;
using TwinCAT.TypeSystem;
using TwinCAT.Ads.ValueAccess;
using TwinCAT.Ads.TypeSystem;

namespace Sample
{
    class SymbolBrowserProgramV2VirtualTree
    {
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static void Main(string[] args)
        {
            ConsoleLogger logger = new ConsoleLogger();
            Console.WriteLine("\n");
            Console.WriteLine("Press [Enter] for start: ");
            Console.ReadLine();
            Stopwatch stopper = new Stopwatch();
            // logger.Active = false;
            stopper.Start();
            using (AdsClient client = new AdsClient())
            {
                //client.Synchronize = false;
                // Connect the AdsClient to the device target.
                client.Connect(address);
                // Creates the Symbol Objects as hierarchical tree
                SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree, ValueAccessMode.IndexGroupOffsetPreferred);
                ISymbolLoader symbolLoader = SymbolLoaderFactory.Create(client, settings);
                // Dump Datatypes from Target Device
                Console.WriteLine("Dumping '{0}' DataTypes:", symbolLoader.DataTypes.Count);
                foreach (IDataType type in symbolLoader.DataTypes)
                {
                    logger.DumpType(type);
                }
                Console.WriteLine("\n");
                // Dump Symbols from target device
                Console.WriteLine("Dumping '{0}' Symbols:", symbolLoader.Symbols.Count);
                foreach (ISymbol symbol in symbolLoader.Symbols)
                {
                    logger.DumpSymbol(symbol, 0);
                }
                stopper.Stop();
                TimeSpan elapsed = stopper.Elapsed;
            }
        }
    }
}
```
Examples

The SymbolLoader V2 static object is supported from .NET 2.0 on.

Flat Mode

using System;
using System.Diagnostics;
using System.Threading;
using TwinCAT;
using TwinCAT.Ads;
using TwinCAT.Ads.TypeSystem;
using TwinCAT.Ads.ValueAccess;
using TwinCAT.TypeSystem;
using TwinCAT.TypeSystem.Generic;

namespace Sample
{
    class SymbolBrowserProgramV2Flat
    {
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static void Main(string[] args)
        {
            ConsoleLogger logger = new ConsoleLogger();
            Console.WriteLine(""');
            Console.WriteLine("Press [Enter] for start:");
            Console.ReadLine();
            //logger.Active = false;
            Stopwatch stopper = new Stopwatch();
            // Parse the command line arguments
            AmsAddress address = ArgParser.Parse(args);
            stopper.Start();
            // Create the ADS Client
            using (AdsClient client = new AdsClient())
            {
                //client.Synchronize = false;
                // Connect to Address
                client.Timeout = 30000;
                client.Connect(address);

                SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolLoadMode.Flat, ValueAccessMode.IndexGroupOffsetPreferred);
                ISymbolLoader symbolLoader = SymbolLoaderFactory.Create(client, settings);

                // Dump Datatypes from Target Device
                Console.WriteLine(string.Format("Dumping '{0}' Datatypes:", symbolLoader.DataTypes.Count));
                foreach (IDataType type in symbolLoader.DataTypes)
                {
                    logger.DumpType(type);
                }
                Console.WriteLine(""');

                // Dump Symbols from target device
                Console.WriteLine(string.Format("Dumping '{0}' Symbols:", symbolLoader.Symbols.Count));
                foreach (ISymbol symbol in symbolLoader.Symbols)
                {
                    logger.DumpSymbol(symbol, 0);
                }
            }
        }
    }
}
Examples

Argument Parser

public static class ArgParser
{
    /// <summary>
    /// Parses the arguments.
    /// </summary>
    /// <param name="args">The arguments.</param>
    /// <returns>AmsAddress.</returns>
    public static AmsAddress Parse(string[] args)
    {
        AmsNetId netId = AmsNetId.Local;
        int port = 851;
        if (args != null)
        {
            if (args.Length > 0 && args[0] != null)
                netId = AmsNetId.Parse(args[0]);
            if (args.Length > 1 && args[1] != null)
                port = int.Parse(args[1]);
        }
        return new AmsAddress(netId, port);
    }
}

Dumping Symbols

/// <summary>
/// Console logger
/// </summary>
public class ConsoleLogger
{
    public ConsoleLogger()
    {
    }
    bool _active = true;
    /// <summary>
    /// Gets or sets a value indicating whether this ConsoleLogger is active.
    /// </summary>
    /// <value><c>true</c> if active; otherwise, <c>false</c>.</value>
    public bool Active
    {
        get { return _active; }
        set
        {
            _active = value;
        }
    }
    int _dataTypes = 0;
    /// <summary>
    /// Gets the number of dumped dataTypes.
    /// </summary>
    /// <value>The data types count.</value>
    public int DataTypesCount
    {
        get { return _dataTypes; }
    }
    int _symbols = 0;
    /// <summary>
/// Gets the number of dumped symbols
/// <summary>
/// The symbols count.</value>
/// </summary>
/// public int SymbolsCount
/// {
/// return _symbols;
/// }

/// Dumps the data type.
/// <summary>
/// <param name="dataType">Data Type.</param>
/// </summary>
/// public void DumpType(IDataType dataType)
/// {
/// WriteLine(string.Format("DataType: {0}, Category: {1}, Size: {2}\", dataType.Name, dataType.Category, dataType.Size));
/// switch (dataType.Category)
/// {
/// case DataTypeCategory.Alias:
/// IAliasType alias = (IAliasType)dataType;
/// WriteLine(GetPrefix(1) + string.Format("Alias BaseType: {0}\", alias.BaseTypeName));
/// break;
/// case DataTypeCategory.Enum:
/// //IEnumType<ushort> enumType = (IEnumType<ushort>)dataType;
/// IEnumType enumType = (IEnumType)dataType;
/// WriteLine(GetPrefix(1) + string.Format("Enum BaseType: {0}\", enumType.BaseTypeName));
/// foreach (IEnumValue enumValue in enumType.EnumValues)
/// {
/// WriteLine(GetPrefix(2) + string.Format("Name: {0}, Value: {1}\", enumValue.Name, enumValue.Primitive));
/// }
/// break;
/// case DataTypeCategory.Array:
/// IArrayType arrayType = (IArrayType)dataType;
/// int i = 0;
/// foreach (IDimension dim in arrayType.Dimensions)
/// {
/// WriteLine(GetPrefix(2) + string.Format("[0]: LowerBound: {1}, Elements: {2}\", i +
/// , dim.LowerBound, dim.ElementCount));
/// }
/// break;
/// case DataTypeCategory.Struct:
/// IStructType structType = (IStructType)dataType;
/// foreach (IMember member in structType.Members)
/// {
/// WriteLine(GetPrefix(2) + string.Format("Offset {0}: Name: {1}, Type: {2}\", member.Offset,
/// , member.InstanceName, member.TypeName));
/// }
/// break;
/// default:
/// break;
/// }
/// foreach (ITypeAttribute attribute in dataType.Attributes)
/// {
/// WriteLine(GetPrefix(1) + string.Format("{0}: {1}\", attribute.Name, attribute.Value));
/// }
/// if (!string.IsNullOrEmpty(dataType.Comment))
/// {
/// WriteLine(GetPrefix(1) + string.Format("Comment: {0}\", dataType.Comment));
/// }
/// IRpcCallableType rpcCallable = dataType as IRpcCallableType;
/// if (rpcCallable != null)
/// {
/// foreach (IRpcMethod rpcMethod in rpcCallable.RpcMethods)
/// {
/// if (string.IsNullOrEmpty(rpcMethod.Comment))
/// {
/// WriteLine(GetPrefix(1) + string.Format("Method: {0}\", rpcMethod));
/// }
/// else
/// {
/// WriteLine(GetPrefix(1) + string.Format("Method: {0}, Comment: {1}\", rpcMethod, rpcMethod.Comment));
/// }
/// }
TwinCAT.Ads Namespaces

...}

///// <summary>
///// Dumps the Datatype to Console
///// </summary>
///// <param name="dataType">DataType.</param>
///public void DumpType(ITcAdsDataType dataType)
///{
//    // Dump the Attributes (PLC Metadata)
//    foreach (ITypeAttribute attribute in dataType.Attributes)
//    {
//        WriteLine(GetPrefix(1) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Value));
//    }
//    // Dumps the Datatype to Console
//    WriteLine(string.Format("DataType: {0}, Category: {1}, Size: {2}", dataType.Name, dataType.Category, dataType.Size));
//    // if {dataType.BaseType} != null
//    if (dataType.BaseType != null)
//    {
//        WriteLine(GetPrefix(1) + string.Format("BaseType: {0}", dataType.BaseType));
//    } else {
//        switch (dataType.Category)
//        {
//            case DataTypeCategory.Enum:
//                foreach (IEnumValue enumValue in dataType.EnumValues)
//                {
//                    WriteLine(GetPrefix(2) + string.Format("Name: {0}, Value: {1}", enumValue.Name, enumValue.Primitive));
//                }
//                break;
//            case DataTypeCategory.Array:
//                int i = 0;
//                foreach (IDimension dim in dataType.Dimensions)
//                {
//                    WriteLine(GetPrefix(2) + string.Format("{0}: LowerBound: {1}, Elements: {2}", i++, dim.LowerBound, dim.ElementCount));
//                }
//                break;
//            case DataTypeCategory.Struct:
//                foreach (ITcAdsSubItem subItem in dataType.SubItems)
//                {
//                    WriteLine(GetPrefix(2) + string.Format("Offset {0}: Name: {1}, Type: {2}", subItem.Off set, subItem.SubItemsName, subItem.Name));
//                }
//                break;
//            default:
//                break;
//        }
//        _dataTypes++;
//    }
//}
/// <summary>
/// Dump Symbol
/// </summary>
/// <param name="symbol">The symbol.</param>
/// <param name="level">Output indentation level</param>
public void DumpSymbol(ISymbol symbol, int level)
{
    IDataType type = symbol.DataType as IDataType;
    foreach (ITypeAttribute attribute in symbol.Attributes)
    {
        WriteLine(GetPrefix(level) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Value));
    }
    WriteLine(GetPrefix(level) + string.Format("Instance {0}: " + symbol.InstanceName + ", Category: {1}, Size: {2}", symbol.InstanceName, symbol.TypeName, (IAdsSymbol)symbol).IndexGroup.ToString("x"), ((IAdsSymbol)symbol).IndexOffset.ToString("x"), symbol.Size));
    if (symbol.Category == DataTypeCategory.Array)
    {
        IArrayInstance arrInstance = (IArrayInstance)symbol;
        IArrayType arrType = (IArrayType)symbol.DataType;
    }
}
int count = 0;
level++;

foreach (ISymbol arrayElement in arrInstance.Elements)
{
    DumpSymbol(arrayElement, level);
    count++;
    if (count > 20) // Write only the first 20 to limit output
        break;
}
else if (symbol.Category == DataTypeCategory.Struct)
{
    IStructInstance structInstance = (IStructInstance)symbol;
    IStructType structType = (IStructType)symbol.DataType;
    level++;

    foreach (ISymbol member in structInstance.MemberInstances)
    {
        DumpSymbol(member, level);
    }
    _symbols++;
}

///// <summary>
///// Dumps the specified Symbol to the Console
///// </summary>
///// <param name="symbol">The symbol.</param>
///// <param name="level">The level.</param>
//public void DumpSymbol(IAdsSymbol2 symbol, int level)
//{
//    // Dump Attributes of the Symbol
//    foreach (ITypeAttribute attribute in symbol.Attributes)
//    {
//        WriteLine(GetPrefix(level) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Value));
//    }

//    // ITcAdsSymbolBrowser subSymbolProvider = (ITcAdsSymbolBrowser)symbol;
//    // Dump The Symbol
//    WriteLine(GetPrefix(level) + string.Format("{{ {0} : {1} }}", symbol.Name, symbol.TypeName, symbol.DataTypeId, symbol.IndexGroup.ToString("x"), symbol.IndexOffset.ToString("x"), subSymbolProvider.SubSymbols.Count, symbol.Size));
//    level++;
//    // Dump all SubSymbols with indentation
//    foreach (IAdsSymbol2 subSymbol in ((ITcAdsSymbolBrowser)symbol).SubSymbols)
//    {
//        DumpSymbol(subSymbol, level);
//    }
//    _symbols++;
//}

///// <summary>
///// Dump namespace.
///// </summary>
///// <param name="ns">The namespace.</param>
public void DumpNamespace(INamespace<IDataType> ns)
{
    WriteLine("Namespace: {0}, DataTypes: {1}", ns.Name, ns.DataTypes.Count);
    foreach (IDataType type in ns.DataTypes)
    {
        DumpType(type);
    }
}

///// <summary>
///// Get the indentation prefix
///// </summary>
///// <param name="level">The level.</param>
///// <returns>System.String.</returns>
public string GetPrefix(int level)
{
    //...
Examples

The following sample shows how to call (Remote Procedures / Methods) with Virtual Symbols

RPC Call in Virtual Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /*
            {attribute 'TcRpcEnable'}
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
            |l1 : INT := 0;
            |l2 : INT := 0;
            END_VAR
            */
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});

            // Browsing RpcMethods
            foreach (IRpcMethod method in main.RpcMethods)
            {
                // ...
            }
        }
    }
}
Examples

The following sample shows how to call (Remote Procedures / Methods) with Dynamic Symbols.

**RPC Call in Dynamic Mode**
	namespace Sample
	{
	using System;
	using System.Diagnostics;
	using System.Threading;
	using TwinCAT;
	using TwinCAT.Ads;
	using TwinCAT.Ads.TypeSystem;
	using TwinCAT.Ads.ValueAccess;
	using TwinCAT.TypeSystem;
	using TwinCAT.TypeSystem.Generic;

class RpcCallDynamicProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static async void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        CancellationTokenSource cancelSource = new CancellationTokenSource();
        CancellationToken cancel = cancelSource.Token;
        using (AdsClient client = new AdsClient())
        {
            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
            ISymbolLoader dynLoader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
            ResultDynamicSymbols resultGetSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);
            dynamic symbols = resultGetSymbols.Symbols;
            dynamic main = symbols.Main; // Gets the MAIN Instance of the PLC Program
            // Call a Method that has the following signature (within MAIN Program)
            /*
               {attribute 'TcRpcEnable'}
               METHOD PUBLIC M_Add : INT
               VAR_INPUT
               i1 : INT := 0;
               i2 : INT := 0;
               END_VAR
            */
            short result = main.M_Add(3, 4); // Synchronous Call
            // Call a Method that has no parameter and returns VOID
            main.M_Method1(); // Synchronous call
            //Browsing Rpc Methods
            foreach (IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;
                foreach (IRpcMethodParameter parameter in method.Parameters)
Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]
TwinCAT.Ads.TypeSystem.SymbolLoaderFactory [1523]

### 6.2.5.2.9 AdsConnection.CreateVariableHandle Method

Generates a unique handle for an ADS variable.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```c#
public uint CreateVariableHandle(
    string variableName
)
```

**Parameters**

**variableName**

Type: System.String
Name of the ADS variable

**Return Value**

Type: UInt32
The handle of the ADS Variable.

**Implements**

[AdsHandle.CreateVariableHandle(String) [829]]

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

### 6.2.5.2.10 AdsConnection.CreateVariableHandleAsync Method

Determines the Symbol handle by its instance path asynchronously.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public Task<ResultHandle> CreateVariableHandleAsync(
    string variableName,
    CancellationToken cancel
)
```

**Parameters**

- **variableName**
  - Type: `System.String`
  - Name of the variable.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.

**Return Value**

Type: `Task<ResultHandle>`

A task that represents the asynchronous 'CreateVariableHandle' operation. The `ResultHandle` parameter contains the variable handle (`Handle`) and the `ErrorCode` after execution.

**Implements**

`IAdsHandle.CreateVariableHandleAsync(String, CancellationToken) [830]`

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td></td>
</tr>
</tbody>
</table>

**Remarks**

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this `CreateVariableHandleAsync(String, CancellationToken)` is the `DeleteVariableHandleAsync(UInt32, CancellationToken) [432]`.

**Reference**

- AdsConnection Class [357]
- TwinCAT.Ads Namespace [151]
- AdsConnection.DeleteVariableHandleAsync(UInt32, CancellationToken) [432]
- AdsConnection.TryCreateVariableHandle(String, UInt32.) [499]
- AdsConnection.CreateVariableHandle(String) [429]

**6.2.5.2.11 AdsConnection.DeleteDeviceNotification Method**

Deletes an existing notification.
namespace: TwinCAT.Ads
assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C# public void DeleteDeviceNotification(
    uint notificationHandle
) Parameters

notificationHandle Type: System.UInt32
Handle of the notification.

Implements

IAdsNotifications.DeleteDeviceNotification(UInt32) [857]

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

6.2.5.2.12 AdsConnection.DeleteDeviceNotificationAsync Method

Deletes a registered notification asynchronously.

namespace: TwinCAT.Ads
assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C# public Task<ResultAds> DeleteDeviceNotificationAsync(
    uint notificationHandle,
    CancellationToken cancel
) Parameters

notificationHandle Type: System.UInt32
Notification handle.

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultAds> [989].
A task that represents the asynchronous 'DeleteDeviceNotification' operation. The ErrorCode [992] property contains the ADS error code after execution.
Implements

IAdSnNotifications.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [➤ 858]

Remarks

This is the complementary method to AddDeviceNotificationAsync Overload [➤ 847] overloads and should be
called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

AdsConnection Class [➤ 357]
TwinCAT.Ads Namespace [➤ 151]
AddDeviceNotificationAsync Overload [➤ 847]
IAdSnNotifications.AdsNotification [➤ 866]
TryAddDeviceNotification Overload [➤ 859]
AddDeviceNotification Overload [➤ 844]

6.2.5.2.13 AdsConnection.DeleteVariableHandle Method

Releases the handle of a ADS variable again.

Namespace: TwinCAT.Ads [➤ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.99b9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public void DeleteVariableHandle(
    uint variableHandle
)

Parameters

variableHandle Type: System.UInt32
Handle of the ADS variable

Implements

IAdSnHandle.DeleteVariableHandle(UInt32) [➤ 830]

Reference

AdsConnection Class [➤ 357]
TwinCAT.Ads Namespace [➤ 151]

6.2.5.2.14 AdsConnection.DeleteVariableHandleAsync Method

Releases the handle of a ADS variable again (asynchronously)
Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultAds> DeleteVariableHandleAsync(
    uint variableHandle,
    CancellationToken cancel
)
```

Parameters

- **variableHandle**
  - Type: `System.UInt32`
  - Handle of the ADS variable

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.

Return Value

Type: `Task<ResultAds>`.

A task that represents the asynchronous 'ReadState' operation. The `ResultAds` parameter contains the `ErrorCode` of the ADS communication after execution.

Implements

- `IAdsHandle.DeleteVariableHandleAsync(UInt32, CancellationToken)`

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ObjectDisposedException</code></td>
<td></td>
</tr>
<tr>
<td><code>ClientNotConnectedException</code></td>
<td>[63]</td>
</tr>
</tbody>
</table>

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this `DeleteVariableHandleAsync(UInt32, CancellationToken)` is the `CreateVariableHandleAsync(String, CancellationToken)`.

Reference

- `AdsConnection Class` [357]
- `TwinCAT.Ads Namespace` [151]
- `AdsConnection.CreateVariableHandleAsync(String, CancellationToken)` [429]
- `AdsConnection.TryDeleteVariableHandle(UInt32)` [501]
- `AdsConnection.DeleteVariableHandle(UInt32)` [432]
6.2.5.2.15 AdsConnection.Disconnect Method

Disconnects this IConnection.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd6ca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool Disconnect()
```

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IConnection.Disconnect.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class
TwinCAT.Ads Namespace

6.2.5.2.16 AdsConnection.Dispose Method

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd6ca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void Dispose()
```

Implements

IDisposable.Dispose.

Reference

AdsConnection Class
TwinCAT.Ads Namespace
6.2.5.2.17 AdsConnection.InvokeRpcMethod Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="InvokeRpcMethod(String, String, Object.)" /> ![435]</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td><img src="#" alt="InvokeRpcMethod(String, String, Object, Object..)" /> ![437]</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td><img src="#" alt="InvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object..)" /> ![438]</td>
<td>Invokes the specified RPC Method</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class ![357]
TwinCAT.Ads Namespace ![151]

AdsConnection.InvokeRpcMethod Method (String, String, .Object.)

Invokes the specified RPC Method

**Namespace:** TwinCAT.Ads ![151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters
)
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbolPath</td>
<td>System.String</td>
<td>The symbol path.</td>
</tr>
<tr>
<td>methodName</td>
<td>System.String</td>
<td>The method name.</td>
</tr>
<tr>
<td>inParameters</td>
<td>System.Object</td>
<td>The input parameters or NULL</td>
</tr>
</tbody>
</table>
Return Value

Type: Object
The return value of the Method (as object).

Implements

IAdsRpcInvoke.InvokeRpcMethod(String, String, .Object.) [890]

Remarks

This method only supports primitive data types as inParameters. Any available outparameters will be ignored. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"];  // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /* (attribute 'TcRpcEnable')
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
            i1 : INT := 0;
i2 : INT := 0;
            END_VAR
            */
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] { (short)3, (short)4 });

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] { });

            // Browsing RpcMethods
            foreach(IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;

                foreach(IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}
Reference

AdsConnection Class [357]
InvokeRpcMethod Overload [435]
TwinCAT.Ads Namespace [151]

AdsConnection.InvokeRpcMethod Method (String, String, .Object., .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object[] outParameters
)
```

Parameters

symbolPath Type: System.String
The symbol path.

methodName Type: System.String
The method name.

inParameters Type: .System.Object
The input parameters or NULL

outParameters Type: .System.Object..
The output parameters.

Return Value

Type: Object
The return value of the Method (as object).

Implements

IAdsRpcInvoke.InvokeRpcMethod(String, String, .Object., .Object..) [891]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.
**Dynamic Tree Mode**

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program
            // Call a Method that has the following signature (within MAIN Program)
            /*
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
                i1 : INT := 0;
                i2 : INT := 0;
            END_VAR
            */
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short)3, (short)4});
            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});
            // Browsing RpcMethods
            foreach (IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;
                foreach (IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}

**Reference**

AdsConnection Class [151]

InvokeRpcMethod Overload [435]

TwinCAT.Ads Namespace [151]

**AdsConnection.InvokeRpcMethod Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..)**

Invokes the specified RPC Method

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#  
public Object InvokeRpcMethod(  
    string symbolPath,  
    string methodName,  
    Object[] inParameters,  
    AnyTypeSpecifier[] outSpecifiers,  
    AnyTypeSpecifier retSpecifier,  
    out Object[] outParameters  
)

Parameters

symbolPath  
Type: System.String  
The symbol path.

methodName  
Type: System.String  
The method name.

inParameters  
Type: System.Object  
The parameters.

outSpecifiers  
Type: TwinCAT.TypeSystem.AnyTypeSpecifier[]  
The out specifiers (specifying the out types) or NULL.

retSpecifier  
Type: TwinCAT.TypeSystem.AnyTypeSpecifier[]  
The ret specifier (specifying the return value) or NULL.

outParameters  
Type: System.Object  
The out parameters.

Return Value

Type: Object  
The return value of the Method (as object).

Implements

IAdsRpcInvoke.InvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object)  
[893]

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCall1VirtualProgram  
{  
    /// <summary>  
    /// Defines the entry point of the application.  
    /// </summary>  
    static void Main(string[] args)  
    {  
        // Get the AdsAddress from command-line arguments  
        AdsAddress address = ArgParser.Parse(args);  
        using (AdsClient client = new AdsClient())  
        {  
            // Some code to call the method...  
        }  
    }  
}
```csharp
// client.Synchronize = false;

// Connect to the target device
client.Connect(address);

SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolLoaderSettings.VirtualTree);
ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

// Get the Symbols (Dynamic Symbols)
IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

// Call a Method that has the following signature (within MAIN Program)
METHOD PUBLIC M_Add : INT
VAR_INPUT
  i1 : INT := 0;
  i2 : INT := 0;
END_VAR
*/
short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short)3, (short)4});

// Call a Method that has no parameter and returns VOID
main.InvokeRpcMethod("M_Method1", new object[]{});

// Browsing RpcMethods
foreach(IRpcMethod method in main.RpcMethods)
{
  string methodName = method.Name;
  foreach(IRpcMethodParameter parameter in method.Parameters)
  {
    string parameterName = parameter.Name;
    string parameterType = parameter.TypeName;
  }
}
```

**Reference**

*AdsConnection Class [357]*

*InvokeRpcMethod Overload [435]*

*TwinCAT.Ads Namespace [151]*

### 6.2.5.2.18 AdsConnection.InvokeRpcMethodAsync Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="InvokeRpcMethodAsync" /></td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>InvokeRpcMethodA sync</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td></td>
<td>[443]</td>
</tr>
<tr>
<td>InvokeRpcMethodA sync</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td></td>
<td>[445]</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

**AdsConnection.InvokeRpcMethodAsync Method (String, String, .Object., CancellationToken)**

Invokes the specified RPC Method asynchronously

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Task<ResultRpcMethod> InvokeRpcMethodAsync(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    CancellationToken cancel
)
```

**Parameters**

- **symbolPath**: Type: `System.String`
  The symbol/Instance path of the symbol.

- **methodName**: Type: `System.String`
  The method name.

- **inParameters**: Type: `System.Object`
  The parameters.

- **cancel**: Type: `System.Threading.CancellationToken`
  The cancellation token.
Return Value

Type: Task.ResultRpcMethod[1025].

A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethod[1025] results contain the return value together with the output parameters.

Implements

IAdsRpcInvoke.InvokeRpcMethodAsync(String, String, Object, CancellationToken)[895]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier[1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        static void Main(string[] args)
        {
            // Get the AdsAddress from command-line arguments
            AmsAddress address = ArgParser.Parse(args);

            using (AdsClient client = new AdsClient())
            {
                // client.Synchronize = false;
                // Connect to the target device
                client.Connect(address);

                SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
                ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

                // Get the Symbols (Dynamic Symbols)
                IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

                // Call a Method that has the following signature (within MAIN Program)
                /* {attribute 'TcRpcEnable'}
                METHOD PUBLIC M_Add : INT
                VAR_INPUT
                    i1 : INT := 0;
                    i2 : INT := 0;
                END_VAR
                */

                short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short)3, (short)4});

                // Call a Method that has no parameter and returns VOID
                main.InvokeRpcMethod("M_Method1", new object[]{});

                // Browsing RpcMethods
                foreach (IRpcMethod method in main.RpcMethods)
                {
                    string methodName = method.Name;
                    foreach (IRpcMethodParameter parameter in method.Parameters)
                    {
                        string parameterName = parameter.Name;
                        string parameterType = parameter.TypeName;
                    }
                }
            }
        }
    }
}
Reference

AdsConnection Class [357]
InvokeRpcMethodAsync Overload [440]
TwinCAT.Ads Namespace [151]

AdsConnection.InvokeRpcMethodAsync Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultRpcMethod> InvokeRpcMethodAsync(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    CancellationToken cancel)
```

Parameters

- **symbolPath**
  - Type: System.String
  - The symbol/Instance path of the symbol.

- **methodName**
  - Type: System.String
  - The method name.

- **inParameters**
  - Type: .System.Object
  - The parameters.

- **outSpecifiers**
  - Type: .TwinCAT.TypeSystem.AnyTypeSpecifier
  - The out specifiers (specifying the out types) or NULL.

- **retSpecifier**
  - Type: TwinCAT.TypeSystem.AnyTypeSpecifier
  - The ret specifier (specifying the return value) or NULL.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token

Return Value

Type: Task<ResultRpcMethod> [1025]

A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethod [1025] results contains the return value together with the output parameters.
Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set. ReturnValue [1028] and the ErrorCode [992] of the ADS communication after execution.

Implements

IAdsRpcInvoke.InvokeRpcMethodAsync(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken) [897]

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /*
            * {attribute 'TcRpcEnable'}
            * METHOD PUBLIC M_Add : INT
            * VAR_INPUT
            *     i1 : INT := 0;
            *     i2 : INT := 0;
            * END_VAR
            */

            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short)3, (short)4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});

            // Browsing RpcMethods
            foreach(IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;
                foreach(IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}
Reference

AdsConnection Class [357]
InvokeRpcMethodAsync Overload [440]
TwinCAT.Ads Namespace [151]

AdsConnection.InvokeRpcMethodAsync Method (IRpcCallableInstance, IRpcMethod, Object[], AnyTypeSpecifier[], AnyTypeSpecifier, CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultRpcMethod> InvokeRpcMethodAsync(
    IRpcCallableInstance symbol,
    IRpcMethod method,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    CancellationToken cancel
)
```

Parameters

symbol Type: TwinCAT.TypeSystem.IRpcCallableInstance [2104]
The RPC callable symbol.

method Type: TwinCAT.TypeSystem.IRpcMethod [2123]
The method.

inParameters Type: System.Object
The parameters.

outSpecifiers Type: TwinCAT.TypeSystem.AnyTypeSpecifier [1633]
The out specifiers (specifying the out types) or NULL.

retSpecifier Type: TwinCAT.TypeSystem.AnyTypeSpecifier [1633]
The ret specifier (specifying the return value) or NULL.

cancel Type: System.Threading.CancellationToken
The cancellation token

Return Value

Type: Task.ResultRpcMethod [1025].
A task that represents the asynchronous ‘InvokeRpcMethod’ operation. The ResultRpcMethod [1025] results contains the return value together with the output parameters.

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.ReturnValue [1028] and the ErrorCode [992] of the ADS communication after execution.
TwinCAT.Ads Namespaces

Implements

IAdsRpcInvoke.InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken) [899]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>[63]</td>
</tr>
</tbody>
</table>

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        using (AdsClient client = new AdsClient())
        {
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /* (attribute 'TcRpcEnable')
               METHOD PUBLIC M_Add : INT
               VAR_INPUT
               i1 : INT := 0;
               i2 : INT := 0;
               END_VAR
               */
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short)3, (short)4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});

            // Browsing RpcMethods
            foreach(IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;

                foreach(IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}
Reference

AdsConnection Class [357]
InvokeRpcMethodAsync Overload [440]
TwinCAT.Ads Namespace [151]

6.2.5.2.19 AdsConnection.Read Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read(UInt32, Memory)</td>
<td>[447]</td>
</tr>
<tr>
<td>Read(UInt32, UInt32, Memory)</td>
<td>[448]</td>
</tr>
<tr>
<td>Read(UInt32, UInt32, Memory, Void)</td>
<td>[448]</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

AdsConnection.Read Method (UInt32, Memory`1)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
```csharp
public int Read(
    uint variableHandle,
    Memory buffer)
```

Parameters

- variableHandle Type: System.UInt32
- buffer Type: Memory

Return Value

Type: Int32
Reference

AdsConnection Class [► 357]

Read Overload [► 447]

TwinCAT.Ads Namespace [► 151]

**AdsConnection.Read Method (UInt32, UInt32, Memory`1)**

**Namespace:** TwinCAT.Ads [► 151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int Read(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer
)
```

**Parameters**

- **indexGroup** Type: System.UInt32
- **indexOffset** Type: System.UInt32
- **readBuffer** Type: Memory

**Return Value**

Type: Int32

**Reference**

AdsConnection Class [► 357]

Read Overload [► 447]

TwinCAT.Ads Namespace [► 151]

**AdsConnection.Read Method (UInt32, UInt32, Memory`1, Void)**

**Namespace:** TwinCAT.Ads [► 151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int Read(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void timeout
)
```
Parameters

- `indexGroup`  
  Type: `System.UInt32`

- `indexOffset`  
  Type: `System.UInt32`

- `readBuffer`  
  Type: `Memory`

- `timeout`  
  Type: `System.Void`

Return Value

Type: `Int32`

Reference

- `AdsConnection Class` [357]
- `Read Overload` [447]
- `TwinCAT.Ads Namespace` [151]

6.2.5.2.20 AdsConnection.ReadAny Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReadAny.T.(UInt32)</code> [450]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny.T.(UInt32, Int32.)</code> [450]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny(UInt32, Type)</code> [451]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny.T.(UInt32, UInt32)</code> [452]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny(UInt32, Type, Int32.)</code> [453]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny.T.(UInt32, UInt32, Int32.)</code> [454]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny(UInt32, UInt32, Type)</code> [455]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny(UInt32, UInt32, Type, Int32.)</code> [456]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny(UInt32, UInt32, Type, Int32., Int32)</code> [457]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
</tbody>
</table>
AdsConnection.ReadAny<T>. Method (UInt32)

Reads data synchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch/releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public T ReadAny<T>(
    uint variableHandle
)
```

**Parameters**

<table>
<thead>
<tr>
<th>variableHandle</th>
<th>Type: System.UInt32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handle of the ADS variable.</td>
<td></td>
</tr>
</tbody>
</table>

**Type Parameters**

<table>
<thead>
<tr>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>The type of the value to read.</td>
</tr>
</tbody>
</table>

**Return Value**

Type: T
The value of the read symbol.

**Implements**

IAdsAnyAccess.ReadAny<T>(UInt32) [707]

Reference

AdsConnection Class [357]

ReadAny Overload [449]

TwinCAT.Ads Namespace [151]

AdsConnection.ReadAny<T>. Method (UInt32, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch/releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Reference

AdsConnection Class [357]

ReadAny Overload [449]

TwinCAT.Ads Namespace [151]
Syntax

C#

```csharp
public T ReadAny<T>(
    uint variableHandle,
    int[] args
)
```

Parameters

- **variableHandle**
  - Type: `System.UInt32`
  - Handle of the ADS variable.

- **args**
  - Type: `System.Int32`
  - Additional arguments.

Type Parameters

- **T**
  - The type of the value to read.

Return Value

- Type: `T`
  - The value of the read symbol.

Implements

- `IAdsAnyAccess.ReadAny<T>(UInt32, Int32)` [708]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter `args` specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter `args`.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td><code>args[0]</code>: Number of characters in the string typed as FixedLengthZeroTerminated [2395]</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

- AdsConnection Class [357]
- ReadAny Overload [449]
- TwinCAT.Ads Namespace [151]

**AdsConnection.ReadAny Method (UInt32, Type)**

Reads data synchronously from an ADS device and writes it to an object.

- **Namespace:** TwinCAT.Ads [151]
- **Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public Object ReadAny(
    uint variableHandle,
    Type type
)
```

Parameters

- **variableHandle**
  - Type: System.UInt32
  - Handle of the ADS variable.

- **type**
  - Type: System.Type
  - Type of the object to be read.

Return Value

Type: Object

The read object.

Implements

IAdsAnyAccess.ReadAny(UInt32, Type) [709]

Reference

AdsConnection Class [357]

ReadAny Overload [449]

TwinCAT.Ads Namespace [151]

**AdsConnection.ReadAny.T. Method (UInt32, UInt32)**

Reads data synchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public T ReadAny<T>(
    uint indexGroup,
    uint indexOffset
)
```

Parameters

- **indexGroup**
  - Type: System.UInt32
  - Index group of the ADS variable.

- **indexOffset**
  - Type: System.UInt32
  - Index offset of the ADS variable.

Type Parameters

- **T**
  - The type of the object to be read.
Return Value
Type: T
The read value.

Implements

`IAdsAnyAccess.ReadAny(T(UInt32, UInt32))` [710]

Reference

`AdsConnection Class` [357]
`ReadAny Overload` [449]
`TwinCAT.Ads Namespace` [151]

**AdsConnection.ReadAny Method (UInt32, Type, .Int32.)**

Reads data synchronously from an ADS device and writes it to an object.

**Namespace**: TwinCAT.Ads [151]
**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Object ReadAny(
    uint variableHandle,
    Type type,
    int[] args
)
```

**Parameters**

- `variableHandle` Type: System.UInt32
  Handle of the ADS variable.
- `type` Type: System.Type
  Type of the object to be read.
- `args` Type: .System.Int32.
  Additional arguments.

**Return Value**

Type: Object
The read value.

Implements

`IAdsAnyAccess.ReadAny(UInt32, Type, .Int32.)` [710]
Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated</td>
</tr>
<tr>
<td>string[]</td>
<td>2395.</td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]

ReadAny Overload [449]

TwinCAT.Ads Namespace [151]

**AdsConnection.ReadAny.T. Method (UInt32, UInt32, .Int32.)**

Reads data synchronously from an ADS device and writes it to an object.

**Namespace:**  TwinCAT.Ads [151]

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public T ReadAny<T>(
    uint indexGroup,
    uint indexOffset,
    int[] args
)
```

**Parameters**

- **indexGroup**  
  Type: System.UInt32  
  Index group of the ADS variable.

- **indexOffset**  
  Type: System.UInt32  
  Index offset of the ADS variable.

- **args**  
  Type: .System.Int32.  
  Additional arguments.

**Type Parameters**

- **T**  
  The type of the object to be read.

**Return Value**

- **Type:**  
  T  
  The read value.
TwinCAT.Ads Namespaces

Implements
IAdsAnyAccess.ReadAny.T(UInt32, UInt32, .Int32.) [711]

Remarks
As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395].</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference
AdsConnection Class [357]
ReadAny Overload [449]
TwinCAT.Ads Namespace [151]

AdsConnection.ReadAny Method (UInt32, UInt32, Type)
Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>indexGroup</td>
<td>System.UInt32</td>
<td>Index group of the ADS variable.</td>
</tr>
<tr>
<td>indexOffset</td>
<td>System.UInt32</td>
<td>Index offset of the ADS variable.</td>
</tr>
<tr>
<td>type</td>
<td>System.Type</td>
<td>Type of the object to be read.</td>
</tr>
</tbody>
</table>

Return Value

Type: Object
The read value.
**Implements**

**TwinCAT.Ads Namespace**

**Reference**

**AdsConnection Class**

**ReadAny Overload**

**TwinCAT.Ads Namespace**

**AdsConnection.ReadAny Method (UInt32, UInt32, Type, .Int32.)**

Reads data synchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args
)
```

**Parameters**

- **indexGroup**
  - Type: System.UInt32
  - Index group of the ADS variable.

- **indexOffset**
  - Type: System.UInt32
  - Index offset of the ADS variable.

- **type**
  - Type: System.Type
  - Type of the object to be read.

- **args**
  - Type: System.Int32
  - Additional arguments.

**Return Value**

Type: Object

The read value.

**Implements**

**IAdsAnyAccess.ReadAny(UInt32, UInt32, Type) [712]**

**Remarks**

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.
### TwinCAT.Ads Namespaces

#### Type of value Parameter

<table>
<thead>
<tr>
<th>String</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395]</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

### Reference

- AdsConnection Class [357]
- ReadAny Overload [449]
- TwinCAT.Ads Namespace [151]

### AdsConnection.ReadAny Method (UInt32, UInt32, Type, .Int32., Int32)

Reads data synchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args,
    int timeout
)
```

**Parameters**

- **indexGroup**
  - Type: System.UInt32
  - Index group of the ADS variable.

- **indexOffset**
  - Type: System.UInt32
  - Index offset of the ADS variable.

- **type**
  - Type: System.Type
  - Type of the object to be read.

- **args**
  - Type: System.Int32
  - Additional arguments.

- **timeout**
  - Type: System.Int32
  - The timeout.

**Return Value**

- Type: Object
  - The Value of the data marshalled to the specified type.

**Implements**

- IAdsReadWriteTimeoutAccess.ReadAny(UInt32, UInt32, Type, .Int32., Int32) [880]
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>[63]</td>
</tr>
</tbody>
</table>

## Remarks

If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args. The type is limited to Primitive types ('AnyType').

## Reference

- AdsConnection Class [357]
- ReadAny Overload [449]
- TwinCAT.Ads Namespace [151]

### 6.2.5.2.21 AdsConnection.ReadAnyAsync Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAnyAsync(UInt32, CancellationToken)</td>
<td>Reads data synchronously from an ADS device.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Int32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, Int32, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadAnyAsync&lt;T&gt;(UInt32, UInt32, Int32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, Int32, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
</tbody>
</table>

**Reference**

*AdsConnection Class [357]*

*TwinCAT.Ads Namespace [151]*

**AdsConnection.ReadAnyAsync<T>. Method (UInt32, CancellationToken)**

Reads data synchronously from an ADS device.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Shah.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint variableHandle,
    CancellationToken cancel
)

**Parameters**

- **variableHandle**
  - Type: System.UInt32
  - The variable/symbol handle.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

**Type Parameters**

- **T**
  - The Type of the value to be read.

**Return Value**

Type: Task<ResultValue<T>>. A task that represents the asynchronous read operation. The ResultAnyValue parameter contains the read value (Value) and the ErrorCode after execution.
Implements

IAdsAnyAccess.ReadAnyAsync<T>(UInt32, CancellationToken) [715]

Remarks

As object types only primitive types are supported.

Reference

AdsConnection Class [357]

ReadAnyAsync Overload [458]

TwinCAT.Ads Namespace [151]

**AdsConnection.ReadAnyAsync<T> Method (UInt32, .Int32., CancellationToken)**

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

**C#**

```csharp
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint variableHandle,
    int[] args,
    CancellationToken cancel)
```

Parameters

- **variableHandle**
  Type: System.UInt32
  The variable handle.

- **args**
  Type: System.Int32.
  Additional arguments.

- **cancel**
  Type: System.Threading.CancellationToken
  The cancellation token.

Type Parameters

- **T**
  Type of the object to be read

Return Value

Type: Task<ResultValue<T>>

A task that represents the asynchronous read operation. The ResultValue<T>.Value parameter contains the read value (Value [1032]) and the ErrorCode [992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync<T>(UInt32, .Int32., CancellationToken) [715]
Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395].</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
ReadAnyAsync Overload [458]
TwinCAT.Ads Namespace [151]

AdsConnection.ReadAnyAsync Method (UInt32, Type, CancellationToken)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultAnyValue> ReadAnyAsync(
    uint variableHandle,
    Type type,
    CancellationToken cancel
)
```

Parameters

- `variableHandle`: Type: `System.UInt32`
The variable/symbol handle.
- `type`: Type: `System.Type`
The type of the object to be read.
- `cancel`: Type: `System.Threading.CancellationToken`
The cancellation token.

Return Value

Type: `Task<ResultAnyValue> [998]`
A task that represents the asynchronous read operation. The `ResultAnyValue [998]` parameter contains the read value (Value [1000]) and the `ErrorCode [992]` after execution.

Implements

`IAdsAnyAccess.ReadAnyAsync(UInt32, Type, CancellationToken) [716]`
Remarks
As object types only primitive types are supported.

Reference
AdsConnection Class [357]
ReadAnyAsync Overload [458]
TwinCAT.Ads Namespace [151]

AdsConnection.ReadAnyAsync<T> Method (UInt32, UInt32, CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#

```csharp
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint indexGroup,
    uint indexOffset,
    CancellationToken cancel
)
```

Parameters

- **indexGroup**
  Type: System.UInt32
  Index group of the ADS variable.

- **indexOffset**
  Type: System.UInt32
  Index offset of the ADS variable.

- **cancel**
  Type: System.Threading.CancellationToken
  The cancellation token.

Type Parameters

- **T**

Return Value

Type: Task<ResultValue<T>>[1029].T.
The asynchronous result.

Return Value

Type: Task<ResultValue<T>>[1029].T.
A task that represents the asynchronous read operation. The ResultValue.TValue[1029] parameter contains the read value (Value[1032]) and the ErrorCode[992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync<T>(UInt32, UInt32, CancellationToken)[717]
AdsConnection.ReadAnyAsync Method (UInt32, Type, .Int32., CancellationToken)

Reads data synchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Task<ResultAnyValue> ReadAnyAsync(
    uint variableHandle,
    Type type,
    int[] args,
    CancellationToken cancel)
```

**Parameters**

- `variableHandle`  
  Type: System.UInt32  
  The variable handle.

- `type`  
  Type: System.Type  
  Type of the object to be read.

- `args`  
  Type: System.Int32  
  Additional arguments.

- `cancel`  
  Type: System.Threading.CancellationToken  
  The cancellation token.

**Return Value**

Type: Task<ResultAnyValue> [998].  
A task that represents the asynchronous read operation. The ResultAnyValue [998] parameter contains the read value (Value [1000]) and the ErrorCode [992] after execution.

**Implements**

IAdsAnyAccess.ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [718]

**Remarks**

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395].</td>
</tr>
</tbody>
</table>
### AdsConnection.ReadAnyAsync.T. Method (UInt32, UInt32, Int32, CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint indexGroup,
    uint indexOffset,
    int[] args,
    CancellationToken cancel)
```

#### Parameters

- **indexGroup**
  - Type: `System.UInt32`
  - Index group of the ADS variable.

- **indexOffset**
  - Type: `System.UInt32`
  - Index offset of the ADS variable.

- **args**
  - Type: `System.Int32`
  - Additional arguments.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.

#### Type Parameters

- **T**
  - The type of the result value.

#### Return Value

- Type: `Task<ResultValue<T>>`
  - A task that represents the asynchronous read operation. The `ResultValue.TValue` parameter contains the read value (`Value`) and the `ErrorCode` after execution.

#### Implements

- `IAdsAnyAccess.ReadAnyAsync<T>(UInt32, UInt32, Int32, CancellationToken)`
Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395].</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]

ReadAnyAsync Overload [458]

TwinCAT.Ads Namespace [151]

AdsConnection.ReadAnyAsync Method (UInt32, UInt32, Type, CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  
public Task<ResultAnyValue> ReadAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Type type,
    CancellationToken cancel
)

Parameters

indexGroup Type: System.UInt32
Index group of the ADS variable.

indexOffset Type: System.UInt32
Index offset of the ADS variable.

type Type: System.Type
Type of the object to be read.

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultAnyValue> [998].
A task that represents the asynchronous read operation. The ResultAnyValue [998] parameter contains the read value (Value [1000]) and the ErrorCode [992] after execution.
TwinCAT.Ads Namespaces

Implements

IAdsAnyAccess.ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [720]

Reference

AdsConnection Class [357]

ReadAnyAsync Overload [458]

TwinCAT.Ads Namespace [151]

AdsConnection.ReadAnyAsync Method (UInt32, UInt32, Type, .Int32., CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd2ca3e72bc0ea15da1c14

Syntax

C#

public Task<ResultAnyValue> ReadAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args,
    CancellationToken cancel
)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>indexGroup</td>
<td>System.UInt32</td>
<td>Index group of the ADS variable.</td>
</tr>
<tr>
<td>indexOffset</td>
<td>System.UInt32</td>
<td>Index offset of the ADS variable.</td>
</tr>
<tr>
<td>type</td>
<td>System.Type</td>
<td>Type of the object to be read.</td>
</tr>
<tr>
<td>cancel</td>
<td>System.Threading.CancellationToken</td>
<td>The cancellation token.</td>
</tr>
</tbody>
</table>

Return Value

Type: Task<ResultAnyValue> [998].
A task that represents the asynchronous read operation. The ResultAnyValue [998] parameter contains the read value (Value [1000]) and the ErrorCode [992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [721]
Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated[\p.2395].</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [\p.357]
ReadAnyAsync Overload [\p.458]
TwinCAT.Ads Namespace [\p.151]

6.2.5.2.22 AdsConnection.ReadAnyString Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAnyString(UInt32, Int32, Encoding)</td>
<td>Reads a string from the specified symbol/variable.</td>
</tr>
<tr>
<td>ReadAnyString(UInt32, UInt32, Int32, Encoding)</td>
<td>Reads as string from a specified address.</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [\p.357]
TwinCAT.Ads Namespace [\p.151]

AdsConnection.ReadAnyString Method (UInt32, Int32, Encoding)

Reads a string from the specified symbol/variable.

Namespace: TwinCAT.Ads [\p.151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public string ReadAnyString(
    uint variableHandle,
    int len,
    Encoding encoding
)
```
Parameters

variableHandle Type: System.UInt32
The variable handle.

len Type: System.Int32
The length.

encoding Type: System.Text.Encoding
The encoding.

Return Value

Type: String
The string value.

Implements

IAdsAnyAccess.ReadAnyString(UInt32, Int32, Encoding) [722]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>[63]</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]

ReadAnyString Overload [467]

TwinCAT.Ads Namespace [151]

**AdsConnection.ReadAnyString Method (UInt32, UInt32, Int32, Encoding)**

Reads as string from a specified address.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public string ReadAnyString(
    uint indexGroup,
    uint indexOffset,
    int len,
    Encoding encoding
)
```

Parameters

indexGroup Type: System.UInt32
The index group.
indexOffset  Type: System.UInt32
The index offset.

len  Type: System.Int32
The string length to be read.

encoding  Type: System.Text.Encoding
The encoding.

Return Value
Type: String
System.String.

Implements
IAdsAnyAccess.ReadAnyString(UInt32, UInt32, Int32, Encoding) [723]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>[63]</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]

ReadAnyString Overload [467]

TwinCAT.Ads Namespace [151]

6.2.5.2.23 AdsConnection.ReadAnyStringAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAnyStringAsyncc(UInt32, Int32, Encoding, CancellationToken) [470]</td>
<td>Reads a string asynchronously from the specified symbol/variable</td>
</tr>
<tr>
<td>ReadAnyStringAsyncc(UInt32, UInt32, Int32, Encoding, CancellationToken) [471]</td>
<td>read any string as an asynchronous operation.</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
AdsConnection.ReadAnyStringAsync Method (UInt32, Int32, Encoding, CancellationToken)

Reads a string asynchronously from the specified symbol/variable

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultAnyValue> ReadAnyStringAsync(
    uint variableHandle,
    int len,
    Encoding encoding,
    CancellationToken cancel
)
```

Parameters

- `variableHandle`: Type: System.UInt32
  The variable handle.
- `len`: Type: System.Int32
  The length.
- `encoding`: Type: System.Text.Encoding
  The encoding.
- `cancel`: Type: System.Threading.CancellationToken
  The cancellation token.

Return Value

Type: Task<ResultAnyValue>
A task that represents the asynchronous read operation. The ResultAnyValue parameter contains the read string (Value) and the ErrorCode after execution.

Implements

- AdsAnyAccess.ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken)

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

- AdsConnection Class
- ReadAnyStringAsync Overload

TwinCAT.Ads Namespace
TwinCAT.Ads Namespace [151]

AdsConnection.ReadAnyStringAsync Method (UInt32, UInt32, Int32, Encoding, CancellationToken)

read any string as an asynchronous operation.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdcd3a3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultAnyValue> ReadAnyStringAsync(
    uint indexGroup,
    uint indexOffset,
    int len,
    Encoding encoding,
    CancellationToken cancel
)
```

Parameters

- **indexGroup**
  - Type: System.UInt32
  - The index group.
- **indexOffset**
  - Type: System.UInt32
  - The index offset.
- **len**
  - Type: System.Int32
  - The string length to be read.
- **encoding**
  - Type: System.Text.Encoding
  - The encoding.
- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

Return Value

Type: Task<ResultAnyValue> [998].
A task that represents the asynchronous read operation. The ResultAnyValue [998] parameter contains the read value (Value) and the ErrorCode [992] after execution.

Implements

IAdsAnyAccess.ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [725]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException [63]</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
6.2.5.2.24 AdsConnection.ReadAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adsConnection.ReadAsync(UInt32, Memory, Void)</td>
<td></td>
</tr>
<tr>
<td>adsConnection.ReadAsync(UInt32, UInt32, Memory, Void)</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

AdsConnection.ReadAsync Method (UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultRead> ReadAsync(
    uint variableHandle,
    Memory readBuffer,
    void cancel
)
```

Parameters

<table>
<thead>
<tr>
<th>variableHandle</th>
<th>Type: System.UInt32</th>
</tr>
</thead>
<tbody>
<tr>
<td>readBuffer</td>
<td>Type: Memory</td>
</tr>
<tr>
<td>cancel</td>
<td>Type: System.Void</td>
</tr>
</tbody>
</table>

Return Value

Type: Task.ResultRead [1008].

Reference

AdsConnection Class [357]
ReadAsync Overload [472]
TwinCAT.Ads Namespace [►151]

AdsConnection.ReadAsync Method (UInt32, UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [►151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
```csharp
public Task<ResultRead> ReadAsync(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void cancel
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>indexGroup</td>
<td>System.UInt32</td>
</tr>
<tr>
<td>indexOffset</td>
<td>System.UInt32</td>
</tr>
<tr>
<td>readBuffer</td>
<td>Memory</td>
</tr>
<tr>
<td>cancel</td>
<td>System.Void</td>
</tr>
</tbody>
</table>

Return Value

Type: Task<ResultRead> [►1008].

Reference

AdsConnection Class [►357]

ReadAsync Overload [►472]

TwinCAT.Ads Namespace [►151]

6.2.5.2.25 AdsConnection.ReadDataType Method

Call this method to obtain information about the specified data type.

Namespace: TwinCAT.Ads [►151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
```csharp
public IDataType ReadDataType(
    string typeName
)
```
Parameters

typeName
  Type: System.String
  Name of the data type (without namespace)

Return Value

Type: IDataType [1986]
An containing the requested type.

Implements

IAdsSymbolicAccess.ReadDataType(String) [942]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>typeName</td>
</tr>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException [63]</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]
IAdsSymbolicAccess.TryReadDataType(String, IDataType) [952]
IAdsSymbolicAccess.ReadDataTypeAsync(String, CancellationToken) [943]

6.2.5.2.26 AdsConnection.ReadDataTypeAsync Method

read data type as an asynchronous operation.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultValue<IDataType>> ReadDataTypeAsync(
    string typeName,
    CancellationToken cancel
)
```

Parameters

typeName
  Type: System.String
  Name of the data type.

cancel
  Type: System.Threading.CancellationToken
  The cancel token.
Return Value

Type: Task.ResultValue [1029]. IDataType [1986].
A task that represents the asynchronous 'ReadDataType' operation. The ResultValue.TValue [1029] parameter contains the read value (Value [1032]) and the ErrorCode [992] after execution.

Implements

IAdsSymbolicAccess.ReadDataTypeAsync(String, CancellationToken) [943]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td>typeName</td>
</tr>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]
IAdsSymbolicAccess.ReadDataType(String) [942]
IAdsSymbolicAccess.TryReadDataType(String, IDataType.) [952]

6.2.5.2.27 AdsConnection.ReadDeviceInfo Method

Reads the identification and version number of an ADS server.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public DeviceInfo ReadDeviceInfo()
```

Return Value

Type: DeviceInfo [698]
DeviceInfo struct containing the name of the device and the version information.

Implements

IAdsConnection.ReadDeviceInfo. [795]
### 6.2.5.2.28 AdsConnection.ReadDeviceInfoAsync Method

Reads the identification and version number of an ADS server.

**Namespace:** TwinCAT.Ads
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public Task<ResultDeviceInfo> ReadDeviceInfoAsync(
    CancellationToken cancel
)
```

#### Parameters

- `cancel`  
  **Type:** System.Threading.CancellationToken  
  The cancellation token.

#### Return Value

**Type:** Task<ResultDeviceInfo>

A task that represents the asynchronous 'ReadDeviceState' operation. The ResultDeviceInfo parameter contains the valueDeviceInfo and the ErrorCode of the ADS communication after execution.

**Implements**

|Task, ReadDeviceInfoAsync(CancellationToken)| 796 |

#### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>[63]</td>
</tr>
</tbody>
</table>

#### Reference

**AdsConnection Class [357]**

**TwinCAT.Ads Namespace [151]**
6.2.5.2.29 AdsConnection.ReadState Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadState</td>
<td>Reads the ADS status and the device status from an ADS server.</td>
</tr>
<tr>
<td>ReadState(Int32)</td>
<td>Reads the ADS status and the device status from an ADS server.</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

AdsConnection.ReadState Method

Reads the ADS status and the device status from an ADS server.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public StateInfo ReadState()

Return Value

Type: StateInfo [1041]
The ADS statue and device status.

Implements

IAdsStateProvider.ReadState. [931]

Reference

AdsConnection Class [357]
ReadState Overload [477]
TwinCAT.Ads Namespace [151]

AdsConnection.ReadState Method (Int32)

Reads the ADS status and the device status from an ADS server.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

public StateInfo ReadState(
    int timeout
)

Parameters

timeout Type: System.Int32
The timeout.

Return Value

Type: StateInfo [1041]
The ADS statue and device status.

Implements

IAdsStateControlTimeout.ReadState(Int32) [923]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]

ReadState Overload [477]

TwinCAT.Ads Namespace [151]

6.2.5.2.30 AdsConnection.ReadStateAsync Method

Read the ADS State asynchronously

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public Task<ResultReadDeviceState> ReadStateAsync(
    CancellationToken cancel
)

Parameters

cancel Type: System.Threading.CancellationToken
The cancellation token.
Return Value

Type: Task.ResultReadDeviceState | 1016.
A task that represents the asynchronous 'ReadState' operation. The ResultReadDeviceState | 1016 parameter contains the value State | 1018 and the ErrorCode | 992 of the ADS communication after execution.

Implements

IAdsStateProvider.ReadStateAsync(CancellationToken) | 931

Reference

AdsConnection Class | 357
TwinCAT.Ads Namespace | 151

6.2.5.2.31 AdsConnection.ReadSymbol Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

Namespace: TwinCAT.Ads | 151
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public IAdsSymbol ReadSymbol(
    string name
)

Parameters

name Type: System.String
Name of the symbol.

Return Value

Type: IAdsSymbol | 1379
A IAdsSymbol | 1379 containing the requested symbol information or null if symbol could not be found.

Implements

IAdsSymbolicAccess.ReadSymbol(String) | 943

Reference

AdsConnection Class | 357
TwinCAT.Ads Namespace | 151

6.2.5.2.32 AdsConnection.ReadSymbolAsync Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.
TwinCAT.Ads Namespaces

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+

Syntax

C#

```csharp
public Task<ResultValue<IAdsSymbol>> ReadSymbolAsync(
    string name,
    CancellationToken cancel
)
```

Parameters

- **name**
  - Type: System.String
  - Name of the symbol.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancel token.

Return Value

Type: Task<ResultValue<IAdsSymbol>>

A task that represents the asynchronous 'ReadSymbolInfo' operation. The ResultValue.TValue parameter contains the read value (Value) and the ErrorCode after execution.

Implements

IAdsSymbolicAccess.ReadSymbolAsync(String, CancellationToken)

Reference

AdsConnection Class
TwinCAT.Ads Namespace

6.2.5.2.33 AdsConnection.ReadValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadValue.T(String)</td>
<td>Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>ReadValue(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes(UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.</td>
</tr>
<tr>
<td>ReadValue.T(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>ReadValue(String, Type)</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class
TwinCAT.Ads Namespace
**AdsConnection.ReadValue.T. Method (String)**

Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public T ReadValue<T>(
    string name
)
```

**Parameters**

- **name**
  - Type: System.String
  - Name of the ADS symbol.

**Type Parameters**

- **T**
  - The value type

**Return Value**

- Type: T
  - Value of the symbol

**Implements**

IAdsSymbolicAccess.ReadValue.T.(String)

**Reference**

- AdsConnection Class
- ReadValue Overload
- TwinCAT.Ads Namespace

---

**AdsConnection.ReadValue Method (ISymbol)**

Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes(UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Object ReadValue(
    ISymbol symbol
)
```
Parameters

symbol
Type: TwinCAT.TypeSystem.ISymbol
The symbol that should be read.

Return Value

Type: Object
The value of the symbol as an object.

Implements

IAdsSymbolicAccess.ReadValue(ISymbol)

Reference

AdsConnection Class
ReadValue Overload
TwinCAT.Ads Namespace

AdsConnection.ReadValue.T Method (ISymbol)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public T ReadValue<T>(
    ISymbol symbol
)
```

Parameters

symbol
Type: TwinCAT.TypeSystem.ISymbol
The symbol that should be read.

Type Parameters

T The value type.

Return Value

Type: T
The value of the symbol.

Implements

IAdsSymbolicAccess.ReadValue.T.(ISymbol)
Remarks
Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference
AdsConnection Class [357]
ReadValue Overload [480]
TwinCAT.Ads Namespace [151]

AdsConnection.ReadValue Method (String, Type)
Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

```csharp
public Object ReadValue(
    string name,
    Type type
)
```

Parameters

name Type: System.String
Name of the ADS symbol.

type Type: System.Type
Managed type of the ADS symbol.

Return Value

Type: Object
Value of the symbol

Implements

[AdsSymbolicAccess.ReadValue(String, Type)] [947]

Reference

AdsConnection Class [357]
ReadValue Overload [480]
TwinCAT.Ads Namespace [151]
6.2.5.2.34 AdsConnection.ReadValueAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadValueAsync.T. (String, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>ReadValueAsync(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.</td>
</tr>
<tr>
<td>ReadValueAsync.T. (ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously and returns it as an object.</td>
</tr>
<tr>
<td>ReadValueAsync(String, Type, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [➤ 357]

TwinCAT.Ads Namespace [➤ 151]

AdsConnection.ReadValueAsync.T. Method (String, CancellationToken)

Reads the value of a symbol asynchronously.

Namespace: TwinCAT.Ads [➤ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultValue<T>> ReadValueAsync<T>(
    string name,
    CancellationToken cancel)
```

Parameters

| name | Type: System.String |
| cancel | Type: System.Threading.CancellationToken |

Type Parameters

T The value type.
TwinCAT.Ads Namespaces

Return Value

Type: Task<ResultValue [1029].
A task that represents the asynchronous read operation. The ResultAnyValue [998] parameter contains the read value (Value [1032]) and the ErrorCode [992] after execution.

Implements

| AdsSymbolicAccess.ReadValueAsync<T>(String, CancellationToken) [948] |

Remarks

The parameter type must have the same layout as the ADS symbol.

Reference

AdsConnection Class [357]
ReadValueAsync Overload [484]
TwinCAT.Ads Namespace [151]

AdsConnection.ReadValueAsync Method (ISymbol, CancellationToken)

Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public Task<ResultAnyValue> ReadValueAsync(
    ISymbol symbol,
    CancellationToken cancel
)

Parameters

| symbol | Type: TwinCAT.TypeSystem.ISymbol [2176] |
|        | The symbol that should be read. |
| cancel | Type: System.Threading.CancellationToken |
|        | The cancel token. |

Return Value

Type: Task<ResultAnyValue [998].
The value of the symbol as an object.

Implements

| AdsSymbolicAccess.ReadValueAsync<T(ISymbol, CancellationToken) [949] |
 Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>63</td>
</tr>
</tbody>
</table>

 Reference

 AdsConnection Class [357]
 ReadValueAsync Overload [484]
 TwinCAT.Ads Namespace [151]

 AdsConnection.ReadValueAsync.T. Method (ISymbol, CancellationToken)

 Reads the value of a symbol asynchronously and returns it as an object.

 Namespace: TwinCAT.Ads [151]
 Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

 Syntax

 C#

 ```csharp
 public Task<ResultValue<T>> ReadValueAsync<T>(
     ISymbol symbol,
     CancellationToken cancel
 )
 ```

 Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbol</td>
<td>TwinCAT.TypeSystem.ISymbol</td>
<td>The symbol that should be read.</td>
</tr>
<tr>
<td>cancel</td>
<td>System.Threading.CancellationToken</td>
<td>The cancel token.</td>
</tr>
</tbody>
</table>

 Type Parameters

 T The value type.

 Return Value

 Type: Task<ResultValue<T>> The task that represents the asynchronous read operation. The ResultAnyValue parameter contains the read value (Value) and the ErrorCode after execution.

 Implements

 IAdsSymbolicAccess.ReadValueAsync.T.ISymbol, CancellationToken) [950]

 Remarks

 Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.
AdsConnection.ReadValueAsync Method (String, Type, CancellationToken)

Reads the value of a symbol asynchronously.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultAnyValue> ReadValueAsync(
    string name,
    Type type,
    CancellationToken cancel
)
```

Parameters

- **name**: Type: System.String
  Name of the ADS symbol.
- **type**: Type: System.Type
  Managed type of the ADS symbol.
- **cancel**: Type: System.Threading.CancellationToken
  The cancel token.

Return Value

Type: Task<ResultAnyValue>
A task that represents the asynchronous read operation. The ResultAnyValue parameter contains the read value (Value) and the ErrorCode after execution.

Implements

IAdsSymbolicAccess.ReadValueAsync(String, Type, CancellationToken)

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

The parameter type must have the same layout as the ADS symbol.
6.2.5.2.35  AdsConnection.ReadWrite Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsConnection.ReadWrite(UInt32, Memory, Void)</td>
<td>[488]</td>
</tr>
<tr>
<td>AdsConnection.ReadWrite(UInt32, UInt32, Memory, Void)</td>
<td>[489]</td>
</tr>
<tr>
<td>AdsConnection.ReadWrite(UInt32, UInt32, Memory, Void, Byte)</td>
<td>[489]</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

AdsConnection.ReadWrite Method (UInt32, Memory`1, Void)

Namespace:  TwinCAT.Ads [151]
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public int ReadWrite(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer
)

Parameters

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>variableHandle</td>
<td>System.UInt32</td>
</tr>
<tr>
<td>readBuffer</td>
<td>Memory</td>
</tr>
<tr>
<td>writeBuffer</td>
<td>System.Void</td>
</tr>
</tbody>
</table>

Return Value

Type:  Int32
AdsConnection.ReadWrite Method (UInt32, UInt32, Memory`1, Void)

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int ReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer
)
```

**Parameters**

- `indexGroup` Type: System.UInt32
- `indexOffset` Type: System.UInt32
- `readBuffer` Type: Memory
- `writeBuffer` Type: System.Void

**Return Value**

Type: Int32

**Reference**

AdsConnection Class [357]

ReadWrite Overload [488]

TwinCAT.Ads Namespace [151]

AdsConnection.ReadWrite Method (UInt32, UInt32, Memory`1, Void, Byte)

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int ReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte value
)
```

**Parameters**

- `indexGroup` Type: System.UInt32
- `indexOffset` Type: System.UInt32
- `readBuffer` Type: Memory
- `writeBuffer` Type: System.Void
- `value` Type: System.Byte

**Return Value**

Type: Int32

**Reference**

AdsConnection Class [357]

ReadWrite Overload [488]

TwinCAT.Ads Namespace [151]
Syntax

C#

```csharp
public int ReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte timeout)
```

Parameters

- **indexGroup** Type: `System.UInt32`
- **indexOffset** Type: `System.UInt32`
- **readBuffer** Type: `Memory`
- **writeBuffer** Type: `System.Void`
- **timeout** Type: `System.Byte`

Return Value

Type: `Int32`

Reference

- AdsConnection Class [☞ 357]
- ReadWrite Overload [☞ 488]
- TwinCAT.Ads Namespace [☞ 151]

6.2.5.2.36 AdsConnection.ReadWriteAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="giorno.png" alt="ReadWriteAsync" /></td>
<td>ReadWriteAsync(UInt32, Memory, Void, Byte) [☞ 491]</td>
</tr>
<tr>
<td><img src="giorno.png" alt="ReadWriteAsync" /></td>
<td>ReadWriteAsync(UInt32, Uint32, Memory, Void, Byte) [☞ 491]</td>
</tr>
</tbody>
</table>

Reference

- AdsConnection Class [☞ 357]
- TwinCAT.Ads Namespace [☞ 151]
AdssConnection.ReadWriteAsync Method (UInt32, Memory`1, Void, Byte)

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public Task<ResultReadWrite> ReadWriteAsync(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer,
    byte cancel
)
```

### Parameters

- **variableHandle**
  - Type: System.UInt32

- **readBuffer**
  - Type: Memory

- **writeBuffer**
  - Type: System.Void

- **cancel**
  - Type: System.Byte

### Return Value

Type: Task<ResultReadWrite>.

### Reference

- AdsConnection Class
- ReadWriteAsync Overload
- TwinCAT.Ads Namespace

AdssConnection.ReadWriteAsync Method (UInt32, UInt32, Memory`1, Void, Byte)

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public Task<ResultReadWrite> ReadWriteAsync(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte cancel
)
```
6.2.5.2.37 AdsConnection.RegisterAdsStateChangedAsync Method

Registers for AdsStateChanged [➤ 567] events as an asynchronous operation.

**Namespace:** TwinCAT.Ads [➤ 151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Task<ResultAds> RegisterAdsStateChangedAsync(
    EventHandler<AdsStateChangedEventArgs> handler,
    CancellationToken cancel)
```

**Parameters**

- **handler**
  Type: System.EventHandler<AdsStateChangedEventArgs> [➤ 627].
  The handler function to be registered for AdsStateChanged calls.

- **cancel**
  Type: System.Threading.CancellationToken
  The cancellation token.

**Return Value**

Type: Task<ResultAds> [➤ 989].
A task that represents the asynchronous 'RegisterAdsStateChanged' operation. The ResultAds [➤ 989] parameter contains the state the ErrorCode [➤ 992] of the ADS communication after execution.
6.2.5.2.38   AdsConnection.RegisterSymbolVersionChangedAsync Method

Registers the symbol version changed asynchronously.

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public Task<ResultAds> RegisterSymbolVersionChangedAsync(
    EventHandler<AdsSymbolVersionChangedEventArgs> handler,
    CancellationToken cancel)

Parameters

handler
Type: System.EventHandler<AdsSymbolVersionChangedEventArgs>
The handler function to register.

cancel
Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultAds>
A task that represents the asynchronous 'RegisterSymbolVersionChanged' operation. The ResultAds parameter contains the value ErrorCode of the ADS communication after execution.

Implements

IAdsSymbolChangedProvider.RegisterSymbolVersionChangedAsync(EventHandler<AdsSymbolVersionChangedEventArgs, CancellationToken>) [935]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]
### AdsConnection.TryAddDeviceNotification Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="494" alt="TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.)." /></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td><img src="495" alt="TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)." /></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
</tbody>
</table>

#### Reference

- AdsConnection Class [357]
- TwinCAT.Ads Namespace [151]

### AdsConnection.TryAddDeviceNotification Method (String, Int32, NotificationSettings, Object, UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public AdsErrorCode TryAddDeviceNotification(
    string variableName,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    out uint handle
)
```

#### Parameters

- **variableName**
  - Type: `System.String`
  - Name of the variable.

- **dataSize**
  - Type: `System.Int32`
  - Maximum amount of data in bytes to receive with this ADS Notification.

- **settings**
  - Type: `TwinCAT.Ads.NotificationSettings` [979]
  - The notification settings.

- **userData**
  - Type: `System.Object`
  - This object can be used to store user specific data.
handle  

Type: System.UInt32
The notification handle.

Return Value

Type: AdsErrorCode [575]
The ADS ErrorCode.

Implements

IAdsNotifications.TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [859]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

The 

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [564] as value. Because notifications allocate TwinCAT system resources, a complementary call to TryDeleteDeviceNotification(UInt32) [500] should always be called when the notification is not used anymore.

Reference

AdsConnection Class [357]

TryAddDeviceNotification Overload [494]

TwinCAT.Ads Namespace [151]

AdsConnection.AdsNotification [564]

AdsConnection.TryDeleteDeviceNotification(UInt32) [500]

AddDeviceNotification Overload [844]

AddDeviceNotificationAsync Overload [847]

TryAddDeviceNotification Overload [859]

AdsConnection.TryAddDeviceNotification Method (UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [564] event.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
public AdsErrorCode TryAddDeviceNotification(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    out uint handle
)
```

Parameters

- **indexGroup**
  - Type: System.UInt32
  - The index group number of the requested ADS service.

- **indexOffset**
  - Type: System.UInt32
  - The index offset number of the requested ADS service.

- **dataSize**
  - Type: System.Int32
  - Maximum amount of data in bytes to receive with this ADS Notification.

- **settings**
  - Type: TwinCAT.Ads.NotificationSettings
  - The Notification settings.

- **userData**
  - Type: System.Object
  - This object can be used to store user specific data (tag data)

- **handle**
  - Type: System.UInt32
  - The notification handle.

Return Value

Type: AdsErrorCode
- The ADS error code.

Implements

IAdsNotifications.TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32)

Remarks

The `dataSize` Parameter defines the amount of bytes, that will be attached to the AdsNotification as value.
Because notifications allocate TwinCAT system resources, a complementary call to TryDeleteDeviceNotification(UInt32) should always called when the notification is not used anymore.

Reference

- AdsConnection Class
- TryAddDeviceNotification Overload
- TwinCAT.Ads Namespace
- AdsConnection.TryDeleteDeviceNotification(UInt32)
- AdsConnection.AdsNotification
- AdsConnection.AdsNotificationError
AddDeviceNotification Overload [844]

TryAddDeviceNotification Overload [859]

[Overload: TwinCAT.Ads.IAdsNotifications..AddDeviceNotificationAsync] Overload

### 6.2.5.2.40 AdsConnection.TryAddDeviceNotificationEx Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryAddDeviceNotificationEx(String,</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by</td>
</tr>
<tr>
<td>NotificationSettings, Object, Type, Int32., UnInt32.) [497]</td>
<td>the AdsNotification event.</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(UInt32,</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by</td>
</tr>
<tr>
<td>UInt32, NotificationSettings, Object, Type, Int32., UnInt32.) [498]</td>
<td>the AdsNotificationEx [566] event.</td>
</tr>
</tbody>
</table>

**Reference**

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

**AdsConnection.TryAddDeviceNotificationEx Method (String, NotificationSettings, Object, Type, .Int32., UInt32.)**

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AdsErrorCode TryAddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    out uint handle
)
```

**Parameters**

**symbolPath**

Type: System.String

The symbol path..
settings

Type: TwinCAT.Ads.NotificationSettings
The settings.

userData

Type: System.Object
This object can be used to store user specific data.

type

Type: System.Type
Type of the object stored in the event argument.

args

Type: System.Int32
Additional arguments.

handle

Type: System.UInt32
The handle.

Return Value

Type: AdsErrorCode
The handle of the notification.

Implements

IAdsNotifications.TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32., UInt32.)

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class

TryAddDeviceNotificationEx Overload

TwinCAT.Ads Namespace

AdsConnection.TryAddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C# public AdsErrorCode TryAddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    out uint handle
)
**Parameters**

- **indexGroup**
  - Type: `System.UInt32`
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - Type: `System.UInt32`
  - Contains the index offset number of the requested ADS service.

- **settings**
  - Type: `TwinCAT.Ads.NotificationSettings`
  - The Notification settings.

- **userData**
  - Type: `System.Object`
  - This object can be used to store user specific data (tag data)

- **type**
  - Type: `System.Type`
  - Type of the object stored in the event argument ('AnyType')

- **args**
  - Type: `System.Int32`
  - The 'AnyType' arguments.

- **handle**
  - Type: `System.UInt32`
  - The notification handle.

**Return Value**

- Type: `AdsErrorCode`
  - The ADS Error code.

**Implements**

- `IAdsNotifications.TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.)`

**Remarks**

If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive types (AnyType) are supported by this method. Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotification(UInt32)` should always be called when the notification is not used anymore.

**Reference**

- `AdsConnection Class`
- `TryAddDeviceNotificationEx Overload`
- `TwinCAT.Ads Namespace`
- `AdsConnection.DeleteDeviceNotification(UInt32)`
- `AdsConnection.AdsNotificationEx`
- `AdsConnection.AdsNotificationError`
- `AddDeviceNotificationEx Overload`
- `TryAddDeviceNotificationEx Overload`
- `AddDeviceNotificationExAsync Overload`

### 6.2.5.2.41 AdsConnection.TryCreateVariableHandle Method

Read (determine) the Symbol handle by its name/path
### TwinCAT.Ads Namespaces

**Namespace:** [TwinCAT.Ads](#151)  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public AdsErrorCode TryCreateVariableHandle(
    string symbolName,
    out uint variableHandle
)
```

#### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type: System.String</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbolName</td>
<td>SymbolName / Path.</td>
</tr>
<tr>
<td>variableHandle</td>
<td>System.UInt32.</td>
</tr>
<tr>
<td></td>
<td>The handle.</td>
</tr>
</tbody>
</table>

#### Return Value

Type: [AdsErrorCode](#575)  
AdsErrorCode.

#### Implements

[IAdsHandle.TryCreateVariableHandle(String, UInt32.)](#834)

#### Reference

[AdsConnection Class](#357)  
[TwinCAT.Ads Namespace](#151)  

### 6.2.5.2.42 AdsConnection.TryDeleteDeviceNotification Method

Deletes a registered notification.

**Namespace:** [TwinCAT.Ads](#151)  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public AdsErrorCode TryDeleteDeviceNotification(
    uint notificationHandle
)
```

#### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type: System.UInt32</th>
</tr>
</thead>
<tbody>
<tr>
<td>notificationHandle</td>
<td>Notification handle.</td>
</tr>
</tbody>
</table>
Return Value

Type: AdsErrorCode [575]
The ADS error code.

Implements

IAdsNotifications.TryDeleteDeviceNotification(UInt32) [864]

Remarks

This is the complementary method to TryAddDeviceNotification Overload [859] overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]
AddDeviceNotification Overload [844]
AdsConnection.AdsNotification [564]
TryAddDeviceNotification Overload [859]
AddDeviceNotificationAsync Overload [847]

6.2.5.2.43 AdsConnection.TryDeleteVariableHandle Method

Releases the specified symbol/variable handle synchronously.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryDeleteVariableHandle(
    uint variableHandle
);
```

Parameters

variableHandle Type: System.UInt32
Handle of the ADS variable

Return Value

Type: AdsErrorCode [575]
The ADS error code.

Implements

IAdsHandle.TryDeleteVariableHandle(UInt32) [835]
Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>[63]</td>
</tr>
</tbody>
</table>

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this TryDeleteVariableHandle(UInt32) is the TryCreateVariableHandle(String, UInt32) [499]

Reference

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

AdsConnection.TryCreateVariableHandle(String, UInt32) [499]

AdsConnection.DeleteVariableHandleAsync(UInt32, CancellationToken) [432]

AdsConnection.DeleteVariableHandle(UInt32) [432]

6.2.5.2.44 AdsConnection.TryInvokeRpcMethod Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object)</td>
<td>Invokes the rpc method.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object., AnyTypeSpecifier., AnyTypeSpecifier., Object., Object.)</td>
<td>Invokes the rpc method.</td>
</tr>
</tbody>
</table>

**Reference**

AdsConnection Class [► 357]

TwinCAT.Ads Namespace [► 151]

**AdsConnection.TryInvokeRpcMethod Method (String, String, Object., Object.)**

Invokes the specified RPC Method

**Namespace:** TwinCAT.Ads [► 151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object retValue
)
```

**Parameters**

- **symbolPath**
  Type: `System.String`
  The symbol path.

- **methodName**
  Type: `System.String`
  The method name.

- **inParameters**
  Type: `System.Object`
  The parameters.

- **retValue**
  Type: `System.Object`
  The return value of the RPC method as object.

**Return Value**

Type: `AdsErrorCode` [► 575]

The ADS Error Code.

**Implements**

`IAdsRpcInvoke.TryInvokeRpcMethod(String, String, Object., Object.)` [► 901]
Remarks

Because this overload doesn’t provide any AnyTypeSpecifier [1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /*
             * {attribute 'ToRpcEnable'}
             * METHOD PUBLIC M_Add : INT
             * VAR_INPUT
             *   11 : INT := 0;
             *   12 : INT := 0;
             * END_VAR
             *
             */
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});

            // Browsing RpcMethods
            foreach(IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;

                foreach(IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}

Reference

AdsConnection Class [357]

TryInvokeRpcMethod Overload [502]

TwinCAT.Ads Namespace [151]
**AdsConnection.TryInvokeRpcMethod Method** (String, String, Object, Object, Object)

Invokes the specified RPC Method

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object[] outParameters,
    out Object retValue
)
```

**Parameters**

- **symbolPath**
  - Type: `System.String`
  - The symbol path.

- **methodName**
  - Type: `System.String`
  - The method name.

- **inParameters**
  - Type: `System.Object`
  - The parameters.

- **outParameters**
  - Type: `System.Object`
  - The out parameters.

- **retValue**
  - Type: `System.Object`
  - The return value of the RPC method as object.

**Return Value**

Type: `AdsErrorCode`

The ADS Error Code.

**Implements**

IAdsRpcInvoke.TryInvokeRpcMethod(String, String, Object, Object, Object)

**Remarks**

Because this overload doesn't provide any `AnyTypeSpecifier` specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

**Examples**

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

**Dynamic Tree Mode**

```csharp
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
```
// Get the AdsAddress from command-line arguments
AmsAddress address = ArgParser.Parse(args);

using (AdsClient client = new AdsClient())
{
    //client.Synchronize = false;
    // Connect to the target device
    client.Connect(address);

    SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolLoadMode.VirtualTree);
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

    // Get the Symbols (Dynamic Symbols)
    IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

    // Call a Method that has the following signature (within MAIN Program)
    /* (attribute 'TcRpcEnable')
    METHOD PUBLIC M_Add : INT
    VAR_INPUT
    i1 : INT := 0;
    i2 : INT := 0;
    END_VAR
    */

    short result = (short)main.InvokeRpcMethod("M_Add", new object[] { (short) 3, (short) 4 });

    // Call a Method that has no parameter and returns VOID
    main.InvokeRpcMethod("M_Method1", new object[] { });

    // Browsing RpcMethods
    foreach (IRpcMethod method in main.RpcMethods)
    {
        string methodName = method.Name;

        foreach (IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
    }
}

Reference

AdsConnection Class [» 357]

TryInvokeRpcMethod Overload [» 502]

TwinCAT.Ads Namespace [» 151]

AdsConnection.TryInvokeRpcMethod Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.)

Invokes the rpc method.

Namespace: TwinCAT.Ads [» 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
Parameters

symbolPath Type: System.String
The symbol.

methodName Type: System.String
Name of the method.

inParameters Type: System.Object
The parameters.

outSpecifiers Type: TwinCAT.TypeSystem.AnyTypeSpecifier[]
The out specifiers (specifying the out types) or NULL.

retSpecifier Type: TwinCAT.TypeSystem.AnyTypeSpecifier[]
The ret specifier (specifying the return value) or NULL.

outParameters Type: System.Object
The out parameters.

retValue Type: System.Object
The return value of the RPC method.

Return Value

Type: AdsErrorCode[]
AdsErrorCode.

Implements

IAdsRpcInvoke.TryInvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object)[]

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        using (AdsClient client = new AdsClient())
        {
            client.Connect(address);
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);
        }
    }
}
SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

// Get the Symbols (Dynamic Symbols)
IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

// Call a method that has the following signature (within MAIN Program)
/* {attribute 'TcRpcEnable'}
METHOD PUBLIC M_Add : INT
VAR_INPUT
   i1 : INT := 0;
   i2 : INT := 0;
END_VAR
*/
short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short)3, (short)4});

// Call a method that has no parameter and returns VOID
main.InvokeRpcMethod("M_Method1", new object[]{});

// Browsing RpcMethods
foreach(IRpcMethod method in main.RpcMethods)
{
   string methodName = method.Name;
   foreach(IRpcMethodParameter parameter in method.Parameters)
   {
      string parameterName = parameter.Name;
      string parameterType = parameter.TypeName;
   }
}

Reference

AdsConnection Class [357]

TryInvokeRpcMethod Overload [502]

TwinCAT.Ads Namespace [151]


Invokes the rpc method.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public AdsErrorCode TryInvokeRpcMethod(
   IRpcCallableInstance symbol,
   IRpcMethod method,
   Object[] inParameters,
   AnyTypeSpecifier[] outSpecifiers,
   AnyTypeSpecifier retSpecifier,
   out Object[] outParameters,
   out Object retValue
)
Parameters

- **symbol**: Type: `TwinCAT.TypeSystem.IRpcCallableInstance` [2104]
The RPC callable symbol.
- **method**: Type: `TwinCAT.TypeSystem.IRpcMethod` [2123]
The method.
- **inParameters**: Type: `System.Object`
The parameters.
- **outSpecifiers**: Type: `TwinCAT.TypeSystem.AnyTypeSpecifier` [1633]
The out specifiers (specifying the out types) or NULL.
- **retSpecifier**: Type: `TwinCAT.TypeSystem.AnyTypeSpecifier` [1633]
The ret specifier (specifying the return value) or NULL.
- **outParameters**: Type: `System.Object`
The out parameters.
- **retValue**: Type: `System.Object`
The return value of the RPC method.

Return Value

Type: `AdsErrorCode` [575]
`AdsErrorCode`.

Implements

`IAdsRpcInvoke.TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object)` [906]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ObjectDisposedException</code></td>
<td></td>
</tr>
<tr>
<td><code>ClientNotConnectedException</code> [63]</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```csharp
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        AdsClient client = new AdsClient();
    }
}
```
```csharp
// Connect to the target device
client.Connect(address);

SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolLoadMode.VirtualTree);
ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

// Get the Symbols (Dynamic Symbols)
IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

// Call a Method that has the following signature (within MAIN Program)
/*
   METHOD PUBLIC M_Add : INT
      VAR_INPUT
         i1 : INT := 0;
         i2 : INT := 0;
      END_VAR
*/
short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short)3, (short)4});

// Call a Method that has no parameter and returns VOID
main.InvokeRpcMethod("M_Method1", new object[] {});

// Browsing RpcMethods
foreach(IRpcMethod method in main.RpcMethods)
{
    string methodName = method.Name;
    foreach(IRpcMethodParameter parameter in method.Parameters)
    {
        string parameterName = parameter.Name;
        string parameterType = parameter.TypeName;
    }
}
```

Reference

**AdsConnection Class** [357]

**TryInvokeRpcMethod Overload** [502]

**TwinCAT.Ads Namespace** [151]

### 6.2.5.2.45 AdsConnection.TryRead Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="511" alt="TryRead(UInt32, Memory, Void)" /></td>
<td>TryRead(UInt32, Memory, Void) [511]</td>
</tr>
<tr>
<td><img src="511" alt="TryRead(UInt32, UInt32, Memory, Void)" /></td>
<td>TryRead(UInt32, UInt32, Memory, Void) [511]</td>
</tr>
<tr>
<td><img src="512" alt="TryRead(UInt32, UInt32, Memory, Void, Byte)" /></td>
<td>TryRead(UInt32, UInt32, Memory, Void, Byte) [512]</td>
</tr>
</tbody>
</table>
### AdsConnection.TryRead Method (UInt32, Memory`1, Void)

#### Namespace:
TwinCAT.Ads

#### Assembly:
TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

```csharp
public AdsErrorCode TryRead(
    uint variableHandle,
    Memory readBuffer,
    void readBytes
)
```

#### Parameters

- **variableHandle**
  - Type: `System.UInt32`

- **readBuffer**
  - Type: `Memory`

- **readBytes**
  - Type: `System.Void`

#### Return Value

Type: `AdsErrorCode`

### Reference

AdsConnection Class [› 357]

TwinCAT.Ads Namespace [› 151]

### AdsConnection.TryRead Method (UInt32, UInt32, Memory`1, Void)

#### Namespace:
TwinCAT.Ads

#### Assembly:
TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

```csharp
public AdsErrorCode TryRead(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void readBytes
)
```

### Reference

AdsConnection Class [› 357]

TryRead Overload [› 510]

TwinCAT.Ads Namespace [› 151]

### AdsConnection.TryRead Method (UInt32, UInt32, Memory`1, Void)

#### Namespace:
TwinCAT.Ads

#### Assembly:
TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

```csharp
public AdsErrorCode TryRead(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void readBytes
)
```
Parameters

indexGroup Type: System.UInt32
indexOffset Type: System.UInt32
readBuffer Type: Memory
readBytes Type: System.Byte

Return Value

Type: AdsErrorCode [575]

Reference

AdsConnection Class [357]
TryRead Overload [510]
TwinCAT.Ads Namespace [151]

AdsConnection.TryRead Method (UInt32, UInt32, Memory`1, Void, Byte)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public AdsErrorCode TryRead(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void timeout,
    byte readBytes
)

Parameters

indexGroup Type: System.UInt32
indexOffset Type: System.UInt32
readBuffer Type: Memory
timeout Type: System.Void
readBytes Type: System.Byte

Return Value

Type: AdsErrorCode [575]
6.2.5.2.46  AdsConnection.TryReadDataType Method

Call this method to obtain information about the specified data type.

Namespace:  TwinCAT.Ads [151]
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public AdsErrorCode TryReadDataType(
    string typeName,
    out IDataType dataType
)

Parameters

typeName               Type: System.String
                        Name of the symbol.

dataType               Type: TwinCAT.TypeSystem.IDataType [1986],
                        The symbol.

Return Value

Type: AdsErrorCode [575]
A IDataType [1986] containing the requested symbol information or null if symbol could not be found.

Implements

IAdsSymbolicAccess.TryReadDataType(String, IDataType.) [952]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td></td>
</tr>
<tr>
<td>OutOfRangeException</td>
<td></td>
</tr>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>[63]</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]
6.2.5.2.47 AdsConnection.TryReadState Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryReadState(StateInfo.)</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.</td>
</tr>
<tr>
<td>TryReadState(Int32, StateInfo.)</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

AdsConnection.TryReadState Method (StateInfo.)

Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryReadState(
    out StateInfo stateInfo
)
```

Parameters

stateInfo Type: TwinCAT.Ads.StateInfo [1041]. The ADS statue and device status.

Return Value

Type: AdsErrorCode [575]
AdsErrorCode of the ads read state call. Check for AdsErrorCode.NoError to see if call was successful.

Implements

IAdsStateProvider.TryReadState(StateInfo.) [932]

Reference

AdsConnection Class [357]
TryReadState Overload [514]

TwinCAT.Ads Namespace [151]

**AdsConnection.TryReadState Method (Int32, StateInfo.)**

Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AdsErrorCode TryReadState(
    int timeout,
    out StateInfo stateInfo
)
```

**Parameters**

- **timeout**
  
  Type: System.Int32
  The timeout.

- **stateInfo**
  
  Type: TwinCAT.Ads.StateInfo [1041].
  The ADS statue and device status.

**Return Value**

Type: AdsErrorCode [575]
AdsErrorCode of the ads read state call. Check for AdsErrorCode.NoError to see if call was successful.

**Implements**

IAdsStateControlTimeout.TryReadState(Int32, StateInfo.) [924]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

- AdsConnection Class [357]
- TryReadState Overload [514]
- TwinCAT.Ads Namespace [151]

6.2.5.2.48 AdsConnection.TryReadSymbol Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.
### Namespace

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public AdsErrorCode TryReadSymbol(
    string symbolPath,
    out IAdsSymbol symbol
)
```

### Parameters

- **symbolPath**
  - Type: System.String
  - Name of the symbol.

- **symbol**
  - Type: TwinCAT.Ads.TypeSystem.IAdsSymbol [1379]
  - The symbol.

### Return Value

- Type: AdsErrorCode [575]

A IAdsSymbol [1379] containing the requested symbol information or null if symbol could not be found.

### Implements

IAdsSymbolicAccess.TryReadSymbol(String, IAdsSymbol) [953]

### Reference

- AdsConnection Class [357]

### TwinCAT.Ads Namespace [151]

#### 6.2.5.2.49 AdsConnection.TryReadValue Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryReadValue&lt;T&gt;(String, T) [517]</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
<tr>
<td>TryReadValue&lt;T&gt;(ISymbol, T) [518]</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>TryReadValue&lt;T&gt;(ISymbol, Object) [517]</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryReadValue&lt;String, Type, Object) [519]</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
</tbody>
</table>

### Reference

- AdsConnection Class [357]
TwinCAT.Ads Namespaces

**AdsConnection.TryReadValue<T>. Method (String, T.)**

Reads the value of a symbol and returns the value as object.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll)

Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AdsErrorCode TryReadValue<T>(
    string name,
    out T value
)
```

**Parameters**

- **name**
  - Type: System.String
  - Name of the ADS symbol.

- **value**
  - Type: T.
  - The read value of the Symbol.

**Type Parameters**

- **T**
  - The value type.

**Return Value**

- Type: AdsErrorCode

The AdsErrorCode.

**Implements**

IAdsSymbolicAccess.TryReadValue<T>(String, T.)

**Remarks**

The parameter type must have the same layout as the ADS symbol.

**Reference**

- AdsConnection Class
- TryReadValue Overload
- TwinCAT.Ads Namespace

**AdsConnection.TryReadValue Method (ISymbol, Object.)**

Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.
Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryReadValue(
    ISymbol symbol,
    out Object value
)
```

Parameters

symbol Type: TwinCAT.TypeSystem.ISymbol [2176]
The symbol that should be read.

value Type: System.Object.
The value.

Return Value

Type: AdsErrorCode [575]
Value of the symbol

Implements

IAdsSymbolicAccess.TryReadValue(ISymbol, Object.) [954]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException [63]</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'. Structs are not supported.

Reference

AdsConnection Class [357]

TryReadValue Overload [516]

TwinCAT.Ads Namespace [151]

AdsConnection.TryReadValue.T. Method (ISymbol, T.)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C# public AdsErrorCode TryReadValue<T>(
    ISymbol symbol,
    out T value
)

Parameters

symbol Type: TwinCAT.TypeSystem.ISymbol
    The symbol that should be read.
value Type: T.
    The value.

Type Parameters

T The value type.

Return Value

Type: AdsErrorCode
    The ADS Error Code

Implements

IAdsSymbolicAccess.TryReadValue.T.(ISymbol, T.)

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

AdsConnection Class
TryReadValue Overload
TwinCAT.Ads Namespace

AdsConnection.TryReadValue Method (String, Type, Object.)

Reads the value of a symbol and returns the value as object.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C# public AdsErrorCode TryReadValue(
    string name,
    Type type,
    out Object value
)
Parameters

- **name**
  - Type: System.String
  - Name of the ADS symbol.

- **type**
  - Type: System.Type
  - Managed type of the ADS symbol.

- **value**
  - Type: System.Object
  - The read value of the Symbol.

Return Value

Type: AdsErrorCode [575]
The AdsErrorCode [575].

Implements

IASymbolicAccess.TryReadValue(String, Type, Object) [956]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>[63]</td>
</tr>
</tbody>
</table>

Remarks

The parameter type must have the same layout as the ADS symbol.

Reference

- AdsConnection Class [357]
- TryReadValue Overload [516]
- TwinCAT.Ads Namespace [151]

### 6.2.5.2.50 AdsConnection.TryReadWrite Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ] TryReadWriter(UInt32, Memory, Void, Byte) [521]</td>
<td></td>
</tr>
<tr>
<td>![ ] TryReadWriter(UInt32, UInt32, Memory, Void, Byte) [522]</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, UInt32, Memory, Void, Byte, ReadOnlyMemory)</td>
<td>[522]</td>
</tr>
</tbody>
</table>

**Reference**

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

**AdsConnection.TryReadWrite Method (UInt32, Memory`1, Void, Byte)**

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AdsErrorCode TryReadWrite(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer,
    byte readBytes
)
```

**Parameters**

- **variableHandle** Type: System.UInt32
- **readBuffer** Type: Memory
- **writeBuffer** Type: System.Void
- **readBytes** Type: System.Byte

**Return Value**

Type: AdsErrorCode [575]

**Reference**

AdsConnection Class [357]

TryReadWrite Overload [520]

TwinCAT.Ads Namespace [151]
AdsConnection.TryReadWrite Method (UInt32, UInt32, Memory`1, Void, Byte)

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AdsErrorCode TryReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte readBytes
)
```

**Parameters**

- **indexGroup** Type: System.UInt32
- **indexOffset** Type: System.UInt32
- **readBuffer** Type: Memory
- **writeBuffer** Type: System.Void
- **readBytes** Type: System.Byte

**Return Value**

Type: AdsErrorCode

**Reference**

- AdsConnection Class [151]
- TryReadWrite Overload [520]
- TwinCAT.Ads Namespace [151]

AdsConnection.TryReadWrite Method (UInt32, UInt32, Memory`1, Void, Byte, ReadOnlyMemory`1)

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AdsErrorCode TryReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
)
```
Parameters

indexGroup Type: System.UInt32
indexOffset Type: System.UInt32
readBuffer Type: Memory
writeBuffer Type: System.Void
timeout Type: System.Byte
readBytes Type: ReadOnlyMemory

Return Value

Type: AdsErrorCode [575]

Reference

AdsConnection Class [357]
TryReadWrite Overload [520]
TwinCAT.Ads Namespace [151]

6.2.5.2.51 AdsConnection.TryWrite Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="524" alt="TryWrite" /></td>
<td>TryWrite(UInt32, ReadOnlyMemory) [524]</td>
</tr>
<tr>
<td><img src="524" alt="TryWrite" /></td>
<td>TryWrite(UInt32, UInt32, ReadOnlyMemory) [524]</td>
</tr>
<tr>
<td><img src="525" alt="TryWrite" /></td>
<td>TryWrite(UInt32, UInt32, Void) [525]</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]
AdsConnection.TryWrite Method (UInt32, ReadOnlyMemory‘1)

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public AdsErrorCode TryWrite(
    uint variableHandle,
    ReadOnlyMemory writeBuffer
)
```

**Parameters**

- `variableHandle`  
  Type: System.UInt32

- `writeBuffer`  
  Type: ReadOnlyMemory

**Return Value**

Type: AdsErrorCode [575]

**Reference**

AdsConnection Class [357]  
TryWrite Overload [523]  
TwinCAT.Ads Namespace [151]

AdsConnection.TryWrite Method (UInt32, UInt32, ReadOnlyMemory‘1)

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public AdsErrorCode TryWrite(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer
)
```

**Parameters**

- `indexGroup`  
  Type: System.UInt32

- `indexOffset`  
  Type: System.UInt32

- `writeBuffer`  
  Type: ReadOnlyMemory
Return Value

Type: AdsErrorCode [575]

Reference

AdsConnection Class [357]

TryWrite Overload [523]

TwinCAT.Ads Namespace [151]

AdsConnection.TryWrite Method (UInt32, UInt32, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

class public AdsErrorCode TryWrite(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer,
    void timeout
)

Parameters

indexGroup Type: System.UInt32
indexOffset Type: System.UInt32
writeBuffer Type: ReadOnlyMemory
timeout Type: System.Void

Return Value

Type: AdsErrorCode [575]

Reference

AdsConnection Class [357]

TryWrite Overload [523]

TwinCAT.Ads Namespace [151]
### TwinCAT.Ads Namespaces

#### 6.2.5.2.52 AdsConnection.TryWriteControl Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryWriteControl(StateInfo) ![526]</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, Int32) ![527]</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, ReadOnlyMemory) ![528]</td>
<td></td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, ReadOnlyMemory, Void) ![528]</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

AdsConnection Class ![357]

TwinCAT.Ads Namespace ![151]

**AdsConnection.TryWriteControl Method (StateInfo)**

Changes the ADS status and the device status of an ADS server.

**Namespace:** TwinCAT.Ads ![151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AdsErrorCode TryWriteControl(
    StateInfo stateInfo
)
```

**Parameters**

- `stateInfo` Type: TwinCAT_Ads_StateInfo ![1041]
  
New ADS status and device status.

**Return Value**

Type: AdsErrorCode ![575]

AdsErrorCode.
Implements

IAdsStateControl.TryWriteControl(StateInfo) [917]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]

TryWriteControl Overload [526]

TwinCAT.Ads Namespace [151]

**AdsConnection.TryWriteControl Method (StateInfo, Int32)**

Changes the ADS status and the device status of an ADS server.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public AdsErrorCode TryWriteControl(
    StateInfo stateInfo,
    int timeout
)
```

**Parameters**

- **stateInfo**  
  Type: TwinCAT.Ads.StateInfo [1041]  
  New ADS status and device status.

- **timeout**  
  Type: System.Int32  
  The timeout.

**Return Value**

Type: AdsErrorCode [575]  
AdsErrorCode.

Implements

IAdsStateControlTimeout.TryWriteControl(StateInfo, Int32) [925]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>
AdsConnection.TryWriteControl Method (StateInfo, ReadOnlyMemory`1)

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public AdsErrorCode TryWriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer
)
```

**Parameters**

- `stateInfo` Type: TwinCAT.Ads.StateInfo
- `writeBuffer` Type: ReadOnlyMemory

**Return Value**

Type: AdsErrorCode

**Reference**

AdsConnection Class

TryWriteControl Overload

TwinCAT.Ads Namespace

AdsConnection.TryWriteControl Method (StateInfo, ReadOnlyMemory`1, Void)

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public AdsErrorCode TryWriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer,
    void timeout
)
```
Parameters

stateInfo
Type: TwinCAT.Ads.StateInfo

writeBuffer
Type: ReadOnlyMemory

timeout
Type: System.Void

Return Value
Type: AdsErrorCode

Reference

AdsConnection Class
TryWriteControl Overload
TwinCAT.Ads Namespace

6.2.5.2.53 AdsConnection.TryWriteValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="529" alt="TryWriteValue(String, Object)" /></td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td><img src="530" alt="TryWriteValue(T, String, T)" /></td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td><img src="531" alt="TryWriteValue(ISymbol, Object)" /></td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td><img src="532" alt="TryWriteValue(T, ISymbol, T)" /></td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class
TwinCAT.Ads Namespace

AdsConnection.TryWriteValue Method (String, Object)

Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public AdsErrorCode TryWriteValue(
    string name,
    Object value
)
```

**Parameters**

- **name**
  - Type: `System.String`
  - Name of the ADS symbol.

- **value**
  - Type: `System.Object`
  - Object holding the value to be written to the ADS symbol

**Return Value**

Type: `AdsErrorCode` [575]

`AdsErrorCode`.

**Implements**

`IAdsSymbolicAccess.TryWriteValue(String, Object)` [957]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ObjectDisposedException</code></td>
<td></td>
</tr>
<tr>
<td><code>ClientNotConnectedException</code></td>
<td>63</td>
</tr>
</tbody>
</table>

**Reference**

- `AdsConnection Class` [357]
- `TryWriteValue Overload` [529]
- `TwinCAT.Ads Namespace` [151]

**AdsConnection.TryWriteValue<T> Method (String, T)**

Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.

**Namespace:** `TwinCAT.Ads` [151]

**Assembly:** `TwinCAT.Ads (in TwinCAT.Ads.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AdsErrorCode TryWriteValue<T>(
    string name,
    T value
)
```
Parameters

name  
Type: System.String  
Name of the ADS symbol.

value  
Type: T  
Object holding the value to be written to the ADS symbol

Type Parameters

T  
The value type.

Return Value

Type: AdsErrorCode [575]
AdsErrorCode.

Implements

IAdsSymbolicAccess.TryWriteValue.T.(String, T) [958]

Reference

AdsConnection Class [357]
TryWriteValue Overload [529]
TwinCAT.Ads Namespace [151]

AdsConnection.TryWriteValue Method (ISymbol, Object)

 Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public AdsErrorCode TryWriteValue(
    ISymbol symbol,
    Object val
)

Parameters

symbol  
Type: TwinCAT.TypeSystem.ISymbol [2176]
The symbol the value is written to.

val  
Type: System.Object  
The value to write.

Return Value

Type: AdsErrorCode [575]
AdsErrorCode.
TwinCAT.Ads Namespaces

Implements

IAdsSymbolicAccess.TryWriteValue(ISymbol, Object) [p. 959]

Reference

AdsConnection Class [p. 357]
TryWriteValue Overload [p. 529]
TwinCAT.Ads Namespace [p. 151]

AdsConnection.TryWriteValue<T>. Method (ISymbol, T)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [p. 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public AdsErrorCode TryWriteValue<T>(
    ISymbol symbol,
    T val
)

Parameters

symbol

Type: TwinCAT.TypeSystem.ISymbol [p. 2176]
The symbol the value is written to.

val

Type: T
The value to write.

Type Parameters

T

The value type.

Return Value

Type: AdsErrorCode [p. 575]
AdsErrorCode.

Implements

IAdsSymbolicAccess.TryWriteValue<T>(ISymbol, T) [p. 960]

Reference

AdsConnection Class [p. 357]
TryWriteValue Overload [p. 529]
TwinCAT.Ads Namespace [p. 151]
6.2.5.2.54  AdsConnection.UnregisterAdsStateChangedAsync Method

Registers for AdsStateChanged [67] events as an asynchronous operation.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
public Task<ResultAds> UnregisterAdsStateChangedAsync(
    EventHandler<AdsStateChangedEventArgs> handler,
    CancellationToken cancel
)
```

**Parameters**

- **handler**
  Type: `System.EventHandler<AdsStateChangedEventArgs>` [627]
  The handler function to be unregistered.

- **cancel**
  Type: `System.Threading.CancellationToken`
  The cancellation token.

**Return Value**

Type: `Task<ResultAds>` [989]
A task that represents the asynchronous 'UnregisterAdsStateChanged' operation. The `ResultAds` [989] parameter contains the state the `ErrorCode` [992] of the ADS communication after execution.

**Implements**

`IAdsStateProvider.UnregisterAdsStateChangedAsync(EventHandler<AdsStateChangedEventArgs>, CancellationToken)` [933]

**Reference**

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

6.2.5.2.55  AdsConnection.UnregisterSymbolVersionChangedAsync Method

Unregisters the symbol version changed asynchronous.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
public Task<ResultAds> UnregisterSymbolVersionChangedAsync(
    EventHandler<AdsSymbolVersionChangedEventArgs> handler,
    CancellationToken cancel
)
```

**Parameters**

- **handler**
  Type: `System.EventHandler<AdsSymbolVersionChangedEventArgs>` [627]
  The handler function to be unregistered.

- **cancel**
  Type: `System.Threading.CancellationToken`
  The cancellation token.

**Implements**

`IAdsStateProvider.UnregisterSymbolVersionChangedAsync(EventHandler<AdsSymbolVersionChangedEventArgs>, CancellationToken)` [933]

**Reference**

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]
Parameters

**handler**
Type: `System.EventHandler<AdsSymbolVersionChangedEventArgs>`
The handler function to unregister.

**cancel**
Type: `System.Threading.CancellationToken`
The cancellation token.

Return Value

Type: `Task<ResultAds>`
A task that represents the asynchronous 'UnregisterSymbolVersionChangedAsync' operation. The `ResultAds` parameter contains the value `ErrorCode` of the ADS communication after execution.

Implements

`IAdsSymbolChangedProvider.UnregisterSymbolVersionChangedAsync(EventHandler<AdsSymbolVersionChangedEventArgs>, CancellationToken)`

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

`AdsConnection Class`  
`TwinCAT.Ads Namespace`  

6.2.5.2.56 AdsConnection.Write Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Write(UInt32, ReadOnlyMemory)</code></td>
<td>Trigger Client Method/Command.</td>
</tr>
<tr>
<td><code>Write(UInt32, UInt32)</code></td>
<td>Trigger Client Method/Command.</td>
</tr>
<tr>
<td><code>Write(UInt32, UInt32, Int32)</code></td>
<td>Trigger Client Method/Command.</td>
</tr>
<tr>
<td><code>Write(UInt32, UInt32, ReadOnlyMemory)</code></td>
<td>Trigger Client Method/Command.</td>
</tr>
<tr>
<td><code>Write(UInt32, UInt32, ReadOnlyMemory, Void)</code></td>
<td>Trigger Client Method/Command.</td>
</tr>
</tbody>
</table>
AdsConnection.Write Method (UInt32, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void Write(
    uint variableHandle,
    ReadOnlyMemory writeBuffer
)
```

Parameters

- `variableHandle` Type: System.UInt32
- `writeBuffer` Type: ReadOnlyMemory

Reference

AdsConnection Class [357]

AdsConnection.Write Method (UInt32, UInt32)

Trigger Client Method/Command.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void Write(
    uint indexGroup,
    uint indexOffset
)
```

Parameters

- `indexGroup` Type: System.UInt32
  Contains the index group number of the requested ADS service.
- `indexOffset` Type: System.UInt32
  Contains the index offset number of the requested ADS service.
**Implements**

IAdsReadWrite2.Write(UInt32, UInt32) [878]

**Remarks**

This method is used to trigger Client Methods/Commands without parameters.

**Reference**

AdsConnection Class [357]

Write Overload [534]

TwinCAT.Ads Namespace [151]

---

**AdsConnection.Write Method (UInt32, UInt32, Int32)**

Trigger Client Method/Command.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void Write(  
    uint indexGroup,  
    uint indexOffset,  
    int timeout  
)
```

**Parameters**

- **indexGroup**
  
  Type: System.UInt32
  
  Contains the index group number of the requested ADS service.

- **indexOffset**
  
  Type: System.UInt32
  
  Contains the index offset number of the requested ADS service.

- **timeout**
  
  Type: System.Int32
  
  The timeout.

**Remarks**

This method is used to trigger Client Methods/Commands without parameters.

**Reference**

AdsConnection Class [357]

Write Overload [534]

TwinCAT.Ads Namespace [151]
AdsConnection.Write Method (UInt32, UInt32, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public void Write(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer
)

Parameters

- indexGroup Type: System.UInt32
- indexOffset Type: System.UInt32
- writeBuffer Type: ReadOnlyMemory

Reference

AdsConnection Class
Write Overload
TwinCAT.Ads Namespace

AdsConnection.Write Method (UInt32, UInt32, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public void Write(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer,
    void timeout
)

Parameters

- indexGroup Type: System.UInt32
- indexOffset Type: System.UInt32
- writeBuffer Type: ReadOnlyMemory
- timeout Type: System.Void
### AdsConnection.WriteAny Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteAny(UInt32, Object) [538]</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object, Int32) [539]</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAny(UInt32, UInt32, Object) [540]</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAny(UInt32, UInt32, Object, Int32) [541]</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
</tbody>
</table>

### Reference

**AdsConnection Class** [357]

**TwinCAT.Ads Namespace** [151]
Parameters

<table>
<thead>
<tr>
<th>parameter</th>
<th>type</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variableHandle</td>
<td>System.UInt32</td>
<td>handle of the ADS variable.</td>
</tr>
<tr>
<td>value</td>
<td>System.Object</td>
<td>object to write to the ADS device.</td>
</tr>
</tbody>
</table>

Implements

IAdsAnyAccess.WriteAny(UInt32, Object) [726]

Reference

AdsConnection Class [357]
WriteAny Overload [538]
TwinCAT.Ads Namespace [151]

AdsConnection.WriteAny Method (UInt32, Object, Int32.)

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc720ea15da1c14

Syntax

C#
public void WriteAny(
    uint variableHandle,
    Object value,
    int[] args
)

Parameters

<table>
<thead>
<tr>
<th>parameter</th>
<th>type</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variableHandle</td>
<td>System.UInt32</td>
<td>handle of the ADS variable.</td>
</tr>
<tr>
<td>value</td>
<td>System.Object</td>
<td>object to write to the ADS device.</td>
</tr>
<tr>
<td>args</td>
<td>System.Int32</td>
<td>additional arguments.</td>
</tr>
</tbody>
</table>

Implements

IAdsAnyAccess.WriteAny(UInt32, Object, Int32.) [727]

Reference

AdsConnection Class [357]
WriteAny Overload [538]
TwinCAT.Ads Namespace [151]
**AdsConnection.WriteAny Method (UInt32, UInt32, Object)**

Writes an object synchronously to an ADS device.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value
)
```

**Parameters**

- **indexGroup**
  - Type: `System.UInt32`
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - Type: `System.UInt32`
  - Contains the index offset number of the requested ADS service.

- **value**
  - Type: `System.Object`
  - Object to write to the ADS device.

**Implements**

`IAdsAnyAccess.WriteAny(UInt32, UInt32, Object)`

**Reference**

- AdsConnection Class
- WriteAny Overload
- TwinCAT.Ads Namespace

**AdsConnection.WriteAny Method (UInt32, UInt32, Object, Int32.)**

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args
)
```
Parameters

- **indexGroup**
  - Type: `System.UInt32`
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - Type: `System.UInt32`
  - Contains the index offset number of the requested ADS service.

- **value**
  - Type: `System.Object`
  - Object to write to the ADS device.

- **args**
  - Type: `System.Int32`
  - Additional arguments.

Implements

- IAdsAnyAccess.WriteAny(UInt32, UInt32, Object, Int32)

Reference

- AdsConnection Class [357]
- WriteAny Overload [538]
- TwinCAT.Ads Namespace [151]

### AdsConnection.WriteAny Method (UInt32, UInt32, Object, Int32, Int32)

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

**C#**

```csharp
public void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args,
    int timeout
)
```

**Parameters**

- **indexGroup**
  - Type: `System.UInt32`
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - Type: `System.UInt32`
  - Contains the index offset number of the requested ADS service.

- **value**
  - Type: `System.Object`
  - Object to write to the ADS device.

- **args**
  - Type: `System.Int32`
  - Additional arguments.

- **timeout**
  - Type: `System.Int32`
  - The timeout.
TwinCAT.Ads Namespaces

Implements

IAdsReadWriteTimeoutAccess.WriteAny(UInt32, UInt32, Object, _Int32_, Int32) [¶ 885]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [¶ 357]

WriteAny Overload [¶ 538]

TwinCAT.Ads Namespace [¶ 151]

6.2.5.2.58 AdsConnection.WriteAnyAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="542" alt="WriteAnyAsync" /></td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td><img src="543" alt="WriteAnyAsync" /></td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td><img src="545" alt="WriteAnyAsync" /></td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td><img src="545" alt="WriteAnyAsync" /></td>
<td>write any as an asynchronous operation.</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [¶ 357]

TwinCAT.Ads Namespace [¶ 151]

AdsConnection.WriteAnyAsync Method (UInt32, Object, CancellationToken)

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public Task<ResultWrite> WriteAnyAsync(
    uint variableHandle,
    Object value,
    CancellationToken cancel
)
```

### Parameters

- **variableHandle**
  - Type: System.UInt32
  - Handle of the ADS variable.

- **value**
  - Type: System.Object
  - Object to write to the ADS device.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

### Return Value

- Type: Task<ResultWrite>
  - A task that represents the asynchronous task operation. The result parameter ResultWrite of the write operation contains the ErrorCode.

### Implements

- IAdsAnyAccess.WriteAnyAsync(UInt32, Object, CancellationToken)

### Remarks

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

### Reference

- AdsConnection Class
- WriteAnyAsync Overload
- TwinCAT.Ads Namespace

**AdsConnection.WriteAnyAsync Method (UInt32, Object, .Int32., CancellationToken)**

Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
Syntax

C#

```csharp
public Task<ResultWrite> WriteAnyAsync(
    uint variableHandle,
    Object value,
    int[] args,
    CancellationToken cancel
)
```

Parameters

- variableHandle  
  Type: System.UInt32  
  Handle of the ADS variable.

- value  
  Type: System.Object  
  Object to write to the ADS device.

- args  
  Type: System.Int32  
  Additional arguments.

- cancel  
  Type: System.Threading.CancellationToken  
  The cancellation token.

Return Value

Type: Task<ResultWrite>  
Task<ResultWrite>.

Implements

IAdsAnyAccess.WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) 

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotFoundException [63]</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395].</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

- AdsConnection Class [357]
- WriteAnyAsync Overload [542]
- TwinCAT.Ads Namespace [151]
AdsConnection.WriteAnyAsync Method (UInt32, UInt32, Object, CancellationToken)

Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

- **Namespace**: TwinCAT.Ads
- **Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Task<ResultWrite> WriteAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Object value,
    CancellationToken cancel
)
```

**Parameters**

- **indexGroup**
  - Type: System.UInt32
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - Type: System.UInt32
  - Contains the index offset number of the requested ADS service.

- **value**
  - Type: System.Object
  - Object to write to the ADS device.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

**Return Value**

- **Type**: Task<ResultWrite>
  - A task that represents the asynchronous task operation. The result parameter `ResultWrite` of the write operation contains the `ErrorCode`.

**Implements**

- `IAdsAnyAccess.WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)`

**Reference**

- `AdsConnection Class` ([357])
- `WriteAnyAsync Overload` ([542])
- `TwinCAT.Ads Namespace` ([151])

**AdsConnection.WriteAnyAsync Method (UInt32, UInt32, Object, .Int32., CancellationToken)**

write any as an asynchronous operation.

- **Namespace**: TwinCAT.Ads
- **Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public Task<ResultWrite> WriteAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args,
    CancellationToken cancel
)
```

Parameters

- **indexGroup**
  Type: **System.UInt32**
  Contains the index group number of the requested ADS service.

- **indexOffset**
  Type: **System.UInt32**
  Contains the index offset number of the requested ADS service.

- **value**
  Type: **System.Object**
  Object to write to the ADS device.

- **args**
  Type: **System.Int32**
  Additional arguments.

- **cancel**
  Type: **System.Threading.CancellationToken**
  The cancellation token.

Return Value

Type: **Task<ResultWrite>**

A task that represents the asynchronous task operation. The result parameter **ResultWrite** of the write operation contains the **ErrorCode**.

Implements

- **IAdsAnyAccess.WriteAnyAsync(UInt32, UInt32, Object, Int32, CancellationToken)**

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ObjectDisposedException</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ClientNotConnectedException</strong> [63]</td>
<td></td>
</tr>
</tbody>
</table>

Reference

- **AdsConnection Class** [357]
- **WriteAnyAsync Overload** [542]
- **TwinCAT.Ads Namespace** [151]
6.2.5.2.59 AdsConnection.WriteAnyStringAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteAnyStringAsync(String, String, Int32, Encoding, CancellationToken)</td>
<td>write any string as an asynchronous operation.</td>
</tr>
<tr>
<td>WriteAnyStringAsync(UInt32, String, Int32, Encoding, CancellationToken)</td>
<td>write any string as an asynchronous operation.</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

AdsConnection.WriteAnyStringAsync Method (String, String, Int32, Encoding, CancellationToken)

write any string as an asynchronous operation.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C# public Task<ResultWrite> WriteAnyStringAsync(
    string symbolPath,
    string value,
    int length,
    Encoding encoding,
    CancellationToken cancel
)

Parameters

- symbolPath: Type: System.String
  The symbol path.
- value: Type: System.String
  The value.
- length: Type: System.Int32
  The length of the string to write
- encoding: Type: System.Text.Encoding
  The encoding.
- cancel: Type: System.Threading.CancellationToken
  The cancellation token.
Return Value

Type: Task<ResultWrite>
Task<ResultWrite>.

Remarks

ATTENTION: Potentially this method is unsafe because following data can be overwritten after the string symbol. Please be sure to specify the string length lower than the string size reserved within the process image! The String is written with the specified encoding.

Reference

AdsConnection Class [357]

WriteAnyAsync Overload [547]

TwinCAT.Ads Namespace [151]

**AdsConnection.WriteAnyAsync Method (UInt32, String, Int32, Encoding, CancellationToken)**

write any string as an asynchronous operation.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```c#  
public Task<ResultWrite> WriteAnyStringAsync(
    uint variableHandle,
    string value,
    int length,
    Encoding encoding,
    CancellationToken cancel)
```  

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variableHandle</td>
<td>System.UInt32</td>
<td>The variable handle.</td>
</tr>
<tr>
<td>value</td>
<td>System.String</td>
<td>The value.</td>
</tr>
<tr>
<td>length</td>
<td>System.Int32</td>
<td>The length of the string to write</td>
</tr>
<tr>
<td>encoding</td>
<td>System.Text.Encoding</td>
<td>The encoding.</td>
</tr>
<tr>
<td>cancel</td>
<td>System.Threading.CancellationToken</td>
<td>The cancellation token.</td>
</tr>
</tbody>
</table>

**Return Value**

Type: Task<ResultWrite>
Task<ResultWrite>. 
Remarks
ATTENTION: Potentially this method is unsafe because following data can be overwritten after the string symbol. Please be sure to specify the string length lower than the string size reserved within the process image! The String is written with the specified encoding.

Reference
AdsConnection Class [357]
WriteAnyAsStringAsync Overload [547]
TwinCAT.Ads Namespace [151]

6.2.5.2.60 AdsConnection.WriteAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteAsync(UInt32, ReadOnlyMemory, Void) [549]</td>
<td>Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.</td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, CancellationToken) [550]</td>
<td></td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void) [551]</td>
<td></td>
</tr>
</tbody>
</table>

Reference
AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

AdsConnection.WriteAsync Method (UInt32, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultWrite> WriteAsync(
    uint variableHandle,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

Parameters

variableHandle Type: System.UInt32
Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.

**Namespace**: TwinCAT.Ads

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public Task<ResultWrite> WriteAsync(
    uint indexGroup,
    uint indexOffset,
    CancellationToken cancel
)
```

**Parameters**

- **indexGroup**: `System.UInt32`<br> The index group.
- **indexOffset**: `System.UInt32`<br> The index offset.
- **cancel**: `System.Threading.CancellationToken`<br> The cancellation token.

**Return Value**

Type: `Task<ResultWrite>`

A task that represents the asynchronous ‘ReadWrite’ operation. The `ResultWrite` parameter contains the `ErrorCode` after execution.
AdsConnection.WriteAsync Method (UInt32, UInt32, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public Task<ResultWrite> WriteAsync(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer,
    void cancel
)

Parameters

indexGroup Type: System.UInt32
indexOffset Type: System.UInt32
writeBuffer Type: ReadOnlyMemory
cancel Type: System.Void

Return Value

Type: Task<ResultWrite> [1032].

Reference

AdsConnection Class [357]
WriteAsync Overload [549]
TwinCAT.Ads Namespace [151]

6.2.5.2.61 AdsConnection.WriteControl Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>WriteControl(StateInfo, Int32)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>WriteControl(StateInfo, ReadOnlyMemory)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteControl(StateInfo, ReadOnlyMemory, Void)</td>
<td>[554]</td>
</tr>
</tbody>
</table>

**Reference**

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

**AdsConnection.WriteControl Method (StateInfo)**

Changes the ADS status and the device status of an ADS server.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void WriteControl(
    StateInfo stateInfo
)
```

**Parameters**

- **stateInfo** Type: TwinCAT.Ads.StateInfo [1041]
  
  New ADS status and device status.

**Implements**

IAdsStateControl.WriteControl(StateInfo) [918]

**Reference**

AdsConnection Class [357]

WriteControl Overload [551]

TwinCAT.Ads Namespace [151]

**AdsConnection.WriteControl Method (StateInfo, Int32)**

Changes the ADS status and the device status of an ADS server.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public void WriteControl(
    StateInfo stateInfo,
    int timeout
)
```

Parameters

- `stateInfo` Type: TwinCAT.Ads.StateInfo
  New ADS status and device status.
- `timeout` Type: System.Int32
  The timeout.

Implements

- IAdsStateControlTimeout.WriteControl(StateInfo, Int32)

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

- AdsConnection Class
- WriteControl Overload
- TwinCAT.Ads Namespace

AdsConnection.WriteControl Method (StateInfo, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void WriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer
)
```

Parameters

- `stateInfo` Type: TwinCAT.Ads.StateInfo
  New ADS status and device status.
- `writeBuffer` Type: ReadOnlyMemory
  The timeout.

Reference

- AdsConnection Class
**AdsWithControl Overload [551]**

**TwinCAT.Ads Namespace [151]**

---

**AdsWithConnection.WriteControl Method (StateInfo, ReadOnLyMemory`1, Void)**

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void WriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer,
    void timeout
)
```

**Parameters**

- **stateInfo**
  - Type: TwinCAT.Ads.StateInfo [1041]

- **writeBuffer**
  - Type: ReadOnlyMemory

- **timeout**
  - Type: System.Void

**Reference**

- AdsConnection Class [357]
- WriteControl Overload [551]
- TwinCAT.Ads Namespace [151]

---

### 6.2.5.2.62 AdsWithConnection.WriteControlAsync Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteControlAsync(AdsWithState, UInt16, CancellationToken) [555]</td>
<td>Changes the ADS status and device status of the ADS server asynchronously.</td>
</tr>
<tr>
<td>WriteControlAsync(AdsWithState, UInt16, ReadOnLyMemory, Void) [555]</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

- AdsConnection Class [357]
AdsConnection.WriteControlAsync Method (AdsState, UInt16, CancellationToken)

Changes the ADS status and device status of the ADS server asynchronously.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Task<ResultAds> WriteControlAsync(
    AdsState state,
    ushort deviceState,
    CancellationToken cancel
)
```

**Parameters**

- **state**
  
  Type: TwinCAT.Ads.AdsState
  
  The ADS state.

- **deviceState**
  
  Type: System.UInt16
  
  The device state.

- **cancel**
  
  Type: System.Threading.CancellationToken
  
  The cancellation token.

**Return Value**

Type: Task<ResultAds>

A task that represents the asynchronous 'WriteControl' operation. The ResultAds parameter contains the state the ErrorCode of the ADS communication after execution.

**Implements**

IAdsStateControl.WriteControlAsync(AdsState, UInt16, CancellationToken)

**Reference**

- AdsConnection Class
- WriteControlAsync Overload
- TwinCAT.Ads Namespace

AdsConnection.WriteControlAsync Method (AdsState, UInt16, ReadOnlyMemory`1, Void)

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
6.2.5.2.63  AdsConnection.WriteSymbolAsync Method

Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.

Namespace:  TwinCAT.Ads
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultWrite> WriteSymbolAsync(
    string name,
    Object value,
    CancellationToken cancel
)
```

Parameters

name Type: System.String
Name of the ADS symbol.

value Type: System.Object
Object holding the value to be written to the ADS symbol.

cancel Type: System.Threading.CancellationToken
The cancel token.
Return Value

Type: Task.ResultWrite [1032].
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [1032] parameter contains the ErrorCode [992] after execution.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException [63]</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

6.2.5.2.64  AdsConnection.WriteValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValue(String, Object) [557]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue(T (String, T) [558]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue(ISymbol, Object) [559]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValue.T (ISymbol, T) [560]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]
TwinCAT.Ads Namespace [151]

AdsConnection.WriteValue Method (String, Object)

Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
## Syntax

### C#

```csharp
public void WriteValue(
    string name,
    Object value
)
```

### Parameters

- **name**
  - Type: `System.String`
  - Name of the ADS symbol.

- **value**
  - Type: `System.Object`
  - Object holding the value to be written to the ADS symbol

### Implements

`IAdsSymbolicAccess.WriteValue(String, Object)`

### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ObjectDisposedException</code></td>
<td></td>
</tr>
<tr>
<td><code>ClientNotConnectedException</code></td>
<td>[63]</td>
</tr>
</tbody>
</table>

### Reference

- `AdsConnection Class` [357]
- `WriteValue Overload` [557]
- `TwinCAT.Ads Namespace` [151]

### `AdsConnection.WriteValue<T>. Method (String, T)`

Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.

**Namespace:** `TwinCAT.Ads` [151]

**Assembly:** `TwinCAT.Ads.dll` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

### C#

```csharp
public void WriteValue<T>(
    string name,
    T value
)
```

### Parameters

- **name**
  - Type: `System.String`
  - Name of the ADS symbol.
value

Object holding the value to be written to the ADS symbol

**Type Parameters**

T

the value type.

**Implements**

IAdsSymbolicAccess.WriteValue.T.(String, T) [961]

**Reference**

AdsConnection Class [357]

WriteValue Overload [557]

TwinCAT.Ads Namespace [151]

**AdsConnection.WriteValue Method (ISymbol, Object)**

Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void WriteValue(
    ISymbol symbol,
    Object val
)
```

**Parameters**

symbol

Type: TwinCAT.TypeSystem.ISymbol [2176]

The symbol the value is written to.

val

Type: System.Object

The value to write.

**Implements**

IAdsSymbolicAccess.WriteValue(ISymbol, Object) [962]

**Reference**

AdsConnection Class [357]

WriteValue Overload [557]

TwinCAT.Ads Namespace [151]
**AdsConnection.WriteValue.T. Method (ISymbol, T)**

Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void WriteValue<T>(
    ISymbol symbol,
    T val
)
```

**Parameters**

- **symbol**
  - Type: TwinCAT.TypeSystem.ISymbol [2176]
  - The symbol the value is written to.

- **val**
  - Type: T
  - The value to write.

**Type Parameters**

- **T**
  - The value type.

**Implements**

IAdsSymbolicAccess.WriteValue.T.(ISymbol, T) [963]

**Reference**

AdsConnection Class [357]

WriteValue Overload [557]

TwinCAT.Ads Namespace [151]

### 6.2.5.2.65 AdsConnection.WriteValueAsync Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![emoji symbol] WriteValueAsync.T.</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter</td>
</tr>
<tr>
<td>![emoji symbol] WriteValueAsync(ISymbol, Object, CancellationToken)</td>
<td>type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>![emoji symbol] WriteValueAsync(ISymbol, Object, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteValueAsync&lt;T, ISymbol, T, CancellationToken&gt;</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

**AdsConnection.WriteValueAsync<T> Method (String, T, CancellationToken)**

Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [151]  
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public Task<ResultWrite> WriteValueAsync<T>(
    string name,
    T value,
    CancellationToken cancel
)
```

**Parameters**

- **name**  
  Type: System.String  
  Name of the ADS symbol.

- **value**  
  Type: T  
  Object holding the value to be written to the ADS symbol.

- **cancel**  
  Type: System.Threading.CancellationToken  
  The cancel token.

**Type Parameters**

- **T**  
  The value type.

**Return Value**

Type: Task<ResultWrite> [1032].  
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [1032] parameter contains the ErrorCode [992] after execution.

**Implements**

IAdsSymbolicAccess.WriteValueAsync<T, ISymbol, T, CancellationToken> [964]

**Reference**

AdsConnection Class [357]
WriteValueAsync Overload [560]

TwinCAT.Ads Namespace [151]

**AdsConnection.WriteValueAsync Method (ISymbol, Object, CancellationToken)**

Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Task<ResultWrite> WriteValueAsync(
    ISymbol symbol,
    Object val,
    CancellationToken cancel
)
```

**Parameters**

- **symbol**
  - Type: TwinCAT.TypeSystem.ISymbol [2176]
  - The symbol the value is written to.

- **val**
  - Type: System.Object
  - The value to write.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancel token.

**Return Value**

Type: Task<ResultWrite> [1032].

**Implements**

IAdsSymbolicAccess.WriteValueAsync(ISymbol, Object, CancellationToken) [965]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
<tr>
<td>ClientNotConnectedException</td>
<td>[63]</td>
</tr>
</tbody>
</table>

**Reference**

AdsConnection Class [357]

WriteValueAsync Overload [560]

TwinCAT.Ads Namespace [151]
AdsConnection.WriteValueAsync.<T>. Method (ISymbol, T, CancellationToken)

Wrote a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public Task<ResultWrite> WriteValueAsync<T>(
    ISymbol symbol,
    T val,
    CancellationToken cancel
)

Parameters

symbol 
Type: TwinCAT.TypeSystem.ISymbol 
The symbol the value is written to.

val 
Type: T 
The value to write.

cancel 
Type: System.Threading.CancellationToken 
The cancellation token.

Type Parameters

T 
The value type.

Return Value

Type: Task<ResultWrite> 
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite parameter contains the ErrorCode after execution.

Implements

IAdsSymbolicAccess.WriteValueAsync.<T>.(ISymbol, T, CancellationToken)

Reference

AdsConnection Class
WriteValueAsync Overload
TwinCAT.Ads Namespace

6.2.5.3 AdsConnection Events

The AdsConnection type exposes the following members.
Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsNotification [564]</td>
<td>Occurs when the ADS device sends a notification to the client.</td>
</tr>
<tr>
<td>AdsNotificationError [565]</td>
<td>Occurs when a exception has occurred during notification management.</td>
</tr>
<tr>
<td>AdsNotificationEx [566]</td>
<td>Occurs when the ADS devices sends a notification to the client.</td>
</tr>
<tr>
<td>AdsStateChanged [567]</td>
<td>Occurs when ADS State has been changed.</td>
</tr>
<tr>
<td>AdsSumNotification [568]</td>
<td>Occurs when Notifications are send (bundled notifications)</td>
</tr>
<tr>
<td>AdsSymbolVersionChanged</td>
<td>Occurs when the symbol version has been changed.</td>
</tr>
<tr>
<td>ConnectionStateChanged</td>
<td>Occurs when connection status of the AdsConnection [357] has been changed.</td>
</tr>
<tr>
<td>RouterStateChanged</td>
<td>Occurs when [router state changed].</td>
</tr>
</tbody>
</table>

Reference

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

6.2.5.3.1 AdsConnection.AdsNotification Event

Occurs when the ADS device sends a notification to the client.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934f07da3e72bc0ea15da1c14

Syntax

C#

```csharp
public event EventHandler<AdsNotificationEventArgs> AdsNotification
```

Value

Type: System.EventHandler<AdsNotificationEventArgs> [597].

Implements

IAdsNotifications.AdsNotification [866]
Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

The Event Argument contains the raw data value of the notification, not marshalled to .NET types.

Examples

Example of receiving AdsNotification [866] events.

Trigger on changed values by ADS Notifications

```csharp
private async Task RegisterNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;
    using (AdsClient client = new AdsClient())
    {
        // Add the Notification event handler
        client.AdsNotification += Client_AdsNotification2;

        // Connect to target
        client.Connect(AmsNetId.Local, 851);
        uint notificationHandle = 0;

        // Notification to a DINT Type (UINT32)
        // Check for change every 200 ms
        byte[] notificationBuffer = new byte[sizeof(UInt32)];
        int size = sizeof(UInt32);

        ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", size, new NotificationSettings(AdsTransModeOnChange, 200, 0), null, cancel);

        if (result.Succeeded)
        {
            notificationHandle = result.Handle;
            await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
            // Unregister the Event / Handle
            ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
        }

        client.AdsNotification -= Client_AdsNotification2;
    }
}

private void Client_AdsNotification2(object sender, AdsNotificationEventArgs e)
{
    // Or here we know about UDINT type --> can be marshalled as UINT32
    uint nCounter = BinaryPrimitives.ReadUInt32LittleEndian(e.Data.Span);

    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread
    /*SynchronizationContext syncContext = SynchronizationContext.Current;
     _context.Post(status => someLabel.Text = nCounter.ToString(), null); // Non-blocking post */
}
```

Reference

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

6.2.5.3.2 AdsConnection.AdsNotificationError Event

Occurs when a exception has occurred during notification management.
**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public event EventHandler<AdsNotificationErrorEventArgs> AdsNotificationError
```

### Value

**Type:** System.EventHandler<AdsNotificationErrorEventArgs> [595]

### Implements

IAdsNotifications.AdsNotificationError [867]

### Remarks

The occurrence of this event can have two different reasons:

1. Indicates an internal error occurred during Notification management.
2. The registered notification becomes invalid on the server, eg. after a PLC Download / Online Change. If the ADS Server detects that the (still registered) Notification Sender is getting invalid, it sends an error notification so that the client will be informed about detached notifications. The event arguments contains the AdsInvalidNotificationException which describes the invalid notification handle by its Handle property.

### Reference

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

TwinCAT.Ads.AdsInvalidNotificationException [591]

AdsConnection.AdsNotification [564]

AdsConnection.AdsNotificationEx [566]

---

### 6.2.5.3.3 AdsConnection.AdsNotificationEx Event

Occurs when the ADS devices sends a notification to the client.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public event EventHandler<AdsNotificationExEventArgs> AdsNotificationEx
```

### Value

**Type:** System.EventHandler<AdsNotificationExEventArgs> [601]
TwinCAT.Ads Namespaces

**Implements**

IAdsNotifications.AdsNotificationEx [868]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

**Remarks**

The Notification event arguments marshals the data value automatically to the specified .NET Type with ANY_TYPE marshellers.

**Examples**

Example of receiving AdsNotificationEx [868] events.

**Trigger on changed values by ADS Notifications**

```csharp
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.AdsNotificationEx += Client_AdsNotificationEx;
    client.Connect(AmsNetId.Local, 851);

    // Add UDINT
    ResultHandle resultHandle = await client.AddDeviceNotificationExAsync("MAIN.udint", new NotificationSettings(AdsTransModeOnChange, 200, 200), null, typeof(uint), null, cancel);
    await Task.Delay(5000, cancel); // Wait ....
    ResultAds resultHandleDelete = await client.DeleteDeviceNotificationAsync(resultHandle.Handle, cancel); // Unregister Event
}
```

**Reference**

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

### 6.2.5.3.4 AdsConnection.AdsStateChanged Event

Occurs when ADS State has been changed.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public event EventHandler<AdsStateChangedEventArgs> AdsStateChanged
```

**Value**

Type: System.EventHandler<AdsStateChangedEventArgs> [627].
Implements

IAdsStateProvider.AdsStateChanged [› 934]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

This event occurs asynchronously if the synchronized flag is not set.

Reference

AdsConnection Class [› 357]
TwinCAT.Ads Namespace [› 151]

### 6.2.5.3.5 AdsConnection.AdsSumNotification Event

Occurs when Notifications are send (bundled notifications)

Namespace: TwinCAT.Ads [› 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public event EventHandler<AdsSumNotificationEventArgs> AdsSumNotification
```

Value

Type: System.EventHandler<AdsSumNotificationEventArgs>.

Implements

IAdsNotifications.AdsSumNotification [› 868]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

As an optimization, this event receives all ADS Notifications that occurred at one point in time together. As consequence, the overhead of handler code is reduced, what can be important if notifications are triggered in a high frequency and the event has to be synchronized to the UI thread context. Because multiple notifications are bound together, less thread synchronization is necessary. The AdsNotification [› 866] and AdsNotificationEx [› 868] events shouldn't be used when SumNotifications are registered, because they have an performance side effect to this AdsSumNotification [› 868] event. The full performance is reached only, when all notifications are handled on this event.
Examples

Example of receiving AdsSumNotification [868] events.

Trigger on changed values by ADS Notifications

```csharp
private async Task RegisterSumNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;

    using (AdsClient client = new AdsClient())
    {
        // Add the Notification event handler
        client.AdsSumNotification += Client_SumNotification;

        // Connect to target
        client.Connect(AmsNetId.Local, 851);
        uint notificationHandle = 0;

        // Notification to a DINT Type (UINT32)
        // Check for change every 200 ms
        byte[] notificationBuffer = new byte[sizeof(UInt32)];
        ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", sizeof(UInt32), new NotificationSettings(AdsTransModeOnChange, 200, 0), null, cancel);
        if (result.Succeeded)
        {
            notificationHandle = result.Handle;
            await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
            // Unregister the Event / Handle
            ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
        }
    }

    client.AdsNotification -= Client_AdsNotification2;
}

private void Client_SumNotification(object sender, AdsSumNotificationEventArgs e)
{
    // Timestamp of the Notification List
    DateTimeOffset dateTime = e.TimeStamp;

    // List of Raw ADS Notifications
    IList<Notification> notifications = e.Notifications;
    foreach (Notification notification in notifications)
    {
        // Notifications can be handled more efficiently, because they occur together
        // handler and can be transformed/synchronized in one step compared to AdsClient.AdsNotification events.
    }
}
```

Reference

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

IAdsNotifications.AdsNotification [866]

6.2.5.3.6 AdsConnection.AdsSymbolVersionChanged Event

Occurs when the symbol version has been changed.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public event EventHandler<AdsSymbolVersionChangedEventArgs> AdsSymbolVersionChanged
```

Value

Type: `System.EventHandler<AdsSymbolVersionChangedEventArgs>`

Implements

`IAdsSymbolChangedProvider.AdsSymbolVersionChanged`

Remarks

This is the case when the connected ADS server restarts. This invalidates all actual opened symbol handles. The SymbolVersion counter doesn't trigger, when an online change is made on the PLC (ports 801, ..., 851 ...)

Reference

`AdsConnection Class`

`TwinCAT.Ads Namespace`

6.2.5.3.7 AdsConnection.ConnectionStateChanged Event

Occurs when connection status of the `AdsConnection` has been changed.

Namespace: `TwinCAT.Ads`

Assembly: `TwinCAT.Ads (in TwinCAT.Ads.dll)`

Version: 5.0.294+Branch.releases-5.0.A.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public event EventHandler<ConnectionStateChangedEventArgs> ConnectionStateChanged
```

Value

Type: `System.EventHandler<ConnectionStateChangedEventArgs>`

Implements

`IConnectionStateProvider.ConnectionStateChanged`

Remarks

The Connection state changes only if the `IConnection` is established / shut down or active communication is triggered by the User of the `IConnection` object.

Examples

The following sample shows how to keep the `ConnectionState` updated by triggering ADS Communication.
Trigger ConnectionState changes in WPF Applications

```csharp
private DispatcherTimer _timer = null;
private AdsSession _session = null;
//private AdsConnection _connection = null;

private void Window_Loaded(object sender, RoutedEventArgs e)
{
    _session = new AdsSession(AmsNetId.Local, 10000);
    IConnection connection = _session.Connect();
    tbConnectionState.Text = connection.ConnectionState.ToString();
    _session.ConnectionStateChanged += _session_ConnectionStateChanged;

    _timer = new DispatcherTimer();
    _timer.Interval = TimeSpan.FromMilliseconds(200);
    _timer.Tick += TimerOnTick;
    _timer.Start();
}

private void Window_Unloaded(object sender, RoutedEventArgs e)
{
    _timer.Stop();
    _session.Dispose();
}

private void _session_ConnectionStateChanged(object sender, TwinCAT.ConnectionStateChangedEventArgs e)
{
    // ConnectionStateChanged will be triggered by communication Invokes
    tbConnectionState.Text = e.NewState.ToString();
}

private void TimerOnTick(object sender, EventArgs eventArgs)
{
    // The Timer Event will occur here in the UI thread because its an DispatcherTimer event!
    // An active ADS request will trigger Connection State periodically!
    StateInfo stateInfo;
    if (_session.Connection.TryReadState(out stateInfo) == AdsErrorCode.NoError)
    {
        tbAdsState.Text = stateInfo.AdsState.ToString();
    }
    else
    {
        tbAdsState.Text = "Invalid";
    }
}
```

Reference

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

AdsConnection.ConnectionState [379]

6.2.5.3.8 AdsConnection.RouterStateChanged Event

Occurs when [router state changed].

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public event EventHandler<AmsRouterNotificationEventArgs> RouterStateChanged
```
**Value**

Type: `System.EventHandler.AmsRouterNotificationEventArgs` [695].

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDisposedException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

AdsConnection Class [357]

TwinCAT.Ads Namespace [151]

### 6.2.6 AdsDataTypeArrayInfo Class

Array definition for a single dimension.

**Inheritance Hierarchy**

- `System.Object`
- TwinCAT.Ads.AdsDataTypeArrayInfo

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public class AdsDataTypeArrayInfo
```

The `AdsDataTypeArrayInfo` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elements [573]</td>
<td>Gets the number of elements.</td>
</tr>
<tr>
<td>LowerBound [573]</td>
<td>Gets the lower bound.</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
6.2.6.1 AdsDataTypeArrayInfo Properties

The AdsDataTypeArrayInfo [572] type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elements</td>
<td>Gets the number of elements.</td>
</tr>
<tr>
<td>LowerBound</td>
<td>Gets the lower bound.</td>
</tr>
</tbody>
</table>

### AdsDataTypeArrayInfo.Elements Property

Gets the number of elements.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public int Elements { get; }
```

**Property Value**

Type: Int32

### AdsDataTypeArrayInfo.LowerBound Property

Gets the lower bound.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public int LowerBound { get; }
```

Property Value

Type: Int32

Reference

AdsDataTypeArrayInfo Class [572]

TwinCAT.Ads Namespace [151]

6.2.6.2 AdsDataTypeArrayInfo Methods

The AdsDataTypeArrayInfo [572] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

AdsDataTypeArrayInfo Class [572]

TwinCAT.Ads Namespace [151]

6.2.7 AdsDataTypeId Enumeration

ADS data types.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public enum AdsDataTypeId
```
Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADST_VOID</td>
<td>0</td>
<td>Empty Type (0)</td>
</tr>
<tr>
<td>ADST_INT8</td>
<td>16</td>
<td>Integer 8 Bit (16)</td>
</tr>
<tr>
<td>ADST_UINT8</td>
<td>17</td>
<td>Unsigned Bit 8 Bit (17)</td>
</tr>
<tr>
<td>ADST_INT16</td>
<td>2</td>
<td>Integer 16 Bit (2)</td>
</tr>
<tr>
<td>ADST_UINT16</td>
<td>18</td>
<td>Unsigned integer 16 Bit (18)</td>
</tr>
<tr>
<td>ADST_INT32</td>
<td>3</td>
<td>Integer 32 Bit (3)</td>
</tr>
<tr>
<td>ADST_UINT32</td>
<td>19</td>
<td>Unsigned Integer 32 Bit (19)</td>
</tr>
<tr>
<td>ADST_INT64</td>
<td>20</td>
<td>LONG Integer 64 Bit (20)</td>
</tr>
<tr>
<td>ADST_UINT64</td>
<td>21</td>
<td>Unsigned Long integer 64 Bit (21)</td>
</tr>
<tr>
<td>ADST_REAL32</td>
<td>4</td>
<td>Real (32 Bit) (4)</td>
</tr>
<tr>
<td>ADST_REAL64</td>
<td>5</td>
<td>Real 64 Bit (5)</td>
</tr>
<tr>
<td>ADST_BIGTYPE</td>
<td>65</td>
<td>Blob (65)</td>
</tr>
<tr>
<td>ADST_STRING</td>
<td>30</td>
<td>STRING (30)</td>
</tr>
<tr>
<td>ADST_WSTRING</td>
<td>31</td>
<td>WSTRING (31)</td>
</tr>
<tr>
<td>ADST_REAL80</td>
<td>32</td>
<td>ADS REAL80 (32)</td>
</tr>
<tr>
<td>ADST_BIT</td>
<td>33</td>
<td>ADS BIT (33)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace \[151\]

6.2.8 AdsErrorCode Enumeration

Describes the ADS error that occurred.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public enum AdsErrorCode
```

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>-1</td>
<td>None / Uninitialized</td>
</tr>
<tr>
<td>NoError</td>
<td>0</td>
<td>No Error / Succeeded. Error code: 0(0x000).</td>
</tr>
<tr>
<td>Succeeded</td>
<td>0</td>
<td>No Error / Succeeded. Error code: 0(0x000).</td>
</tr>
<tr>
<td>InternalError</td>
<td>1</td>
<td>Internal Error. Error code: 1(0x001).</td>
</tr>
<tr>
<td>NoRTime</td>
<td>2</td>
<td>No Realtime. Error code: 2(0x002).</td>
</tr>
<tr>
<td>LockedMemoryError</td>
<td>3</td>
<td>Allocation locked memory error. Error code: 3(0x003).</td>
</tr>
<tr>
<td>Member name</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>MailBoxError</td>
<td>4</td>
<td>Insert mailbox error. Error code: 4(0x004).</td>
</tr>
<tr>
<td>WrongHMsg</td>
<td>5</td>
<td>Wrong receive HMSG. Error code: 5(0x005).</td>
</tr>
<tr>
<td>TargetPortNotFound</td>
<td>6</td>
<td>Target port not found. Error code: 6(0x006).</td>
</tr>
<tr>
<td>TargetMachineNotFound</td>
<td>7</td>
<td>Target machine not found. Error code: 7(0x007).</td>
</tr>
<tr>
<td>UnknownCommandID</td>
<td>8</td>
<td>Unknown command ID. Error code: 8(0x008).</td>
</tr>
<tr>
<td>BadTaskID</td>
<td>9</td>
<td>Bad task ID. Error code: 9(0x009).</td>
</tr>
<tr>
<td>NoIO</td>
<td>10</td>
<td>No IO. Error code: 10(0x00A).</td>
</tr>
<tr>
<td>UnknownAmsCommand</td>
<td>11</td>
<td>Unknown AMS command. Error code: 11(0x00B).</td>
</tr>
<tr>
<td>Win32Error</td>
<td>12</td>
<td>Win 32 error. Error code: 12(0x00C).</td>
</tr>
<tr>
<td>PortNotConnected</td>
<td>13</td>
<td>Port is not connected. Error code: 13(0x00D).</td>
</tr>
<tr>
<td>InvalidAmsLength</td>
<td>14</td>
<td>Invalid AMS length. Error code: 14(0x00E).</td>
</tr>
<tr>
<td>InvalidAmsNetID</td>
<td>15</td>
<td>Invalid AMS Net ID. Error code: 15(0x00F).</td>
</tr>
<tr>
<td>LowInstallLevel</td>
<td>16</td>
<td>Low Installation level. Error code: 16(0x010).</td>
</tr>
<tr>
<td>NoDebug</td>
<td>17</td>
<td>No debug available. Error code: 17(0x011).</td>
</tr>
<tr>
<td>PortDisabled</td>
<td>18</td>
<td>Port disabled. Error code: 18(0x012).</td>
</tr>
<tr>
<td>PortConnected</td>
<td>19</td>
<td>Port is already connected. Error code: 19(0x013).</td>
</tr>
<tr>
<td>AmsSyncWin32Error</td>
<td>20</td>
<td>AMS Sync Win32 error. Error code: 20(0x014).</td>
</tr>
<tr>
<td>SyncTimeOut</td>
<td>21</td>
<td>AMS Sync timeout. Error code: 21(0x015).</td>
</tr>
<tr>
<td>AmsSyncAmsError</td>
<td>22</td>
<td>AMS Sync AMS error Error code: 22(0x016).</td>
</tr>
<tr>
<td>AmsSyncNoIndexMap</td>
<td>23</td>
<td>AMS Sync no index map. Error code: 23(0x017).</td>
</tr>
<tr>
<td>InvalidAmsPort</td>
<td>24</td>
<td>Invalid AMS port. Error code: 24(0x018).</td>
</tr>
<tr>
<td>NoMemory</td>
<td>25</td>
<td>No memory. Error code: 25(0x019).</td>
</tr>
<tr>
<td>TCPSendError</td>
<td>26</td>
<td>TCP send error. Error code: 26(0x01A).</td>
</tr>
<tr>
<td>Member name</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HostUnreachable</td>
<td>27</td>
<td>Host unreachable. Error code: 27(0x1B).</td>
</tr>
<tr>
<td>AmsInvalidFragment</td>
<td>28</td>
<td>Invalid AMS fragment. Error code: 28(0x1C).</td>
</tr>
<tr>
<td>NoLockedMemory</td>
<td>1280</td>
<td>Router: no locked memory. Error code: 1280(0x500).</td>
</tr>
<tr>
<td>ResizeMemory</td>
<td>1281</td>
<td>Router: The size of the router memory could not be changed. Error code: 1281(0x501).</td>
</tr>
<tr>
<td>MailboxFull</td>
<td>1282</td>
<td>Router: mailbox full. Error code: 1282(0x502).</td>
</tr>
<tr>
<td>DebugBoxFull</td>
<td>1283</td>
<td>Router: The mailbox has reached the maximum number of possible messages. Error code: 1283(0x503).</td>
</tr>
<tr>
<td>UnknownPortType</td>
<td>1284</td>
<td>Router: Unknown Port Type Error code: 1284(0x504).</td>
</tr>
<tr>
<td>RouterNotInitialized</td>
<td>1285</td>
<td>Router: Router is not initialized. Error code: 1285(0x505).</td>
</tr>
<tr>
<td>PortAlreadyInUse</td>
<td>1286</td>
<td>Router: The desired port number is already assigned. Error code: 1286(0x506).</td>
</tr>
<tr>
<td>PortNotRegistered</td>
<td>1287</td>
<td>Router: Port not registered. Error code: 1287(0x507).</td>
</tr>
<tr>
<td>NoMoreQueues</td>
<td>1288</td>
<td>Router: The maximum number of Ports reached. Error code: 1288(0x508).</td>
</tr>
<tr>
<td>InvalidPort</td>
<td>1289</td>
<td>Router: The port is invalid. Error code: 1289(0x509).</td>
</tr>
<tr>
<td>RouterNotActive</td>
<td>1290</td>
<td>Router: TwinCAT Router not active. Error code: 1290(0x50A).</td>
</tr>
<tr>
<td>DeviceError</td>
<td>1792</td>
<td>Error class &lt;device error&gt; Error code: 1792(0x700).</td>
</tr>
<tr>
<td>DeviceServiceNotSupported</td>
<td>1793</td>
<td>Service is not supported by server. Error code: 1793(0x701).</td>
</tr>
<tr>
<td>DeviceInvalidGroup</td>
<td>1794</td>
<td>Invalid index group. Error code: 1794(0x702).</td>
</tr>
<tr>
<td>DeviceInvalidOffset</td>
<td>1795</td>
<td>Invalid index offset. Error code: 1795(0x703).</td>
</tr>
<tr>
<td>DeviceInvalidAccess</td>
<td>1796</td>
<td>Reading/writing not permitted. Error code: 1796(0x704).</td>
</tr>
<tr>
<td>DeviceInvalidSize</td>
<td>1797</td>
<td>Parameter size not correct. Error code: 1797(0x705).</td>
</tr>
<tr>
<td>DeviceInvalidData</td>
<td>1798</td>
<td>Invalid parameter value(s). Error code: 1798(0x706).</td>
</tr>
<tr>
<td>DeviceNotReady</td>
<td>1799</td>
<td>Device is not in a ready state. Error code: 1799(0x707).</td>
</tr>
<tr>
<td>Member name</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>DeviceBusy</td>
<td>1800</td>
<td>Device is busy. Error code: 1800(0x708).</td>
</tr>
<tr>
<td>DeviceInvalidContext</td>
<td>1801</td>
<td>Invalid context (must be in Windows). Error code: 1801(0x709).</td>
</tr>
<tr>
<td>DeviceNoMemory</td>
<td>1802</td>
<td>Out of memory. Error code: 1802(0x70a).</td>
</tr>
<tr>
<td>DeviceInvalidParam</td>
<td>1803</td>
<td>Invalid parameter value(s). Error code: 1803(0x70b).</td>
</tr>
<tr>
<td>DeviceNot Found</td>
<td>1804</td>
<td>Not found(files, ...). Error code: 1804(0x70c).</td>
</tr>
<tr>
<td>DeviceSyntaxError</td>
<td>1805</td>
<td>Syntax error in command or file. Error code: 1805(0x70d).</td>
</tr>
<tr>
<td>DeviceIncompatible</td>
<td>1806</td>
<td>Objects do not match. Error code: 1806(0x70e).</td>
</tr>
<tr>
<td>DeviceExists</td>
<td>1807</td>
<td>Object already exists. Error code: 1807(0x70f).</td>
</tr>
<tr>
<td>DeviceSymbolNotFound</td>
<td>1808</td>
<td>Symbol not found. Error code: 1808(0x7010).</td>
</tr>
<tr>
<td>DeviceSymbolVersionInvalid</td>
<td>1809</td>
<td>Symbol version is invalid. Error code: 1809(0x711).</td>
</tr>
<tr>
<td>DeviceInvalidState</td>
<td>1810</td>
<td>Server is not in a valid state. Error code: 1810(0x712).</td>
</tr>
<tr>
<td>DeviceTransModeNotSupported</td>
<td>1811</td>
<td>ADS transmode is not supported. Error code: 1811(0x713).</td>
</tr>
<tr>
<td>DeviceNotifyHandleInvalid</td>
<td>1812</td>
<td>Notification handle is invalid. Error code: 1812(0x714).</td>
</tr>
<tr>
<td>DeviceClientUnknown</td>
<td>1813</td>
<td>Notification client not registered. Error code: 1813(0x715).</td>
</tr>
<tr>
<td>DeviceNoMoreHandles</td>
<td>1814</td>
<td>No more notification handles. Error code: 1814(0x716).</td>
</tr>
<tr>
<td>DeviceInvalidWatchsize</td>
<td>1815</td>
<td>Size for watch to big. Error code: 1815(0x717).</td>
</tr>
<tr>
<td>DeviceNotInitialized</td>
<td>1816</td>
<td>Device is not initialized. Error code: 1816(0x718).</td>
</tr>
<tr>
<td>DeviceTimeOut</td>
<td>1817</td>
<td>Device has a timeout. Error code: 1817(0x719).</td>
</tr>
<tr>
<td>DeviceNoInterface</td>
<td>1818</td>
<td>Query interface has failed. Error code: 1818(0x71A).</td>
</tr>
<tr>
<td>DeviceInvalidInterface</td>
<td>1819</td>
<td>Wrong interface required. Error code: 1819(0x71B).</td>
</tr>
<tr>
<td>DeviceInvalidCLSID</td>
<td>1820</td>
<td>Class ID is invalid. Error code: 1820(0x71C).</td>
</tr>
<tr>
<td>DeviceInvalidObjectID</td>
<td>1821</td>
<td>Object ID is invalid. Error code: 1821(0x71D).</td>
</tr>
<tr>
<td>DeviceRequestsIsPending</td>
<td>1822</td>
<td>Device: Request is Pending. Error code: 1822(0x71E).</td>
</tr>
<tr>
<td>Member name</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DeviceRequestIsAborted</td>
<td>1823</td>
<td>Device: Request is Aborted. Error code: 1823(0x71F).</td>
</tr>
<tr>
<td>DeviceSignalWarning</td>
<td>1824</td>
<td>Device: Signal warning. Error code: 1824(0x720).</td>
</tr>
<tr>
<td>DeviceInvalidArrayIndex</td>
<td>1825</td>
<td>Device: Invalid Array Index (ADSERR_DEVICE_INVALIDARRAYINDEX) Error code: 1825(0x721).</td>
</tr>
<tr>
<td>DeviceSymbolNotActive</td>
<td>1826</td>
<td>Device: Symbol not Active Error code: 1826(0x722).</td>
</tr>
<tr>
<td>DeviceAccessDenied</td>
<td>1827</td>
<td>Device: Access denied. Error code: 1827(0x723).</td>
</tr>
<tr>
<td>DeviceLicenseNotFound</td>
<td>1828</td>
<td>Device: license not found Error code: 1828(0x724).</td>
</tr>
<tr>
<td>DeviceLicenseExpired</td>
<td>1829</td>
<td>Device: license expired Error code: 1829(0x725).</td>
</tr>
<tr>
<td>DeviceLicenseExceeded</td>
<td>1830</td>
<td>Device: license exceeded Error code: 1830(0x726).</td>
</tr>
<tr>
<td>DeviceLicenseInvalid</td>
<td>1831</td>
<td>Device: license invalid Error code: 1831(0x727).</td>
</tr>
<tr>
<td>DeviceLicenseSystemId</td>
<td>1832</td>
<td>Device: license invalid system id Error code: 1832(0x728).</td>
</tr>
<tr>
<td>DeviceLicenseNoTimeLimit</td>
<td>1833</td>
<td>Device: license not time limited Error code: 1833(0x729).</td>
</tr>
<tr>
<td>DeviceLicenseFutureIssue</td>
<td>1834</td>
<td>Device: license issue time in the future Error code: 1834(0x72A).</td>
</tr>
<tr>
<td>DeviceLicenseTimeToLong</td>
<td>1835</td>
<td>Device: license time period to long Error code: 1835(0x72B).</td>
</tr>
<tr>
<td>DeviceException</td>
<td>1836</td>
<td>Device: Exception in device specific code Error code: 1836(0x72C).</td>
</tr>
<tr>
<td>DeviceLicenseDuplicated</td>
<td>1837</td>
<td>Device: license file read twice Error code: 1837(0x72D).</td>
</tr>
<tr>
<td>DeviceSignatureInvalid</td>
<td>1838</td>
<td>Device: invalid signature Error code: 1838(0x72E).</td>
</tr>
<tr>
<td>DeviceCertificateInvalid</td>
<td>1839</td>
<td>Device: public key certificate Error code: 1839(0x72F).</td>
</tr>
<tr>
<td>DeviceLicenseOemNotFound</td>
<td>1840</td>
<td>Device: public key of OEM unknown Error code: 1840(0x730).</td>
</tr>
<tr>
<td>DeviceLicenseRestricted</td>
<td>1841</td>
<td>Device: license not valid for this system id type Error code: 1841(0x731).</td>
</tr>
<tr>
<td>DeviceLicenseDemoDenied</td>
<td>1842</td>
<td>Device: trial license denied Error code: 1842(0x732).</td>
</tr>
<tr>
<td>DeviceInvalidFcnId</td>
<td>1843</td>
<td>Device: function id is invalid Error code: 1843(0x733).</td>
</tr>
<tr>
<td>Member name</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DeviceOutOfRange</td>
<td>1844</td>
<td>Device: a parameter, an index, an iterator, ... is out of range. Error code: 1844(0x734).</td>
</tr>
<tr>
<td>DeviceInvalidAlignment</td>
<td>1845</td>
<td>Device: invalid alignment. Error code: 1845(0x735).</td>
</tr>
<tr>
<td>DeviceLicensePlatform</td>
<td>1846</td>
<td>Device: license invalid platform level. Error code: 1846(0x736).</td>
</tr>
<tr>
<td>ClientError</td>
<td>1856</td>
<td>Error class &lt;client error&gt;. Error code: 1856(0x740).</td>
</tr>
<tr>
<td>ClientInvalidParameter</td>
<td>1857</td>
<td>Parameter at service is invalid. Error code: 1857(0x741).</td>
</tr>
<tr>
<td>ClientListEmpty</td>
<td>1858</td>
<td>Polling list is empty. Error code: 1858(0x742).</td>
</tr>
<tr>
<td>ClientVariableInUse</td>
<td>1859</td>
<td>Variable connection is already in use. Error code: 1859(0x743).</td>
</tr>
<tr>
<td>ClientDuplicateInvokeID</td>
<td>1860</td>
<td>Invoke ID already in use. Error code: 1860(0x744).</td>
</tr>
<tr>
<td>ClientSyncTimeOut</td>
<td>1861</td>
<td>Timeout has elapsed. Error code: 1861(0x745).</td>
</tr>
<tr>
<td>ClientW32OR</td>
<td>1862</td>
<td>Error in win32 subsystem. Error code: 1862(0x746).</td>
</tr>
<tr>
<td>ClientTimeoutInvalid</td>
<td>1863</td>
<td>Timeout value is invalid. Error code: 1863(0x747).</td>
</tr>
<tr>
<td>ClientPortNotOpen</td>
<td>1864</td>
<td>ADS port is not opened. Error code: 1864(0x748).</td>
</tr>
<tr>
<td>ClientNoAmsAddr</td>
<td>1865</td>
<td>No AMS Address. Error code: 1865(0x749).</td>
</tr>
<tr>
<td>ClientSyncInternal</td>
<td>1872</td>
<td>An internal in ADS sync has occurred. Error code: 1872(0x750).</td>
</tr>
<tr>
<td>ClientAddHash</td>
<td>1873</td>
<td>Hash table overflow. Error code: 1873(0x751).</td>
</tr>
<tr>
<td>ClientRemoveHash</td>
<td>1874</td>
<td>There are no more symbols in the hash table. Error code: 1874(0x752).</td>
</tr>
<tr>
<td>ClientNoMoreSymbols</td>
<td>1875</td>
<td>There are no more symbols in cache. Error code: 1875(0x753).</td>
</tr>
<tr>
<td>ClientSyncResInvalid</td>
<td>1876</td>
<td>An invalid response has been received. Error code: 1876(0x754).</td>
</tr>
<tr>
<td>ClientSyncPortLocked</td>
<td>1877</td>
<td>Sync port is locked. Error code: 1877(0x755).</td>
</tr>
<tr>
<td>ClientQueueFull</td>
<td>32768</td>
<td>Client queue is full.</td>
</tr>
<tr>
<td>WSA_ConnRefused</td>
<td>10061</td>
<td>Windows sockets connection refused (0x274d, 10061).</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]
6.2.9 AdsErrorCodeExtensions Class

Class AdsErrorCodeExtensions.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.AdsErrorCodeExtensions

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static class AdsErrorCodeExtensions
```

The AdsErrorCodeExtensions type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed</td>
<td>Indicates, that the communication failed with AdsErrorCode.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Indicates, that the communication doesn't show an error.</td>
</tr>
</tbody>
</table>

Remarks

This class extends the AdsErrorCode by Succeeded(AdsErrorCode) and Failed(AdsErrorCode) methods.

Reference

TwinCAT.Ads Namespace

6.2.9.1 AdsErrorCodeExtensions Methods

The AdsErrorCodeExtensions type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed</td>
<td>Indicates, that the communication failed with AdsErrorCode.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Indicates, that the communication doesn't show an error.</td>
</tr>
</tbody>
</table>

Reference

AdsErrorCodeExtensions Class
6.2.9.1.1 AdsErrorCodeExtensions.Failed Method

Indicates, that the communication failed with AdsErrorCode [575].

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```c#
public static bool Failed(
    this AdsErrorCode errorCode
)
```

Parameters

errorCode Type: TwinCAT.Ads.AdsErrorCode [575]
The error code.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type AdsErrorCode [575]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AdsErrorCodeExtensions Class [581]
TwinCAT.Ads Namespace [151]

6.2.9.1.2 AdsErrorCodeExtensions.Succeeded Method

Indicates, that the communication / AdsErrorCode [575] doesn't show an error.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```c#
public static bool Succeeded(
    this AdsErrorCode errorCode
)
```
Parameters

errorCode

Type: TwinCAT.Ads.AdsErrorCode [575]
The error code.

Return Value

Type: Boolean
true if the AdsErrorCode [575] is NoError [575], false otherwise.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type AdsErrorCode [575]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AdsErrorCodeExtensions Class [581]
TwinCAT.Ads Namespace [151]

6.2.10 AdsErrorException Class

The exception that is thrown when an ADS error occurs.

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [57]
      TwinCAT.Ads.AdsErrorException
        TwinCAT.Ads.AdsSumCommandException [632]

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
[SerializableAttribute]
public class AdsErrorException : AdsException

The AdsErrorException type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![创办] AdsErrorException [585]</td>
<td>Initializes a new instance of the AdsErrorException class.</td>
</tr>
<tr>
<td>![创办] AdsErrorException(SerializationInfo, StreamingContext) [586]</td>
<td>Initializes a new instance of the AdsErrorException class.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsErrorException(String, AdsErrorCode)</td>
<td>Initializes a new Instance of the AdsErrorException class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the error code of the Exception.</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create(AdsErrorCode)</td>
<td>Creates the AdsErrorException</td>
</tr>
<tr>
<td>Create(String, AdsErrorCode)</td>
<td>Creates the AdsErrorException</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
|ToString | Creates and returns a string representation of the current exception.  
(Inherited from Exception.) |

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
|SerializeObjectState | Occurs when an exception is serialized to create an exception state object 
that contains serialized data about the exception. (Inherited from 
Exception.) |

#### Reference

**TwinCAT.Ads Namespace** [151]

### 6.2.10.1 AdsErrorException Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsErrorException()</td>
<td>Initializes a new Instance of the AdsErrorException class.</td>
</tr>
<tr>
<td>AdsErrorException(SerializationInfo,</td>
<td>Initializes a new instance of the AdsErrorException class.</td>
</tr>
<tr>
<td>StreamingContext)</td>
<td></td>
</tr>
<tr>
<td>AdsErrorException(String, AdsErrorCode)</td>
<td>Initializes a new Instance of the AdsErrorException class.</td>
</tr>
</tbody>
</table>

#### Reference

**AdsErrorException Class** [583]

**TwinCAT.Ads Namespace** [151]

### 6.2.10.1.1 AdsErrorException Constructor

Initializes a new Instance of the AdsErrorException class.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AdsErrorException()
```

#### Reference

**AdsErrorException Class** [583]
6.2.10.1.2 AdsErrorException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the AdsErrorException class.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected AdsErrorException(
    SerializationInfo info,
    StreamingContext streamingContext
)
```

**Parameters**

  The information.
  The streaming context.

**Reference**

- AdsErrorException Class
- AdsErrorException Overload
- TwinCAT.Ads Namespace

6.2.10.1.3 AdsErrorException Constructor (String, AdsErrorCode)

Initializes a new instance of the AdsErrorException class.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AdsErrorException(
    string message,
    AdsErrorCode errorCode
)
```

**Parameters**

- `message`: Type: System.String
  The message.
- `errorCode`: Type: TwinCAT.Ads.AdsErrorCode
  The error code.
6.2.10.2 AdsErrorException Properties

The `AdsErrorException` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the error code of the Exception.</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

Reference

`AdsErrorException Class [583]`
`AdsErrorException Overload [585]`
`TwinCAT.Ads Namespace [151]`

### AdsErrorException.ErrorCode Property

Gets the error code of the Exception.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```
C#
public AdsErrorCode ErrorCode { get; }
```
Property Value

Type: AdsErrorCode

The error code.

Reference

AdsErrorException Class

TwinCAT.Ads Namespace

6.2.10.3 AdsErrorException Methods

The AdsErrorException type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="589" alt="Create" /></td>
<td>Creates the AdsErrorException</td>
</tr>
<tr>
<td><img src="589" alt="Create" /></td>
<td>Creates the AdsErrorException</td>
</tr>
<tr>
<td><img src="589" alt="Equals" /></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td><img src="589" alt="Finalize" /></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><img src="589" alt="GetBaseException" /></td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td><img src="590" alt="GetHashCode" /></td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td><img src="590" alt="GetObjectData" /></td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)</td>
</tr>
<tr>
<td><img src="590" alt="GetType" /></td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td><img src="590" alt="MemberwiseClone" /></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><img src="590" alt="ToString" /></td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

AdsErrorException Class

TwinCAT.Ads Namespace
6.2.10.3.1 AdsErrorException.Create Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create(AdsErrorCode)</td>
<td>Creates the AdsErrorException</td>
</tr>
<tr>
<td>Create(String, AdsErrorCode)</td>
<td>Creates the AdsErrorException</td>
</tr>
</tbody>
</table>

Reference

AdsErrorException Class [583]
TwinCAT.Ads Namespace [151]

AdsErrorException.Create Method (AdsErrorCode)

Creates the AdsErrorException

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static AdsErrorException Create(
    AdsErrorCode adsErrorCode
)
```

Parameters

adsErrorCode Type: TwinCAT.Ads.AdsErrorCode [575]
The ads error code.

Return Value

Type: AdsErrorException [583]
AdsErrorException.

Reference

AdsErrorException Class [583]
Create Overload [589]
TwinCAT.Ads Namespace [151]

AdsErrorException.Create Method (String, AdsErrorCode)

Creates the AdsErrorException
TwinCAT.Ads Namespaces

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static AdsErrorException Create(
    string message,
    AdsErrorCode adsErrorCode
)
```

Parameters

- **message**
  - Type: System.String
  - The message.

- **adsErrorCode**
  - Type: TwinCAT.Ads.AdsErrorCode
  - The ads error code.

Return Value

Type: AdsErrorException
AdsErrorException.

Reference

- AdsErrorException Class
- Create Overload
- TwinCAT.Ads Namespace

6.2.10.3.2 AdsErrorException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

Parameters

- **info**
  - Type: System.Runtime.Serialization.SerializationInfo
  - The SerializationInfo that holds the serialized object data about the exception being thrown.

- **context**
  - Type: System.Runtime.Serialization.StreamingContext
  - The StreamingContext that contains contextual information about the source or destination.
## TwinsCAT.Ads Namespaces

### 6.2.10.4 AdsErrorException Events

The `AdsErrorException` type exposes the following members.

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object</td>
</tr>
<tr>
<td></td>
<td>that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### 6.2.11 AdsInvalidNotificationException Class

This `AdsInvalidNotificationException` is created if the length of the notification data is 0. This indicates that the notification handle is not valid any more. This exception is passed to the `AdsNotificationErrorEvent`.

#### Inheritance Hierarchy

- System.Object
  - System.Exception
    - `TwinCAT.AdsException` ![57](57)
      - `TwinCAT.Ads.AdsInvalidNotificationException`

#### Syntax

C#  

```csharp
[SerializableAttribute]
public sealed class AdsInvalidNotificationException : AdsException
```

The `AdsInvalidNotificationException` type exposes the following members.
TwinCAT.Ads Namespaces

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Handle</td>
<td>Handle of the notification.</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Gets the Time stamp as long</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]

6.2.11.1 AdsInvalidNotificationException Properties

The AdsInvalidNotificationException [591] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Handle [593]</td>
<td>Handle of the notification.</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TimeStamp [593]</td>
<td>Gets the Time stamp as long</td>
</tr>
</tbody>
</table>

**Reference**

AdsInvalidNotificationException Class [591]

TwinCAT.Ads Namespace [151]

### 6.2.11.1.1 AdsInvalidNotificationException.Handle Property

Handle of the notification.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public uint Handle { get; }
```

**Property Value**

Type: UInt32

The handle.

**Reference**

AdsInvalidNotificationException Class [591]

TwinCAT.Ads Namespace [151]

### 6.2.11.1.2 AdsInvalidNotificationException.TimeStamp Property

Gets the Time stamp as long
TwinCAT.Ads Namespaces

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public DateTimeOffset TimeStamp { get; }
```

Property Value

Type: DateTimeOffset
The time stamp.

Reference

AdsInvalidNotificationException Class [591]
TwinCAT.Ads Namespace [151]

6.2.11.2 AdsInvalidNotificationException Methods

The AdsInvalidNotificationException [591] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).) [594]</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

AdsInvalidNotificationException Class [591]
TwinCAT.Ads Namespace [151]

6.2.11.2.1 AdsInvalidNotificationException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context)
```

Parameters

- `info` Type: `System.Runtime.Serialization.SerializationInfo`
  The `SerializationInfo` that holds the serialized object data about the exception being thrown.

- `context` Type: `System.Runtime.Serialization.StreamingContext`
  The `StreamingContext` that contains contextual information about the source or destination.

Implements

`ISerializable.GetObjectData(SerializationInfo, StreamingContext)`
`Exception.GetObjectData(SerializationInfo, StreamingContext)`

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>info</td>
</tr>
</tbody>
</table>

Reference

`AdsInvalidNotificationException Class` [591]

`TwinCAT.Ads Namespace` [151]

6.2.12 AdsNotificationErrorEventArgs Class

Arguments for the `AdsNotificationError` [867] events.

Inheritance Hierarchy

- `System.Object`
- `System.EventArgs`
- `TwinCAT.Ads.AdsNotificationErrorEventArgs`

Namespace: `TwinCAT.Ads` [151]

Assembly: `TwinCAT.Ads.Abstractions` (in `TwinCAT.Ads.Abstractions.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd6ca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class AdsNotificationErrorEventArgs : EventArgs
```

The `AdsNotificationErrorEventArgs` type exposes the following members.
### TwinCAT.Ads Namespaces

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception [596]</td>
<td>Exception that was caught while handling notifications.</td>
</tr>
</tbody>
</table>

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

#### Reference

TwinCAT.Ads Namespace [151]

### 6.2.12.1 AdsNotificationErrorEventArgs Properties

The AdsNotificationErrorEventArgs [595] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception [596]</td>
<td>Exception that was caught while handling notifications.</td>
</tr>
</tbody>
</table>

#### Reference

AdsNotificationErrorEventArgs Class [595]

TwinCAT.Ads Namespace [151]

### 6.2.12.1.1 AdsNotificationErrorEventArgs.Exception Property

Exception that was caught while handling notifications.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

C#  
```csharp
public Exception Exception { get; }
```

#### Property Value

**Type:** Exception
6.2.12.2  AdsNotificationErrorEventArgs Methods

The AdsNotificationErrorEventArgs type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

AdsNotificationErrorEventArgs Class [595]

TwinCAT.Ads Namespace [151]

6.2.13  AdsNotificationEventArgs Class

Event argument class for AdsNotification [866] events.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.AdsNotificationEventArgs
      TwinCAT.Ads.AdsNotificationExEventArgs [601]
      TwinCAT.Ads.ValueNotificationEventArgs.T. [1053]

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class AdsNotificationEventArgs : EventArgs, INotification
```

The AdsNotificationEventArgs type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Memory object holding the Notification Data/Value.</td>
</tr>
<tr>
<td>Handle</td>
<td>Gets the Notification handle.</td>
</tr>
</tbody>
</table>

Reference

AdsNotificationEventArgs Class [595]

TwinCAT.Ads Namespace [151]
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeStamp [600]</td>
<td>Gets the time stamp of this Notification as DateTimeOffset.</td>
</tr>
<tr>
<td>UserData [600]</td>
<td>Gets the user object. This object is passed by to AddDeviceNotification and can be used to store data.</td>
</tr>
</tbody>
</table>

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

#### Reference

TwinCAT.Ads Namespace [151]

### 6.2.13.1 AdsNotificationEventArgs Properties

The AdsNotificationEventArgs [597] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data [598]</td>
<td>Memory object holding the Notification Data/Value.</td>
</tr>
<tr>
<td>Handle [599]</td>
<td>Gets the Notification handle.</td>
</tr>
<tr>
<td>TimeStamp [600]</td>
<td>Gets the time stamp of this Notification as DateTimeOffset.</td>
</tr>
<tr>
<td>UserData [600]</td>
<td>Gets the user object. This object is passed by to AddDeviceNotification and can be used to store data.</td>
</tr>
</tbody>
</table>

#### Reference

AdsNotificationEventArgs Class [597]

TwinCAT.Ads Namespace [151]

#### 6.2.13.1.1 AdsNotificationEventArgs.Data Property

Memory object holding the Notification Data/Value.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
### Syntax

**C#**

```csharp
public ReadOnlyMemory Data { get; }
```

### Property Value

**Type:** `ReadOnlyMemory`

**Implements**

`INotification.Data` [970]

### Remarks

This Memory object can be seen as binary 'View' to the value object. It represents exactly the data that corresponds to the `Handle` [599].

### Reference

`AdsNotificationEventArgs Class` [597]

`TwinCAT.Ads Namespace` [151]

### 6.2.13.1.2 AdsNotificationEventArgs.Handle Property

**Gets the Notification handle.**

**Namespace:** `TwinCAT.Ads` [151]

**Assembly:** `TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`

**Syntax**

**C#**

```csharp
public uint Handle { get; }
```

### Property Value

**Type:** `UInt32`

**Implements**

`INotification.Handle` [970]

### Remarks

The Notification Handle is the handle that is created during ADS Notification registration (`AddDeviceNotificationAsync Overload` [847], `AddDeviceNotification Overload` [844]) and used for deregistration (`DeleteDeviceNotificationAsync(UInt32, CancellationToken)` [858], `DeleteDeviceNotificationAsync(UInt32, CancellationToken)` [858]).

### Reference

`AdsNotificationEventArgs Class` [597]
6.2.13.1.3 AdsNotificationEventArgs.TimeStamp Property

Gets the time stamp of this Notification as DateTimeOffset.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public DateTimeOffset TimeStamp { get; }
```

**Property Value**

Type: DateTimeOffset

**Implements**

INotification.TimeStamp

**Reference**

AdsNotificationEventArgs Class

TwinCAT.Ads Namespace

---

6.2.13.1.4 AdsNotificationEventArgs.UserData Property

Gets the user object. This object is passed by to AddDeviceNotification and can be used to store data.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Object UserData { get; }
```

**Property Value**

Type: Object

**Implements**

INotification.UserData

**Reference**

AdsNotificationEventArgs Class

TwinCAT.Ads Namespace
6.2.13.2 AdsNotificationEventArgs Methods

The AdsNotificationEventArgs [597] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

AdsNotificationEventArgs Class [597]
TwinCAT.Ads Namespace [151]

6.2.14 AdsNotificationExEventArgs Class

Arguments for AdsNotificationEx [868] events.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.AdsNotificationEventArgs [597]
    TwinCAT.Ads.AdsNotificationExEventArgs

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdc3e72bc0ea15da1c14

Syntax

C#

public sealed class AdsNotificationExEventArgs : AdsNotificationEventArgs

The AdsNotificationExEventArgs type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Memory object holding the Notification Data/Value. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>AdsNotificationEventArgs [597].)</td>
</tr>
<tr>
<td>Handle</td>
<td>Gets the Notification handle. (Inherited from AdsNotificationEventArgs</td>
</tr>
<tr>
<td></td>
<td>[597].)</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Gets the time stamp of this Notification as DateTimeOffset. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>AdsNotificationEventArgs [597].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserData</td>
<td>Gets the user object. This object is passed by to AddDeviceNotification and can be used to store data. (Inherited from AdsNotificationEventArgs)</td>
</tr>
<tr>
<td>Value</td>
<td>Value of the ADS Notification.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads Namespace [151]

### 6.2.14.1 AdsNotificationExEventArgs Properties

The AdsNotificationExEventArgs [601] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Memory object holding the Notification Data/Value. (Inherited from AdsNotificationEventArgs)</td>
</tr>
<tr>
<td>Handle</td>
<td>Gets the Notification handle. (Inherited from AdsNotificationEventArgs)</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Gets the time stamp of this Notification as DateTimeOffset. (Inherited from AdsNotificationEventArgs)</td>
</tr>
<tr>
<td>UserData</td>
<td>Gets the user object. This object is passed by to AddDeviceNotification and can be used to store data. (Inherited from AdsNotificationEventArgs)</td>
</tr>
<tr>
<td>Value</td>
<td>Value of the ADS Notification.</td>
</tr>
</tbody>
</table>

### Reference

AdsNotificationExEventArgs Class [601]

TwinCAT.Ads Namespace [151]

### 6.2.14.1.1 AdsNotificationExEventArgs.Value Property

Value of the ADS Notification.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public Object Value { get; }
```

Property Value

Type: Object

Reference

AdsNotificationExEventArgs Class [601]
TwinCAT.Ads Namespace [151]

6.2.14.2 AdsNotificationExEventArgs Methods

The AdsNotificationExEventArgs [601] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>toString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

AdsNotificationExEventArgs Class [601]
TwinCAT.Ads Namespace [151]

6.2.15 AdsSession Class

AdsSession class

Inheritance Hierarchy

System.Object
  TwinCAT.Session [101]
    TwinCAT.Ads AdsSessionBase [614]
      TwinCAT.Ads.AdsSession

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class AdsSession : AdsSessionBase
```

The AdsSession type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsSession(AmsAddress) [609]</td>
<td>Initializes a new instance of the AdsSessionBase [614] class.</td>
</tr>
<tr>
<td>AdsSession(AmsAddress, SessionSettings) [610]</td>
<td>Initializes a new instance of the AdsSessionBase [614] class.</td>
</tr>
<tr>
<td>AdsSession(AmsNetId, Int32) [610]</td>
<td>Initializes a new instance of the AdsSessionBase [614] class.</td>
</tr>
<tr>
<td>AdsSession(AmsAddress, SessionSettings, Object) [611]</td>
<td>Initializes a new instance of the AdsSessionBase [614] class.</td>
</tr>
<tr>
<td>AdsSession(AmsNetId, Int32, SessionSettings) [611]</td>
<td>Initializes a new instance of the AdsSessionBase [614] class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address [618]</td>
<td>Gets the target address of the AdsSessionBase [614] (Inherited from AdsSessionBase [614].)</td>
</tr>
<tr>
<td>AddressSpecifier [103]</td>
<td>Gets the communication endpoint address string representation. (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Connection [618]</td>
<td>Gets the connection. (Inherited from AdsSessionBase [614].)</td>
</tr>
<tr>
<td>ConnectionState [104]</td>
<td>Gets the current Connection state of the Session [101] (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Disposed [105]</td>
<td>Gets a value indicating whether this Session [101] is disposed. (Inherited from Session [101].)</td>
</tr>
<tr>
<td>EstablishedAt [106]</td>
<td>Gets the UTC time when the session was established. (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Id [106]</td>
<td>Gets the Session Identifier (Inherited from Session [101].)</td>
</tr>
<tr>
<td>IsConnected [107]</td>
<td>Gets a value indicating whether this instance is connected. (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Name [107]</td>
<td>Gets the name of the session (Inherited from Session [101].)</td>
</tr>
<tr>
<td>NetId [619]</td>
<td>Gets the NetId of the Session (Inherited from AdsSessionBase [614].)</td>
</tr>
<tr>
<td>Owner [619]</td>
<td>Gets the Session owner. (Inherited from AdsSessionBase [614].)</td>
</tr>
<tr>
<td>Port [620]</td>
<td>Gets the Ams Port of the Session (Inherited from AdsSessionBase [614].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

**Name** | **Description**
--- | ---
Settings [620] | Gets the settings of the connection. (Inherited from AdsSessionBase[614].)
SymbolServer [108] | Gets the symbol server. (Inherited from Session[101].)

### Methods

**Name** | **Description**
--- | ---
Close [109] | Closes this ISession [88] (Inherited from Session[101].)
Connect [110] | Connects the session. (Inherited from Session[101].)
Disconnect [111] | Disconnects the session from the target. (Inherited from Session[101].)
Dispose. [112] | Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from Session[101].)
Dispose(Boolean) [622] | Releases unmanaged and - optionally - managed resources. (Inherited from AdsSessionBase[614].)
Equals | Determines whether the specified object is equal to the current object. (Inherited from Object.)
GetHashCode | Serves as the default hash function. (Inherited from Object.)
GetSessionName [623] | Gets the name/string identifier of the session. (Inherited from AdsSessionBase[614].)
GetType | Gets the Type of the current instance. (Inherited from Object.)
MemberwiseClone | Creates a shallow copy of the current Object. (Inherited from Object.)
OnConnect [624] | Handler function connecting the Session. (Inherited from AdsSessionBase[614].)
OnCreateSymbolServer [624] | Handler function creating the symbol server object. (Inherited from AdsSessionBase[614].)
OnDisconnect [625] | Called when [disconnect]. (Inherited from AdsSessionBase[614].)
OnGetAddress [625] | Handler function getting the address of the session. (Inherited from AdsSessionBase[614].)
ToString | Returns a string that represents the current object. (Inherited from Object.)

### Events

**Name** | **Description**
--- | ---
ConnectionStateChanged [115] | Occurs when connection status of the IConnectionStateProvider [84] has been changed. (Inherited from Session[101].)
Remarks

On top of the well known AdsClient [154] class that is used traditionally for ADS communication, the AdsSessionBase [614] class provides the following additionally abilities out of the box: These are used to provide more stable connections to ADS Servers than the AdsClient [154] can provide. The main issues are Resurrection / Self-Healing after communication timeouts, faster and less error prone reaction to communication errors (not necessarily waiting for communication timeouts) und enhanced communication diagnosis. These enhanced features are provided by the following additions to the TwinCAT.Ads API:

- AdsConnection [357] class.
- Enhanced diagnosis in form of communication statistics Statistics [621]
- (semi-automatic) Resurrectable client communication with AdsConnection [357] objects.
- Symbol caching SymbolServer [108]
- Fail fast handler for connection stabilization IFailFastHandler

The AdsConnection [357] is established by calling the Connect [110] method. The returned AdsConnection [357] can be used as long the AdsSessionBase [614] exists.

Examples

The following sample shows a simple use of the AdsSessionBase [614] object. The AdsSession object (and the dynamic SymbolLoader features) are only available from .NET 4 and upwards.

Use Session (async)

```csharp
using System;
using System.Diagnostics;
using System.Threading;
using TwinCAT;
using TwinCAT.Ads;
using TwinCAT.Ads.TypeSystem;
using TwinCAT.TypeSystem;
using TwinCAT.TypeSystem.Generic;
namespace Sample
{
    class SessionAsync
    {
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static async void Main(string[] args)
        {
            AmsAddress address = ArgParser.Parse(args);
            CancellationTokenSource cancelSource = new CancellationTokenSource();
            CancellationToken cancel = cancelSource.Token;
            SessionSettings settings = SessionSettings.Default; // Default settings are Async access with Timeout 5 sec
            // Async access is necessary for Console applications!
            using (AdsSession session = new AdsSession(address, settings))
            {
                AdsConnection connection = (AdsConnection) session.Connect(); // Establish the connection
                ConnectionStateChanged += Connection_ConnectionStateChanged;
                ConnectionState connectionState = connection.ConnectionState; // The actual connection state
                // Read the identification and version number of the device
                var result = await connection.ReadDeviceInfoAsync(cancel);
                result.ThrowOnError(); // Throws exception if failed.
                DeviceInfo deviceInfo = result.DeviceInfo;
                Version version = deviceInfo.Version(ConvertToStandard());
                Console.WriteLine(string.Format("DeviceName: {0}\r\nDeviceVersion: {1}\n", deviceInfo.Name, version.ToString(3)));
            }
        }
    }
}```
/// Read the state of the device
var resultReadState = await connection.ReadStateAsync(cancel);
resultReadState.ThrowOnError();

StateInfo stateInfo = resultReadState.State;
AdsState adsState = stateInfo.AdsState;
short deviceState = stateInfo.DeviceState;

Console.WriteLine(string.Format("DeviceState: {0}", deviceState));
Console.WriteLine(string.Format("AdsState : {0}", adsState));

// Other ADS methods (as formerly used on AdsClient) can be used also on connection object:
// connection.ReadAsync(...)
// connection.WriteAsync(...)
// connection.AddDeviceNotificationEx += ...

// Session communication Diagnostic:

int resurrectionTries = connection.ResurrectingTries;
int succeededResurrections = connection.Resurrections;

AdsCommunicationStatistics statistics = session.Statistics; // The communication statistics

// Symbol access:
// The Session holds and Caches the Symbolic data information
var resultDataTypes = await session.SymbolServer.GetDataTypesAsync(cancel);
var resultSymbols = await session.SymbolServer.GetSymbolsAsync(cancel);

if (resultDataTypes.Succeeded && resultSymbols.Succeeded) // Check for succeed
{
    IDataTypeCollection<IDataType> types = resultDataTypes.DataTypes;
    ISymbolCollection<ISymbol> symbols = resultSymbols.Symbols;

    Symbol projectNameSymbol = (Symbol)symbols["TwinCAT_SystemInfoVarList._AppInfo.ProjectName"];  
    var resultReadProjectName = await projectNameSymbol.ReadValueAsync(cancel);
    string projectName = (string)resultReadProjectName.Value;

    // Or use dynamic objects
    dynamic appInfo = symbols["TwinCAT_SystemInfoVarList._AppInfo"];  
    string projectName2 = appInfo.ProjectName; // Property dynamically created (synchronous call)!
    }

Console.WriteLine("" );
Console.WriteLine("Press [Enter] for leave:" );
Console.ReadLine();
}  
}  

private static void Connection_ConnectionStateChanged(object sender, ConnectionStateChangedEventArgs e) { Console.WriteLine("Connection State changed (NewState: {0}, OldState: {1}" , e.NewState , e.OldState); }  

Use Session (sync)

using System;
using System.Diagnostics;
using System.Threading;
using TwinCAT;
using TwinCAT.Ads;
using TwinCAT.Ads.TypeSystem;
using TwinCAT.TypeSystem;
using TwinCAT.TypeSystem.Generic;

namespace Sample
{
    class Session
    {
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static void Main(string[] args) {  

AmsAddress address = ArgParser.Parse(args);
SessionSettings settings = SessionSettings.Default; // Default settings are Async access with Timeout 5 sec

// Async access is necessary for Console applications!
using (AdsSession session = new AdsSession(address, settings))
{
    AdsConnection connection = (AdsConnection) session.Connect(); // Establish the connection
    ConnectionState connectionState = connection.ConnectionState; // The actual connection state

    // Read the identification and version number of the device
    DeviceInfo deviceInfo = connection.ReadDeviceInfo();
    Console.WriteLine(string.Format("DeviceName: {0}", deviceInfo.Name));
    Console.WriteLine(string.Format("DeviceVersion: {0}", version.ToString(3)));

    // Read the state of the device
    StateInfo stateInfo = connection.ReadState();
    AdsState adsState = stateInfo.AdsState;
    short deviceState = stateInfo.DeviceState;
    Console.WriteLine(string.Format("DeviceState: {0}", deviceState));
    Console.WriteLine(string.Format("AdsState: {0}", adsState));

    // Other ADS methods (as formerly used on AdsClient) can be used also on connection object:
    // connection.Read(...)  
    // connection.Write(...)  
    // connection.AddDeviceNotificationEx += ...

    // Session communication Diagnostic:
    int resurrectionTries = connection.ResurrectingTries;
    int succeededResurrections = connection.Resurrections;
    AdsCommunicationStatistics statistics = session.Statistics; // The communication statistics

    // Symbol access:
    // The Session holds and Caches the Symbolic data information
    IDataTypeCollection<IDataType> types = session.SymbolServer.DataTypes;
    ISymbolCollection<ISymbol> symbols = session.SymbolServer.Symbols;
    Symbol projectNameSymbol = (Symbol) symbols["TwinCAT_SystemInfoVarList._AppInfo.ProjectName"];  
    string projectName = (string) projectNameSymbol.ReadValue();

    // Or use dynamic objects
    dynamic appInfo = symbols["TwinCAT_SystemInfoVarList._AppInfo"];  
    string projectName2 = appInfo.ProjectName;
    Console.WriteLine("\n");
    Console.WriteLine("Press [Enter] for leave:");
    Console.ReadLine();
}

private static void Connection_ConnectionStateChanged(object sender, ConnectionStateChangedEventArgs e)
{
    Console.WriteLine("Connection State changed (NewState: {0}, OldState: {1})", e.NewState, e.OldState);
}

Reference

TwinCAT.Ads Namespace [►151]
TwinCAT.Session [►101]
TwinCAT.Ads.IAdsSession [►908]
IInterceptionFactory
### 6.2.15.1 AdsSession Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsSession(AmsAddress)</td>
<td>Initializes a new instance of the AdsSessionBase [614] class.</td>
</tr>
<tr>
<td>AdsSession(AmsAddress, SessionSettings)</td>
<td>Initializes a new instance of the AdsSessionBase [614] class.</td>
</tr>
<tr>
<td>AdsSession(AmsAddress, SessionSettings, Object)</td>
<td>Initializes a new instance of the AdsSessionBase [614] class.</td>
</tr>
</tbody>
</table>

#### Reference

- AdsSession Class [603]
- TwinCAT.Ads Namespace [151]

### 6.2.15.1.1 AdsSession Constructor (AmsAddress)

Initializes a new instance of the AdsSessionBase [614] class.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public AdsSession(
    AmsAddress address
)
```

**Parameters**

- **address**  
  Type: TwinCAT.Ads.AmsAddress [648]  
  The address.
6.2.15.1.2 AdsSession Constructor (AmsAddress, SessionSettings)

Initializes a new instance of the AdsSessionBase [614] class.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public AdsSession(
    AmsAddress address,
    SessionSettings settings
)
```

**Parameters**

- `address`  
  Type: TwinCAT.Ads.AmsAddress [648]  
  The address.

- `settings`  
  Type: TwinCAT.Ads.SessionSettings [1035]  
  The settings.

Reference

AdsSession Class [603]
AdsSession Overload [609]
TwinCAT.Ads Namespace [151]

6.2.15.1.3 AdsSession Constructor (AmsNetId, Int32)

Initializes a new instance of the AdsSessionBase [614] class.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public AdsSession(
    AmsNetId netId,
    int port
)
```
Parameters

netId
Type: TwinCAT.Ads.AmsNetId [665]
The net identifier.

port
Type: System.Int32
The port.

Reference

AdsSession Class [603]
AdsSession Overload [609]
TwinCAT.Ads Namespace [151]

6.2.15.1.4 AdsSession Constructor (AmsAddress, SessionSettings, Object)

Initializes a new instance of the AdsSessionBase [614] class.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public AdsSession(
    AmsAddress address,
    SessionSettings settings,
    Object owner
)

Parameters

address
Type: TwinCAT.Ads.AmsAddress [648]
The address.

settings
Type: TwinCAT.Ads.SessionSettings [1035]
The settings.

owner
Type: System.Object
The session owner

Reference

AdsSession Class [603]
AdsSession Overload [609]
TwinCAT.Ads Namespace [151]

6.2.15.1.5 AdsSession Constructor (AmsNetId, Int32, SessionSettings)

Initializes a new instance of the AdsSessionBase [614] class.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public AdsSession(
    AmsNetId netId,
    int port,
    SessionSettings settings)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>port</td>
<td>System.Int32</td>
<td>The port.</td>
</tr>
</tbody>
</table>

Reference

AdsSession Class [603]

AdsSession Overload [609]

TwinCAT.Ads Namespace [151]

6.2.15.2 AdsSession Properties

The AdsSession [603] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Gets the target address of the AdsSessionBase [614] (Inherited from AdsSessionBase [614].)</td>
</tr>
<tr>
<td>AddressSpecifier</td>
<td>Gets the communication endpoint address string representation. (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the connection. (Inherited from AdsSessionBase [614].)</td>
</tr>
<tr>
<td>ConnectionState</td>
<td>Gets the current Connection state of the Session [101] (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Disposed</td>
<td>Gets a value indicating whether this Session [101] is disposed. (Inherited from Session [101].)</td>
</tr>
<tr>
<td>EstablishedAt</td>
<td>Gets the UTC time when the session was established. (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the Session Identifier (Inherited from Session [101].)</td>
</tr>
<tr>
<td>IsConnected</td>
<td>Gets a value indicating whether this instance is connected. (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the session (Inherited from Session [101].)</td>
</tr>
<tr>
<td>NetId</td>
<td>Gets the NetId of the Session (Inherited from AdsSessionBase [614].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner [619]</td>
<td>Gets the Session owner. (Inherited from AdsSessionBase [614].)</td>
</tr>
<tr>
<td>Port [620]</td>
<td>Gets the Ams Port of the Session (Inherited from AdsSessionBase [614].)</td>
</tr>
<tr>
<td>Settings [620]</td>
<td>Gets the settings of the connection. (Inherited from AdsSessionBase [614].)</td>
</tr>
<tr>
<td>SymbolServer [108]</td>
<td>Gets the symbol server. (Inherited from Session [101].)</td>
</tr>
</tbody>
</table>

**Reference**

AdsSession Class [603]
TwinCAT.Ads Namespace [151]

### 6.2.15.3 AdsSession Methods

The AdsSession [603] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close [109]</td>
<td>Closes this ISession [88] (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Connect [110]</td>
<td>Connects the session. (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Disconnect [111]</td>
<td>Disconnects the session from the target. (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Dispose [112]</td>
<td>Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Dispose(Boolean) [622]</td>
<td>Releases unmanaged and - optionally - managed resources. (Inherited from AdsSessionBase [614].)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetSessionName [623]</td>
<td>Gets the name/string identifier of the session. (Inherited from AdsSessionBase [614].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>OnConnect [624]</td>
<td>Handler function connecting the Session. (Inherited from AdsSessionBase [614].)</td>
</tr>
<tr>
<td>OnCreateSymbolSever [624]</td>
<td>Handler function creating the symbol server object. (Inherited from AdsSessionBase [614].)</td>
</tr>
<tr>
<td>On Disconnect [625]</td>
<td>Called when [disconnect]. (Inherited from AdsSessionBase [614].)</td>
</tr>
</tbody>
</table>
### 6.2.15.4 AdsSession Events

The `AdsSession` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ConnectionStateChanged</strong></td>
<td>Occurs when connection status of the <code>IConnectionStateProvider</code> has been changed. (Inherited from <code>Session</code>.)</td>
</tr>
</tbody>
</table>

### 6.2.16 AdsSessionBase Class

Abstract base class for ADS Sessions.

**Inheritance Hierarchy**

- `System.Object`
  - TwinCAT.Session [101]
    - TwinCAT.Ads.AdsSessionBase
      - TwinCAT.Ads.AdsSession [603]

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
public abstract class AdsSessionBase : Session, IAdsSession, ISession, IConnectionStateProvider, ISymbolServerProvider
```

The `AdsSessionBase` type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsSessionBase</td>
<td>Initializes a new instance of the AdsSessionBase class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Gets the target address of the AdsSessionBase</td>
</tr>
<tr>
<td>AddressSpecifier</td>
<td>Gets the communication endpoint address string representation. (Inherited from Session.)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the connection.</td>
</tr>
<tr>
<td>ConnectionState</td>
<td>Gets the current Connection state of the Session (Inherited from Session.)</td>
</tr>
<tr>
<td>Disposed</td>
<td>Gets a value indicating whether this Session is disposed. (Inherited from Session.)</td>
</tr>
<tr>
<td>EstablishedAt</td>
<td>Gets the UTC time when the session was established. (Inherited from Session.)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the Session Identifier (Inherited from Session.)</td>
</tr>
<tr>
<td>IsConnected</td>
<td>Gets a value indicating whether this instance is connected. (Inherited from Session.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the session (Inherited from Session.)</td>
</tr>
<tr>
<td>NetId</td>
<td>Gets the NetId of the Session</td>
</tr>
<tr>
<td>Owner</td>
<td>Gets the Session owner.</td>
</tr>
<tr>
<td>Port</td>
<td>Gets the Ams Port of the Session</td>
</tr>
<tr>
<td>Settings</td>
<td>Gets the settings of the connection.</td>
</tr>
<tr>
<td>Statistics</td>
<td>Gets the Communication / Session statistics.</td>
</tr>
<tr>
<td>SymbolServer</td>
<td>Gets the symbol server. (Inherited from Session.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Closes this ISession (Inherited from Session.)</td>
</tr>
<tr>
<td>Connect</td>
<td>Connects the session. (Inherited from Session.)</td>
</tr>
<tr>
<td>Disconnect</td>
<td>Disconnects the session from the target. (Inherited from Session.)</td>
</tr>
<tr>
<td>Dispose</td>
<td>Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from Session.)</td>
</tr>
<tr>
<td>Dispose(Boolean)</td>
<td>Releases unmanaged and - optionally - managed resources. (Overrides Session.Dispose(Boolean).)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize [623]</td>
<td>Finalizes an instance of the AdsSessionBase class. (Overrides</td>
</tr>
<tr>
<td></td>
<td>Object.Finalize.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetSessionName [623]</td>
<td>Gets the name/string identifier of the session. (Overrides</td>
</tr>
<tr>
<td></td>
<td>Session.GetSessionName [112].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>OnConnect [624]</td>
<td>Handler function connecting the Session. (Overrides</td>
</tr>
<tr>
<td></td>
<td>Session.OnConnect(Boolean) [113].)</td>
</tr>
<tr>
<td>OnCreateSymbolServer</td>
<td>Handler function creating the symbol server object. (Overrides</td>
</tr>
<tr>
<td></td>
<td>Session.OnCreateSymbolServer [113].)</td>
</tr>
<tr>
<td>OnDisconnect [625]</td>
<td>Called when [disconnect]. (Overrides Session.OnDisconnect [114].)</td>
</tr>
<tr>
<td>OnGetAddress [625]</td>
<td>Handler function getting the address of the session. (Overrides</td>
</tr>
<tr>
<td></td>
<td>Session.OnGetAddress [114].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionStateChanged [115]</td>
<td>Occurs when connection status of the IConnectionStateProvider [84] has been changed. (Inherited from Session [101].)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads Namespace [151]

#### 6.2.16.1 AdsSessionBase Constructor

Initializes a new instance of the AdsSessionBase [614] class.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
protected AdsSessionBase(
    AmsAddress address,
    SessionSettings settings,
    IAdsClientFactory factory,
    Object owner
)
```
### Parameters

**address**  
Type: `TwinCAT.Ads.AmsAddress`  
The address.

**settings**  
Type: `TwinCAT.Ads.SessionSettings`  
The settings.

**factory**  
Type: `IAdsClientFactory`  
The client factory

**owner**  
Type: `System.Object`  
The session owner

### Reference

**AdsSessionBase Class**  
`614`

**TwinCAT.Ads Namespace**  
`151`

### 6.2.16.2 AdsSessionBase Properties

The `AdsSessionBase` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address <code>[618]</code></td>
<td>Gets the target address of the <code>AdsSessionBase</code> <code>[614]</code></td>
</tr>
<tr>
<td>AddressSpecifier</td>
<td>Gets the communication endpoint address string representation. (Inherited from <code>Session</code> <code>[101]</code>.)</td>
</tr>
<tr>
<td>Connection <code>[618]</code></td>
<td>Gets the connection.</td>
</tr>
<tr>
<td>ConnectionState</td>
<td>Gets the current Connection state of the <code>Session</code> <code>[101]</code> (Inherited from <code>Session</code> <code>[101]</code>.)</td>
</tr>
<tr>
<td>Disposed <code>[105]</code></td>
<td>Gets a value indicating whether this <code>Session</code> <code>[101]</code> is disposed. (Inherited from <code>Session</code> <code>[101]</code>.)</td>
</tr>
<tr>
<td>EstablishedAt</td>
<td>Gets the UTC time when the session was established. (Inherited from <code>Session</code> <code>[101]</code>.)</td>
</tr>
<tr>
<td>Id <code>[106]</code></td>
<td>Gets the Session Identifier (Inherited from <code>Session</code> <code>[101]</code>.)</td>
</tr>
<tr>
<td>IsConnected <code>[107]</code></td>
<td>Gets a value indicating whether this instance is connected. (Inherited from <code>Session</code> <code>[101]</code>.)</td>
</tr>
<tr>
<td>Name <code>[107]</code></td>
<td>Gets the name of the session (Inherited from <code>Session</code> <code>[101]</code>.)</td>
</tr>
<tr>
<td>NetId <code>[619]</code></td>
<td>Gets the NetId of the Session</td>
</tr>
<tr>
<td>Owner <code>[619]</code></td>
<td>Gets the Session owner.</td>
</tr>
<tr>
<td>Port <code>[620]</code></td>
<td>Gets the Ams Port of the Session</td>
</tr>
<tr>
<td>Settings <code>[620]</code></td>
<td>Gets the settings of the connection.</td>
</tr>
</tbody>
</table>
### SymbolServer

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolServer</td>
<td>Gets the symbol server. (Inherited from Session [101].)</td>
</tr>
</tbody>
</table>

**Reference**

AdsSessionBase Class [614]

TwinCAT.Ads Namespace [151]

#### 6.2.16.2.1 AdsSessionBase.Address Property

Gets the target address of the AdsSessionBase [614]

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdDca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AmsAddress Address { get; }
```

**Property Value**

Type: AmsAddress [648]

The address.

**Implements**

IAdsSession.Address [910]

**Reference**

AdsSessionBase Class [614]

TwinCAT.Ads Namespace [151]

#### 6.2.16.2.2 AdsSessionBase.Connection Property

Gets the connection.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdDca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AdsConnection Connection { get; protected set; }
```

**Property Value**

Type: AdsConnection [357]

The connection.
6.2.16.2.3  AdsSessionBase.NetId Property

Gets the NetId of the Session

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public AmsNetId NetId { get; }
```

**Property Value**

Type: **AmsNetId**

The net identifier.

**Implements**

IAdsSession.NetId

**Reference**

AdsSessionBase Class [614]

TwinCAT.Ads Namespace [151]

6.2.16.2.4  AdsSessionBase.Owner Property

Gets the Session owner.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public Object Owner { get; }
```

**Property Value**

Type: **Object**

The owner or NULL

**Implements**

IAdsSession.Owner

**Reference**

AdsSessionBase Class [614]

TwinCAT.Ads Namespace [151]
6.2.16.2.5 AdsSessionBase.Port Property

Gets the Ams Port of the Session

**Namespace:** TwinCAT.Ads
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public int Port { get; }
```

**Property Value**

Type: `Int32`
The port.

**Implements**

IAdsSession.Port[911]

6.2.16.2.6 AdsSessionBase.Settings Property

Gets the settings of the connection.

**Namespace:** TwinCAT.Ads
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public SessionSettings Settings { get; }
```

**Property Value**

Type: `SessionSettings`[1035]
The settings.
6.2.16.2.7 AdsSessionBase.Statistics Property

 Gets the Communication / Session statistics.

 Namespace: TwinCAT.Ads [151]
 Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

 Syntax

 C#

 public AdsCommunicationStatistics Statistics { get; }

 Property Value

 Type: AdsCommunicationStatistics [350]
 The communication / Session statistics.

 Reference

 AdsSessionBase Class [614]
 TwinCAT.Ads Namespace [151]

 6.2.16.3 AdsSessionBase Methods

 The AdsSessionBase [614] type exposes the following members.

 Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Closes this ISession [88] (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Connect</td>
<td>Connects the session. (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Disconnect</td>
<td>Disconnects the session from the target. (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Dispose.</td>
<td>Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from Session [101].)</td>
</tr>
<tr>
<td>Dispose(Boolean)</td>
<td>Releases unmanaged and - optionally - managed resources. (Overrides Session.Dispose(Boolean) [112].)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Finalizes an instance of the AdsSessionBase [614] class. (Overrides Object.Finalize.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetSessionName</td>
<td>Gets the name/string identifier of the session. (Overides Session.GetSessionName [112].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>OnConnect</td>
<td>Handler function connecting the Session. (Overides Session.OnConnect(Boolean) [113].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OnCreateSymbolServer</td>
<td>Handler function creating the symbol server object. (Overrides Session.OnCreateSymbolServer.)</td>
</tr>
<tr>
<td>OnDisconnect</td>
<td>Called when [disconnect]. (Overrides Session.OnDisconnect.)</td>
</tr>
<tr>
<td>OnGetAddress</td>
<td>Handler function getting the address of the session. (Overrides Session.OnGetAddress.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Reference**

AdsSessionBase Class ![614]

TwinCAT.Ads Namespace ![151]

#### 6.2.16.3.1 AdsSessionBase.Dispose Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from Session.)</td>
</tr>
<tr>
<td>Dispose(Boolean)</td>
<td>Releases unmanaged and - optionally - managed resources. (Overrides Session.Dispose(Boolean).)</td>
</tr>
</tbody>
</table>

**Reference**

AdsSessionBase Class ![614]

TwinCAT.Ads Namespace ![151]

**AdsSessionBase.Dispose Method (Boolean)**

Releases unmanaged and - optionally - managed resources.

**Namespace:** TwinCAT.Ads ![151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected override void Dispose(
    bool disposing
)
```

**Parameters**

- **disposing**
  - Type: System.Boolean
  - true to release both managed and unmanaged resources; false to release only unmanaged resources.
6.2.16.3.2 AdsSessionBase.Finalize Method

Finalizes an instance of the AdsSessionBase class.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
protected override void Finalize()

Implements

Object.Finalize.

Reference

AdsSessionBase Class [614]

TwinCAT.Ads Namespace [151]

6.2.16.3.3 AdsSessionBase.GetSessionName Method

Gets the name/string identifier of the session.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
protected override string GetSessionName()

**Return Value**

Type: String  
System.String.

Reference

AdsSessionBase Class [614]

TwinCAT.Ads Namespace [151]
### 6.2.16.3.4 AdsSessionBase.OnConnect Method

Handler function connecting the Session.

**Namespace:** TwinCAT.Ads  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
protected override IConnection OnConnect(
    bool reconnect
)
```

**Parameters**

- **reconnect**  
  Type: System.Boolean

**Return Value**

Type: IConnection

IConnection.

**Reference**

- AdsSessionBase Class  
- TwinCAT.Ads Namespace

### 6.2.16.3.5 AdsSessionBase.OnCreateSymbolServer Method

Handler function creating the symbol server object.

**Namespace:** TwinCAT.Ads  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
protected override ISymbolServer OnCreateSymbolServer()
```

**Return Value**

Type: ISymbolServer

ISymbolServer.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SessionNotConnectedException</td>
<td>The connection is not established!</td>
</tr>
</tbody>
</table>
6.2.16.3.6  AdsSessionBase.OnDisconnect Method

Called when [disconnect].

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
protected override bool OnDisconnect()
```

**Return Value**

Type: Boolean
true if XXXX, false otherwise.

**Reference**

AdsSessionBase Class [614]

TwinCAT.Ads Namespace [151]

6.2.16.3.7  AdsSessionBase.OnGetAddress Method

Handler function getting the address of the session.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
protected override string OnGetAddress()
```

**Return Value**

Type: String
System.String.

**Reference**

AdsSessionBase Class [614]

TwinCAT.Ads Namespace [151]

6.2.16.4  AdsSessionBase Events

The AdsSessionBase [614] type exposes the following members.
Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionStateChanged</td>
<td>Occurs when connection status of the IConnectionStateProvider has been changed. (Inherited from Session.)</td>
</tr>
</tbody>
</table>

Reference

AdsSessionBase Class

TwinCAT.Ads Namespace

6.2.17 AdsState Enumeration

Describes the AdsState.

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public enum AdsState
```

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid</td>
<td>0</td>
<td>Ads State is Invalid / Uninitialized</td>
</tr>
<tr>
<td>Idle</td>
<td>1</td>
<td>Idle</td>
</tr>
<tr>
<td>Reset</td>
<td>2</td>
<td>Reset</td>
</tr>
<tr>
<td>Init</td>
<td>3</td>
<td>Initialize</td>
</tr>
<tr>
<td>Start</td>
<td>4</td>
<td>Start</td>
</tr>
<tr>
<td>Run</td>
<td>5</td>
<td>Run</td>
</tr>
<tr>
<td>Stop</td>
<td>6</td>
<td>Stop</td>
</tr>
<tr>
<td>SaveConfig</td>
<td>7</td>
<td>Save Configuration</td>
</tr>
<tr>
<td>LoadConfig</td>
<td>8</td>
<td>Load Configuration</td>
</tr>
<tr>
<td>PowerFailure</td>
<td>9</td>
<td>Power failure</td>
</tr>
<tr>
<td>PowerGood</td>
<td>10</td>
<td>Power Good</td>
</tr>
<tr>
<td>Error</td>
<td>11</td>
<td>Error</td>
</tr>
<tr>
<td>Shutdown</td>
<td>12</td>
<td>Shutdown</td>
</tr>
<tr>
<td>Suspend</td>
<td>13</td>
<td>Suspend</td>
</tr>
<tr>
<td>Resume</td>
<td>14</td>
<td>Resume</td>
</tr>
<tr>
<td>Config</td>
<td>15</td>
<td>Config (System is in config mode)</td>
</tr>
<tr>
<td>Reconfig</td>
<td>16</td>
<td>Reconfig (System should restart in config mode)</td>
</tr>
<tr>
<td>Stopping</td>
<td>17</td>
<td>Stopping</td>
</tr>
<tr>
<td>Incompatible</td>
<td>18</td>
<td>Incompatible</td>
</tr>
<tr>
<td>Exception</td>
<td>19</td>
<td>Exception</td>
</tr>
</tbody>
</table>
6.2.18 AdsStateChangedEventArgs Class

Arguments for the AdsStateChanged [934] event.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.AdsStateChangedEventArgs

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public sealed class AdsStateChangedEventArgs : EventArgs

The AdsStateChangedEventArgs type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsStateChangedEventArgs</td>
<td>Initializes a new instance of the AdsStateChangedEventArgs class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>Current state of the ADS device.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]

6.2.18.1 AdsStateChangedEventArgs Constructor

Initializes a new instance of the AdsStateChangedEventArgs class.
**Syntax**

C#

```csharp
public AdsStateChangedEventArgs(
    StateInfo value
)
```

**Parameters**

value  
Type: TwinCAT.Ads.StateInfo  
Current state of the ADS device.

**Reference**

AdsStateChangedEventArgs Class [627]

TwinCAT.Ads Namespace [151]

### 6.2.18.2 AdsStateChangedEventArgs Properties

The AdsStateChangedEventArgs [627] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>Current state of the ADS device.</td>
</tr>
</tbody>
</table>

**Reference**

AdsStateChangedEventArgs Class [627]

TwinCAT.Ads Namespace [151]

### 6.2.18.2.1 AdsStateChangedEventArgs.State Property

Current state of the ADS device.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public StateInfo State { get; }
```

**Property Value**

Type: StateInfo [1041]  
The state.
6.2.18.3 AdsStateChangedEventArgs Methods

The AdsStateChangedEventArgs type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

AdsStateChangedEventArgs Class [627]
TwinCAT.Ads Namespace [151]

6.2.19 AdsStateChangedEventArgs2 Class

Event Arguments for AdsStateChanged events.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.AdsStateChangedEventArgs2

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public sealed class AdsStateChangedEventArgs2 : EventArgs

The AdsStateChangedEventArgs2 type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Gets the connection.</td>
</tr>
<tr>
<td>NewState</td>
<td>The new state</td>
</tr>
<tr>
<td>OldState</td>
<td>The old state</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Session</td>
<td>The session</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

### Reference

- TwinCAT.Ads Namespace [630]
- System.EventArgs

#### 6.2.19.1 AdsStateChangedEventArgs2 Properties

The `AdsStateChangedEventArgs2` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Gets the connection.</td>
</tr>
<tr>
<td>NewState</td>
<td>The new state</td>
</tr>
<tr>
<td>OldState</td>
<td>The old state</td>
</tr>
<tr>
<td>Session</td>
<td>The session</td>
</tr>
</tbody>
</table>

### Reference

- AdsStateChangedEventArgs2 Class [629]
- TwinCAT.Ads Namespace [630]

#### 6.2.19.1.1 AdsStateChangedEventArgs2.Connection Property

Gets the connection.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public IConnection Connection { get; }
```
**Property Value**

Type: `IConnection` [74]

The connection.

**Reference**

`AdsStateChangedEventArgs2` Class [629]

**TwinCAT.Ads Namespace** [151]

### 6.2.19.1.2 `AdsStateChangedEventArgs2.NewState` Property

The new state

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public StateInfo NewState { get; }
```

**Property Value**

Type: `StateInfo` [1041]

**Reference**

`AdsStateChangedEventArgs2` Class [629]

**TwinCAT.Ads Namespace** [151]

### 6.2.19.1.3 `AdsStateChangedEventArgs2.OldState` Property

The old state

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public StateInfo OldState { get; }
```

**Property Value**

Type: `StateInfo` [1041]

**Reference**

`AdsStateChangedEventArgs2` Class [629]

**TwinCAT.Ads Namespace** [151]
6.2.19.1.4 AdsStateChangedEventArgs2.Session Property

The session

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ISession Session { get; }
```

**Property Value**

**Type:** ISession [88]

**Reference**

AdsStateChangedEventArgs2 Class [629]

TwinCAT.Ads Namespace [151]

6.2.19.2 AdsStateChangedEventArgs2 Methods

The AdsStateChangedEventArgs2 [629] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Reference**

AdsStateChangedEventArgs2 Class [629]

TwinCAT.Ads Namespace [151]

6.2.20 AdsSumCommandException Class

The exception that is thrown when an ADS SumCommandBase error occurs.

**Inheritance Hierarchy**

System.Object

  System.Exception

    TwinCAT.AdsException [57]

      TwinCAT.Ads.AdsErrorException [583]

      TwinCAT.Ads.AdsSumCommandException
Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
[SerializableAttribute]
public sealed class AdsSumCommandException : AdsErrorException
```

The AdsSumCommandException type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsSumCommandException</td>
<td>Initializes a new Instance of the AdsErrorException class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the error code of the Exception. (Inherited from AdsErrorException.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>SumCommand</td>
<td>Gets the sum command.</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides AdsErrorException.GetObjectData(SerializationInfo, StreamingContext) [590].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads Namespace [151]

### 6.2.20.1 AdsSumCommandException Constructor

Initializes a new Instance of the AdsErrorException class.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public AdsSumCommandException(
    string message,
    ISumCommand command
)
```

**Parameters**

| message       | Type: System.String  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>The message.</td>
<td></td>
</tr>
</tbody>
</table>
| command       | Type: TwinCAT.Ads.SumCommand.ISumCommand [1206]  
| The command.  |                      |

**Reference**

AdsSumCommandException Class [632]  
TwinCAT.Ads Namespace [151]

### 6.2.20.2 AdsSumCommandException Properties

The AdsSumCommandException [632] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the error code of the Exception. (Inherited from AdsErrorException [583].)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
### AdsSumCommandException.SumCommand Property

Gets the sum command.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b436095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public ISumCommand SumCommand { get; }
```

**Property Value**

Type: ISumCommand

The sum command.

**Reference**

AdsSumCommandException Class [632]

TwinCAT.Ads Namespace [151]

### 6.2.20.3 AdsSumCommandException Methods

The AdsSumCommandException [632] type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
</tbody>
</table>

(Inherited from Object.)
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides AdsErrorException.GetObjectData(SerializationInfo, StreamingContext) [636])</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

AdsSumCommandException Class [632]

TwinCAT.Ads Namespace [151]

6.2.20.3.1 AdsSumCommandException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

Parameters

- **info**
  - Type: System.Runtime.Serialization.SerializationInfo
  - The SerializationInfo that holds the serialized object data about the exception being thrown.

- **context**
  - Type: System.Runtime.Serialization.StreamingContext
  - The StreamingContext that contains contextual information about the source or destination.

Implements

- ISerializable.GetObjectData(SerializationInfo, StreamingContext)
- Exception.GetObjectData(SerializationInfo, StreamingContext)

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>info</td>
</tr>
</tbody>
</table>
6.2.21 AdsSymbolVersionChangedEventArgs Class

Arguments for the AdsSymbolVersionChanged [637] event.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.AdsSymbolVersionChangedEventArgs

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public sealed class AdsSymbolVersionChangedEventArgs : EventArgs
```

The AdsSymbolVersionChangedEventArgs type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolVersion</td>
<td>Current symbol version device.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]

6.2.21.1 AdsSymbolVersionChangedEventArgs Properties

The AdsSymbolVersionChangedEventArgs [637] type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolVersion</td>
<td>Current symbol version device.</td>
</tr>
</tbody>
</table>

Reference

AdsSymbolVersionChangedEventArgs Class [637]
TwinCAT.Ads Namespace [151]

6.2.21.1 AdsSymbolVersionChangedEventArgs.SymbolVersion Property
Current symbol version device.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public short SymbolVersion { get; }

Property Value
Type: Int16
The symbol version.

Reference

AdsSymbolVersionChangedEventArgs Class [637]
TwinCAT.Ads Namespace [151]

6.2.21.2 AdsSymbolVersionChangedEventArgs Methods
The AdsSymbolVersionChangedEventArgs [637] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
</tbody>
</table>

Reference

AdsSymbolVersionChangedEventArgs Class [637]
TwinCAT.Ads Namespace [151]
6.2.22 AdsTransMode Enumeration

ADS Transmission Mode for ADS Notifications.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public enum AdsTransMode
```

**Members**

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>None / Uninitialized transport mode. No AdsNotification event is fired.</td>
</tr>
</tbody>
</table>
| ClientCycle      | 1     | Client triggered cyclic AdsNotification event. The AdsNotification event is fired cyclically triggered from the client side. Polling is used from the User Application to read values, before they are fired as Notifications. Client side triggering has the following consequences:  
  - The realtime environment on the server side will be less stressed (especially the mailbox queue).  
  - Value requests are serialized one after another and are handled slower (synchronously, not asynchronously)  
  - Implicit synchronization of the events into the UI Thread. |
| ClientOnChange   | 2     | The AdsNotification event is fired when data changes triggered by the client. The AdsNotification event is fired on-change triggered from the client side. Polling is used from the User Application to read values, before they are fired as Notifications. Client side triggering has the following consequences:  
  - The realtime environment on the server side will be less stressed (especially the mailbox queue).  
  - Value requests are serialized one after another and are handled slower (synchronously, not asynchronously)  
  - Implicit synchronization of the events into the UI Thread. |
<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic</td>
<td>3</td>
<td>The <code>AdsNotification</code> [866] event is fired cyclically. The Notification will be registered on the ADS Server side for a cyclical trigger (dependant on time parameter) and is bound to the 'default' task of the addressed target. In case of the PLC target (e.g. Port 851) the default task is the first configured task. Each time the 'default' task has finished its cycle the realtime system will check for the expired cycle time and sends the <code>AdsNotification</code> [866] message on expiry. The used ContextMask for the 'default' task is 0. <strong>Please be aware, that server side 'Change' notifications stress the realtime system and should be handled with care. Therefore, dependent of the cycle time of the task and the capabilities of the system only a limited set of Cyclic Notifications should be used!</strong> A system limit for server side notification registrations is 1024.</td>
</tr>
<tr>
<td>OnChange</td>
<td>4</td>
<td>On-Change <code>AdsNotification</code> [866] event. The Notification will be registered on the ADS Server side for an on-change and optional cyclical trigger (dependant on parameters) and is bound to the 'default' task of the addressed target. In case of the PLC target (e.g. Port 851) the default task is the first configured task. Each time this task has finished its cycle the realtime system will check for the changed value and an optional expired cycle time and sends the <code>AdsNotification</code> [866] message on change or expiry. The used ContextMask for the 'default' task is 0. <strong>Please be aware, that server side 'OnChange' notifications stress the realtime system / the default task with value comparisons. Therefore, dependent of the cycle time of the task and the capabilities of the system a higher amount of notification registrations should be handled with care!</strong> A system limit for server side notification registrations is 1024.</td>
</tr>
<tr>
<td>Member name</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CyclicInContext</td>
<td>5</td>
<td>The AdsNotification[866] event is fired cyclically within the given task context.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A Value of parameter is interpreted as task context number ContextMask[1388]. This can be important, if the notifications have to be synchronous with specific tasks, but should not be used in the default case.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The Notification will be registered on the ADS Server side for a cyclical trigger (dependant on time parameter) and is bound to the task specified by the ContextMask of the addressed target. In case of the PLC target (e.g. Port 851) the ContextMask is the Index of the global TASKINFOARRAY - 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Each time this task has finished its cycle the realtime system will check for the expired cycle time and sends the AdsNotification[866] message on expiry.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Please be aware, that server side 'OnChange' notifications stress the realtime system / the default task with value comparisons. Therefore, dependent of the cycle time of the task and the capabilities of the system only a limited set of OnChange Notifications should be used!</td>
</tr>
<tr>
<td>OnChangeInContext</td>
<td>6</td>
<td>The AdsNotification[866] event is fired when the data changes within the given task context.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A Value of parameter is interpreted as task context number ContextMask[1388]. This can be important, if the notifications have to be synchronously with specific tasks, but should not be used in the default case.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The Notification will be registered on the ADS Server side for an on-change and optional cyclical trigger (dependant on parameters) and is bound to the task specified by the ContextMask of the addressed target. In case of the PLC target (e.g. Port 851) the ContextMask is the Index of the global TASKINFOARRAY - 1. Each time this task has finished its cycle the realtime system will check for the changed value and an optional expired cycle time and sends the AdsNotification[866] message on change or expiry.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Please be aware, that server side 'OnChange' notifications stress the realtime system / the default task with value comparisons. Therefore, dependent of the cycle time of the task and the capabilities of the system only a limited set of OnChange Notifications should be used!</td>
</tr>
</tbody>
</table>
Remarks

The AdsTransMode configures the registration of the AdsNotification [866] at the server system and how the parameters of the AddDeviceNotification(String, Int32, NotificationSettings, Object) [844] are interpreted. The following general scenarios are addressed:

- Cyclic notifications.
- Notifications on value change.
- Server side and Client side notifications.
- Binding of notifications to specific tasks.

In the default case the OnChange or the Cyclic (Server cycle) should be used. All other modes are side cases for special purposes.

More about the AdsNotifications: ADS Notification concept [22].

Reference

TwinCAT.Ads Namespace [151]

IAdsNotifications.AdsNotification [866]

IAdsNotifications.AdsNotificationEx [868]

AddDeviceNotification Overload [844]

AddDeviceNotificationEx Overload [850]

6.2.23 AdsVersion Class

The structure contains the version number, revision number and build number.

Inheritance Hierarchy

System.Object
   TwinCAT.Ads.AdsVersion

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class AdsVersion

The AdsVersion type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsVersion(Byte)</td>
<td>Initializes a new instance of the AdsVersion class.</td>
</tr>
</tbody>
</table>
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build</td>
<td>Gets or sets the build number.</td>
</tr>
<tr>
<td>Empty</td>
<td>Get the Empty/Uninitialized Version (0,0,0)</td>
</tr>
<tr>
<td>IsEmpty</td>
<td>Gets a value indicating whether this instance is empty / uninitialized.</td>
</tr>
<tr>
<td>Revision</td>
<td>Gets or sets the revision number.</td>
</tr>
<tr>
<td>Version</td>
<td>Gets or sets the version number.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]

6.2.23.1 AdsVersion Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsVersion(Byte)</td>
<td>Initializes a new instance of the AdsVersion class.</td>
</tr>
</tbody>
</table>

Reference

AdsVersion Class [642]

TwinCAT.Ads Namespace [151]
6.2.23.1.1 AdsVersion Constructor (.Byte.)

Initializes a new instance of the AdsVersion [642] class.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsVersion(
    byte[] bytes
)
```

Parameters

- **bytes**: Type: `System.Byte`. The bytes.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td><code>bytes</code></td>
</tr>
<tr>
<td>ArgumentException</td>
<td><code>bytes</code></td>
</tr>
</tbody>
</table>

Reference

- AdsVersion Class [642]
- AdsVersion Overload [643]
- TwinCAT.Ads Namespace [151]

6.2.23.1.2 AdsVersion Constructor (Int32, Int32, Int32)


Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsVersion(
    int version,
    int revision,
    int build
)
```

Parameters

- **version**: Type: `System.Int32`. The version.
- **revision**: Type: `System.Int32`. The revision.
build         Type: System.Int32
The build.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td>version or revision</td>
</tr>
</tbody>
</table>

Reference

AdsVersion Class [642]
AdsVersion Overload [643]
TwinCAT.Ads Namespace [151]

6.2.23.2 AdsVersion Properties

The AdsVersion type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build</td>
<td>Gets or sets the build number.</td>
</tr>
<tr>
<td>Empty</td>
<td>Get the Empty/Uninitialized Version (0,0,0)</td>
</tr>
<tr>
<td>IsEmpty</td>
<td>Gets a value indicating whether this instance is empty / uninitialized.</td>
</tr>
<tr>
<td>Revision</td>
<td>Gets or sets the revision number.</td>
</tr>
<tr>
<td>Version</td>
<td>Gets or sets the version number.</td>
</tr>
</tbody>
</table>

Reference

AdsVersion Class [642]
TwinCAT.Ads Namespace [151]

6.2.23.2.1 AdsVersion.Build Property

Gets or sets the build number.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int Build { get; set; }
```
Property Value
Type: Int32

Reference
AdsVersion Class [642]
TwinCAT.Ads Namespace [151]

6.2.23.2.2 AdsVersion.Empty Property
Get the Empty/Uninitialized Version (0,0,0)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
public static AdsVersion Empty { get; }

Property Value
Type: AdsVersion [642]
The empty.

Reference
AdsVersion Class [642]
TwinCAT.Ads Namespace [151]

6.2.23.2.3 AdsVersion.IsEmpty Property
Gets a value indicating whether this instance is empty / uninitialized.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
public bool IsEmpty { get; }

Property Value
Type: Boolean
true if this instance is empty; otherwise, false.

Reference
AdsVersion Class [642]
TwinCAT.Ads Namespace [151]
6.2.23.2.4 **AdsVersion.Revision Property**

Gets or sets the revision number.

**Namespace:** [TwinCAT.Ads](#)  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public byte Revision { get; set; }
```

**Property Value**

Type: Byte

**Reference**

AdsVersion Class  
TwinCAT.Ads Namespace

6.2.23.2.5 **AdsVersion.Version Property**

Gets or sets the version number.

**Namespace:** [TwinCAT.Ads](#)  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public byte Version { get; set; }
```

**Property Value**

Type: Byte

**Reference**

AdsVersion Class  
TwinCAT.Ads Namespace

6.2.23.3 **AdsVersion Methods**

The AdsVersion type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

AdsVersion Class [642]

TwinCAT.Ads Namespace [151]

6.2.23.3.1 AdsVersion.ConvertToStandard Method


Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Version ConvertToStandard()
```

Return Value

Type: Version

Version.

Reference

AdsVersion Class [642]

TwinCAT.Ads Namespace [151]

6.2.24 AmsAddress Class

Ams/Ads Address

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.AmsAddress

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public class AmsAddress
```

The `AmsAddress` type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmsAddress</td>
<td>Protected constructor</td>
</tr>
<tr>
<td>AmsAddress(Int32)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsAddress(String)</td>
<td>Initializes a new instance of the AmsAddress class.</td>
</tr>
<tr>
<td>AmsAddress(AmsAddress)</td>
<td>Copy constructor</td>
</tr>
<tr>
<td>AmsAddress(AmsPort)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsAddress(Byte, Int32)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsAddress(Byte, AmsPort)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsAddress(String, Int32)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsAddress(String, AmsPort)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsAddress(AmsNetId, Int32)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsAddress(AmsNetId, AmsPort)</td>
<td>Constructor</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets an Empty Address.</td>
</tr>
<tr>
<td>NetId</td>
<td>Gets the NetId</td>
</tr>
<tr>
<td>Port</td>
<td>Gets the Port number</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clone</td>
<td>Clones this instance.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [659]</td>
<td>Equals (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse [660]</td>
<td>Parses a string to an AmsAddress object.</td>
</tr>
<tr>
<td>SetNetId [661]</td>
<td>Sets the net identifier.</td>
</tr>
<tr>
<td>SetPort [662]</td>
<td>Sets the port.</td>
</tr>
<tr>
<td>ToString [662]</td>
<td>Converts the Address to String 'NetId:Port' (Overrides Object.ToString.)</td>
</tr>
<tr>
<td>TryParse [662]</td>
<td>Tries to parse the AmsAddress from string.</td>
</tr>
</tbody>
</table>

### Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equality [663]</td>
<td>Operator==</td>
</tr>
<tr>
<td>Inequality [664]</td>
<td>Implements the != operator.</td>
</tr>
</tbody>
</table>

### Remarks

The AmsAddress consists of NetId [657] and Port [658] information and once it is constructed is immutable.

### Reference

TwinCAT.Ads Namespace [151]

### 6.2.24.1 AmsAddress Constructor

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmsAddress [651]</td>
<td>Protected constructor</td>
</tr>
<tr>
<td>AmsAddress(Int32) [651]</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsAddress(String) [652]</td>
<td>Initializes a new instance of the AmsAddress [648] class.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>AmsAddress(AmsAddress)</td>
<td>Copy constructor</td>
</tr>
<tr>
<td>AmsAddress(AmsPort)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsAddress(Byte, Int32)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsAddress(Byte, AmsPort)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsAddress(String, Int32)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsAddress(String, AmsPort)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsAddress(AmsNetId, Int32)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsAddress(AmsNetId, AmsPort)</td>
<td>Constructor</td>
</tr>
</tbody>
</table>

**Reference**

AmsAddress Class [*648*]

TwinCAT.Ads Namespace [*151*]

### 6.2.24.1.1 AmsAddress Constructor

Protected constructor

**Namespace:** TwinCAT.Ads [*151*]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
protected AmsAddress()
```

**Reference**

AmsAddress Class [*648*]

AmsAddress Overload [*650*]

TwinCAT.Ads Namespace [*151*]

### 6.2.24.1.2 AmsAddress Constructor (Int32)

Constructor
Namespace:  TwinCAT.Ads [151]
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public AmsAddress(
    int port
)

Parameters

port  Type:  System.Int32
The port.

Reference

AmsAddress Class [648]
AmsAddress Overload [650]
TwinCAT.Ads Namespace [151]

6.2.24.1.3  AmsAddress Constructor (String)

Initializes a new instance of the AmsAddress [648] class.

Namespace:  TwinCAT.Ads [151]
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public AmsAddress(
    string str
)

Parameters

str  Type:  System.String
The address coded as string (Format NetId:Port, 1.2.3.4.5:Port)

Reference

AmsAddress Class [648]
AmsAddress Overload [650]
TwinCAT.Ads Namespace [151]

6.2.24.1.4  AmsAddress Constructor (AmsAddress)

Copy constructor
**Namespace:** TwinCAT.Ads [➤ 151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public AmsAddress(
    AmsAddress address
)
```

**Parameters**

- `address`  
  *Type:* TwinCAT.Ads.AmsAddress [➤ 648]  
  The address.

### Reference

- AmsAddress Class [➤ 648]  
- AmsAddress Overload [➤ 650]  
- TwinCAT.Ads Namespace [➤ 151]

#### 6.2.24.1.5 AmsAddress Constructor (AmsPort)

**Constructor**

**Namespace:** TwinCAT.Ads [➤ 151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AmsAddress(
    AmsPort port
)
```

**Parameters**

- `port`  
  *Type:* TwinCAT.Ads.AmsPort [➤ 693]  
  The port.

### Reference

- AmsAddress Class [➤ 648]  
- AmsAddress Overload [➤ 650]  
- TwinCAT.Ads Namespace [➤ 151]

#### 6.2.24.1.6 AmsAddress Constructor (.Byte., Int32)

**Constructor**
**Namespace:** TwinCAT.Ads [† 151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AmsAddress(
    byte[] netId,
    int port
)
```

**Parameters**

- **netId**
  - Type: `System.Byte`
  - The net identifier.

- **port**
  - Type: `System.Int32`
  - The port.

**Reference**

- AmsAddress Class [† 648]
- AmsAddress Overload [† 650]
- TwinCAT.Ads Namespace [† 151]

### 6.2.24.1.7 AmsAddress Constructor (.Byte., AmsPort)

**Constructor**

**Namespace:** TwinCAT.Ads [† 151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AmsAddress(
    byte[] netId,
    AmsPort port
)
```

**Parameters**

- **netId**
  - Type: `System.Byte`
  - The net identifier.

- **port**
  - Type: `TwinCAT.Ads.AmsPort` [† 693]
  - The port.

**Reference**

- AmsAddress Class [† 648]
- AmsAddress Overload [† 650]
- TwinCAT.Ads Namespace [† 151]
6.2.24.1.8 AmsAddress Constructor (String, Int32)

Constructor

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AmsAddress(
    string netId,
    int port
)
```

Parameters

- **netId**
  - Type: System.String
  - The net identifier.

- **port**
  - Type: System.Int32
  - The port.

Reference

- AmsAddress Class
- AmsAddress Overload
- TwinCAT.Ads Namespace

6.2.24.1.9 AmsAddress Constructor (String, AmsPort)

Constructor

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AmsAddress(
    string netId,
    AmsPort port
)
```

Parameters

- **netId**
  - Type: System.String
  - The net identifier.

- **port**
  - Type: TwinCAT.Ads.AmsPort
  - The port.

Reference

- AmsAddress Class
- AmsAddress Overload
### 6.2.24.1.10 AmsAddress Constructor (AmsNetId, Int32)

Constructor

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AmsAddress(
    AmsNetId netId,
    int port
)
```

**Parameters**

- **netId**
  - Type: TwinCAT.Ads.AmsNetId
  - Net Id

- **port**
  - Type: System.Int32
  - Port

**Reference**

- AmsAddress Class [648]
- AmsAddress Overload [650]
- TwinCAT.Ads Namespace [151]

### 6.2.24.1.11 AmsAddress Constructor (AmsNetId, AmsPort)

Constructor

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AmsAddress(
    AmsNetId netId,
    AmsPort port
)
```

**Parameters**

- **netId**
  - Type: TwinCAT.Ads.AmsNetId
  - The net identifier.

- **port**
  - Type: TwinCAT.Ads.AmsPort
  - The port.
6.2.24.2 AmsAddress Properties

The *AmsAddress* [648] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets an Empty Address.</td>
</tr>
<tr>
<td>NetId</td>
<td>Gets the NetId</td>
</tr>
<tr>
<td>Port</td>
<td>Gets the Port number</td>
</tr>
</tbody>
</table>

6.2.24.2.1 AmsAddress.Empty Property

Gets an Empty Address.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public static AmsAddress Empty { get; }
```

**Property Value**

Type: *AmsAddress* [648]
The empty.

Reference

*AmsAddress Class* [648]
*TwinCAT.Ads Namespace* [151]

6.2.24.2.2 AmsAddress.NetId Property

Gets the NetId
**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public AmsNetId NetId { get; }
```

### Property Value

**Type:** AmsNetId

The net identifier.

**Reference**

AmsAddress Class [648]

TwinCAT.Ads Namespace [151]

### 6.2.24.2.3 AmsAddress.Port Property

Gets the Port number

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public int Port { get; }
```

### Property Value

**Type:** Int32

The port.

**Reference**

AmsAddress Class [648]

TwinCAT.Ads Namespace [151]

### 6.2.24.3 AmsAddress Methods

The AmsAddress class type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Clone][659]</td>
<td>Clones this instance.</td>
</tr>
<tr>
<td>![Equals][659]</td>
<td>Equals (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Overrides Object.GetHashCode.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse</td>
<td>Parses a string to an AmsAddress object.</td>
</tr>
<tr>
<td>SetNetId</td>
<td>Sets the net identifier.</td>
</tr>
<tr>
<td>SetPort</td>
<td>Sets the port.</td>
</tr>
<tr>
<td>ToString</td>
<td>Converts the Address to String 'NetId:Port' (Overrides Object.ToString.)</td>
</tr>
<tr>
<td>TryParse</td>
<td>Tries to parse the AmsAddress from string.</td>
</tr>
</tbody>
</table>

Reference

AmsAddress Class [648]
TwinCAT.Ads Namespace [151]

### 6.2.24.3.1 AmsAddress.Clone Method

Clones this instance.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AmsAddress Clone()
```

**Return Value**

Type: AmsAddress [648]
AmsAddress.

Reference

AmsAddress Class [648]
TwinCAT.Ads Namespace [151]

### 6.2.24.3.2 AmsAddress.Equals Method

Equals
**Equals Method**

```
public override bool Equals(
    Object obj)
```

**Parameters**

- **obj**
  - Type: `System.Object`
  - The object to compare with the current object.

**Return Value**

- Type: `Boolean`
  - `true` if the specified `Object` is equal to this instance; otherwise, `false`.

**Reference**

- `AmsAddress Class [648]`
- `TwinCAT.Ads Namespace [151]`

### 6.2.24.3.3 AmsAddress.GetHashCode Method

**Gets the HashCode of the Address**

```
public override int GetHashCode()
```

**Return Value**

- Type: `Int32`
  - A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

**Reference**

- `AmsAddress Class [648]`
- `TwinCAT.Ads Namespace [151]`

### 6.2.24.3.4 AmsAddress.Parse Method

**Parses a string to an AmsAddress object.**
**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public static AmsAddress Parse(
    string str
)
```

#### Parameters

- **str**
  - Type: `System.String`
  - The string.

#### Return Value

Type: `AmsAddress` [648]

### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormatException</td>
<td></td>
</tr>
</tbody>
</table>

#### Reference

- AmsAddress Class [648]
- TwinCAT.Ads Namespace [151]

### 6.2.24.3.5 AmsAddress.SetNetId Method

Sets the net identifier.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected void SetNetId(
    AmsNetId netId
)
```

#### Parameters

- **netId**
  - Type: `TwinCAT.Ads.AmsNetId` [665]
  - The net identifier.

#### Reference

- AmsAddress Class [648]
- TwinCAT.Ads Namespace [151]
6.2.24.3.6 AmsAddress.SetPort Method

Sets the port.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

protected void SetPort(
    int port
)

Parameters

port Type: System.Int32
The port.

Reference

AmsAddress Class
TwinCAT.Ads Namespace

6.2.24.3.7 AmsAddress.ToString Method

Converts the Address to String 'NetId:Port'

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public override string ToString()

Return Value

Type: String
A String that represents this instance.

Reference

AmsAddress Class
TwinCAT.Ads Namespace

6.2.24.3.8 AmsAddress.TryParse Method

Tries to parse the AmsAddress from string.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public static bool TryParse(
    string str,
    out AmsAddress address
)
```

**Parameters**

- **str**
  - Type: `System.String`
  - The STR.

- **address**
  - Type: `TwinCAT.Ads.AmsAddress` [648]
  - The address.

**Return Value**

Type: `Boolean`

true if XXXX, false otherwise.

**Reference**

AmsAddress Class [648]

TwinCAT.Ads Namespace [151]

### 6.2.24.4 AmsAddress Operators

The `AmsAddress` [648] type exposes the following members.

#### Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>==</code></td>
<td>Equality [663] Operator==</td>
</tr>
<tr>
<td><code>!=</code></td>
<td>Inequality [664] Implements the <code>!=</code> operator.</td>
</tr>
</tbody>
</table>

**Reference**

AmsAddress Class [648]

TwinCAT.Ads Namespace [151]

### 6.2.24.4.1 AmsAddress.Equality Operator

Operator==

**Namespace**: TwinCAT.Ads [151]

**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public static bool operator ==(
    AmsAddress o1,
    AmsAddress o2
)
```

Parameters

- **o1**
  Type: `TwinCAT.Ads.AmsAddress` [648]
  The o1.

- **o2**
  Type: `TwinCAT.Ads.AmsAddress` [648]
  The o2.

Return Value

Type: `Boolean`
The result of the operator.

Reference

- AmsAddress Class [648]
- TwinCAT.Ads Namespace [151]

### 6.2.24.4.2 AmsAddress.Inequality Operator

Implements the `!=` operator.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static bool operator !=(
    AmsAddress o1,
    AmsAddress o2
)
```

Parameters

- **o1**
  Type: `TwinCAT.Ads.AmsAddress` [648]
  The o1.

- **o2**
  Type: `TwinCAT.Ads.AmsAddress` [648]
  The o2.

Return Value

Type: `Boolean`
The result of the operator.

Reference

- AmsAddress Class [648]
- TwinCAT.Ads Namespace [151]
6.2.25  AmsNetId Class

AMS/ADS Net ID

Inheritance Hierarchy

System.Object
   TwinCAT.Ads.AmsNetId

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

[SerializableAttribute]
public class AmsNetId : IComparable<AmsNetId>, IComparable

The AmsNetId type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmsNetId(Byte.)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsNetId(ReadOnly Span)</td>
<td></td>
</tr>
<tr>
<td>AmsNetId(String)</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsNetId(AmsNetId)</td>
<td>Copy Constructor</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broadcast</td>
<td>Gets the broadcast address (255.255.255.255)</td>
</tr>
<tr>
<td>Empty</td>
<td>Creates an empty NetId (&quot;0.0.0.0.0&quot;)</td>
</tr>
<tr>
<td>IsBroadcast</td>
<td>Gets a value indicating whether this AmsNetId is the broadcast address (255.255.255.255)</td>
</tr>
<tr>
<td>IsEmpty</td>
<td>Gets a value indicating whether this instance is empty / Uninitialized (AmsNetId: 0.0.0.0.0.0)</td>
</tr>
<tr>
<td>IsLocal</td>
<td>Determines, whether the AmsNetId is local.</td>
</tr>
<tr>
<td>IsLoopback</td>
<td>Indicates, that this AmsNetId indicates a Loopback (ID: 127.0.0.1.1)</td>
</tr>
<tr>
<td>IsSubAddress</td>
<td>Gets a value indicating whether this AmsNetId indicates a SubNet.</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### Item [674]
- **Description**: Gets or sets the Byte with the specified index.

### Local [674]
- **Description**: Gets the Local Net ID (System service must be running)

### LocalHost [675]
- **Description**: Creates the local NetId ("127.0.0.1.1.1")

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AsSpan [676]</td>
<td>Returns the AmsNetId as byte Span.</td>
</tr>
<tr>
<td>Clone [677]</td>
<td>Clones the NetId</td>
</tr>
<tr>
<td>CompareTo(Object) [678]</td>
<td>Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.</td>
</tr>
<tr>
<td>CompareTo(AmsNetId) [678]</td>
<td>Compares the current object with another object of the same type.</td>
</tr>
<tr>
<td>Equals(Object) [679]</td>
<td>Equals (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Equals(AmsNetId, AmsNetId) [680]</td>
<td>Determines whether the specified AmsNetIds are equal.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>FromBinHexString [681]</td>
<td>Creates the AmsNetId from bin hex string.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IsSameTarget(AmsNetId, AmsNetId) [682]</td>
<td>Determines whether the AmsNetIds refer to the same target.</td>
</tr>
<tr>
<td>IsSameTarget(AmsNetId, AmsNetId, Boolean) [683]</td>
<td>Determines whether the AmsNetIds refer to the same target.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>NetIdsEqual(Byte.) [684]</td>
<td>Compares the netIds</td>
</tr>
<tr>
<td>NetIdsEqual(Byte, Byte) [685]</td>
<td>Compares the NetIds</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### AmsNetId Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmsNetId(Byte) [668]</td>
<td>Constructor</td>
</tr>
<tr>
<td>AmsNetId(ReadOnly Span) [668]</td>
<td></td>
</tr>
<tr>
<td>AmsNetId(String) [669]</td>
<td>Constructor</td>
</tr>
</tbody>
</table>

---

**Reference**

TwinCAT.Ads Namespace [151]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmsNetId(AmsNetId)[669]</td>
<td>Copy Constructor</td>
</tr>
</tbody>
</table>

References

AmsNetId Class [665]

TwinCAT.Ads Namespace [151]

6.2.25.1.1 AmsNetId Constructor (.Byte.)

Constructor

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15daa1c14

Syntax

C#
```csharp
public AmsNetId(
    byte[] netId
)
```

Parameters

netId Type: System.Byte
Net ID in bytes

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>Not a valid NetId:netId</td>
</tr>
</tbody>
</table>

Reference

AmsNetId Class [665]
AmsNetId Overload [667]
TwinCAT.Ads Namespace [151]

6.2.25.1.2 AmsNetId Constructor (ReadOnlySpan`1)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15daa1c14

Syntax

C#
```csharp
public AmsNetId(
    ReadOnlySpan data
)
```
Parameters
data 

Type: ReadOnlySpan

Reference
AmsNetId Class [665]
AmsNetId Overload [667]
TwinCAT.Ads Namespace [151]

6.2.25.1.3 AmsNetId Constructor (String)

Constructor

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public AmsNetId(
    string netId
)

Parameters
netId 

Type: System.String
NetId as string

Reference
AmsNetId Class [665]
AmsNetId Overload [667]
TwinCAT.Ads Namespace [151]

6.2.25.1.4 AmsNetId Constructor (AmsNetId)

Copy Constructor

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public AmsNetId(
    AmsNetId netId
)
Parameters

netId Type: TwinCAT.Ads.AmsNetId

Reference

AmsNetId Class [665]
AmsNetId Overload [667]
TwinCAT.Ads Namespace [151]

6.2.25.2 AmsNetId Properties

The AmsNetId [665] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broadcast [670]</td>
<td>Gets the broadcast address (255.255.255.255.255)</td>
</tr>
<tr>
<td>Empty [671]</td>
<td>Creates an empty NetId (&quot;0.0.0.0.0&quot;)</td>
</tr>
<tr>
<td>IsBroadcast [671]</td>
<td>Gets a value indicating whether this AmsNetId [665] is the broadcast address (255.255.255.255.255)</td>
</tr>
<tr>
<td>IsEmpty [672]</td>
<td>Gets a value indicating whether this instance is empty / Uninitialized (AmsNetId: 0.0.0.0.0.0)</td>
</tr>
<tr>
<td>IsLocal [672]</td>
<td>Determines, whether the AmsNetId [665] is local.</td>
</tr>
<tr>
<td>IsLoopback [673]</td>
<td>Indicates, that this AmsNetId [665] indicates a Loopback (ID: 127.0.0.1.1)</td>
</tr>
<tr>
<td>IsSubAddress [673]</td>
<td>Gets a value indicating whether this AmsNetId [665] indicates a SubNet.</td>
</tr>
<tr>
<td>Item [674]</td>
<td>Gets or sets the Byte with the specified index.</td>
</tr>
<tr>
<td>Local [674]</td>
<td>Gets the Local Net ID (System service must be running)</td>
</tr>
<tr>
<td>LocalHost [675]</td>
<td>Creates the local NetId (&quot;127.0.0.1.1&quot;)</td>
</tr>
</tbody>
</table>

Reference

AmsNetId Class [665]
TwinCAT.Ads Namespace [151]

6.2.25.2.1 AmsNetId.Broadcast Property

Gets the broadcast address (255.255.255.255.255)
**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public static AmsNetId Broadcast { get; }
```

### Property Value

**Type:** AmsNetId

The broadcast.

**Reference**

AmsNetId Class

TwinCAT.Ads Namespace

---

### 6.2.25.2.2 AmsNetId.Empty Property

Creates an empty NetId ("0.0.0.0.0.0")

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public static AmsNetId Empty { get; }
```

### Property Value

**Type:** AmsNetId

The empty.

**Reference**

AmsNetId Class

TwinCAT.Ads Namespace

---

### 6.2.25.2.3 AmsNetId.IsBroadcast Property

Gets a value indicating whether this AmsNetId is the broadcast address (255.255.255.255.255)

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public bool IsBroadcast { get; }
```
Property Value
Type: Boolean
true if this instance is broadcast; otherwise, false.

Reference
AmsNetId Class [665]
TwinCAT.Ads Namespace [151]

6.2.25.2.4 AmsNetId.IsEmpty Property
Gets a value indicating whether this instance is empty / Uninitialized (AmsNetId: 0.0.0.0.0.0)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
public bool IsEmpty { get; }

Property Value
Type: Boolean
true if this instance is empty; otherwise, false.

Reference
AmsNetId Class [665]
TwinCAT.Ads Namespace [151]

6.2.25.2.5 AmsNetId.IsLocal Property
Determines, whether the AmsNetId [665] is local.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
public bool IsLocal { get; }

Property Value
Type: Boolean
true if this instance is local; otherwise, false.

Remarks
Doesn't check for LocalHost [675].
6.2.25.2.6  AmsNetId.IsLoopback Property

Indicates, that this AmsNetId [665] indicates a Loopback (ID: 127.0.0.1.1)

**Namespace:**  TwinCAT.Ads [151]

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public bool IsLoopback { get; }
```

**Property Value**

Type:  Boolean  
true if this instance is loop back; otherwise, false.

**Remarks**

A Subnet is indicated, when the AmsNetId [665] doesn't end with X.X.X.X.1.1

**Reference**

AmsNetId Class [665]

TwinCAT.Ads Namespace [151]

6.2.25.2.7  AmsNetId.IsSubAddress Property

Gets a value indicating whether this AmsNetId [665] indicates a SubNet.

**Namespace:**  TwinCAT.Ads [151]

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public bool IsSubAddress { get; }
```

**Property Value**

Type:  Boolean  
true if this instance indicates a subnet; otherwise, false.
### 6.2.25.2.8 AmsNetId.Item Property

Gets or sets the `Byte` with the specified index.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```
public byte this[int i] { get; set; }
```

**Parameters**

- `i`  
  Type: `System.Int32`  
  The index.

**Return Value**

Type: `Byte`  
`System.Byte`.

**Reference**

- AmsNetId Class [665]
- TwinCAT.Ads Namespace [151]

### 6.2.25.2.9 AmsNetId.Local Property

Gets the Local Net ID (System service must be running)

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```
public static AmsNetId Local { get; }
```

**Property Value**

Type: `AmsNetId` [665]  
The local.

**Remarks**

The system service must be running

**Reference**

- AmsNetId Class [665]
- TwinCAT.Ads Namespace [151]
### 6.2.25.2.10 AmsNetId.LocalHost Property

Creates the local NetId ("127.0.0.1.1")

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

C#  
```csharp
public static AmsNetId LocalHost { get; }
```

#### Property Value

Type: AmsNetId

The local host.

#### Reference

AmsNetId Class

TwinCAT.Ads Namespace

---

### 6.2.25.3 AmsNetId Methods

The AmsNetId type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="hex" alt="AsSpan" /></td>
<td>Returns the AmsNetId as byte Span.</td>
</tr>
<tr>
<td><img src="hex" alt="Clone" /></td>
<td>Clones the NetId</td>
</tr>
<tr>
<td><img src="hex" alt="CompareTo(Object)" /></td>
<td>Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.</td>
</tr>
<tr>
<td><img src="hex" alt="CompareTo(AmsNetId)" /></td>
<td>Compares the current object with another object of the same type.</td>
</tr>
<tr>
<td><img src="hex" alt="Equals(Object)" /></td>
<td>equals (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td><img src="hex" alt="Equals(AmsNetId, AmsNetId)" /></td>
<td>Determines whether the specified AmsNetId is equal.</td>
</tr>
<tr>
<td><img src="hex" alt="Finalize" /></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><img src="hex" alt="FromBinHexString" /></td>
<td>Creates the AmsNetId from bin hex string.</td>
</tr>
<tr>
<td><img src="hex" alt="GetHashCode" /></td>
<td>Gets the HashCode of the Address (Overrides Object.GetHashCode.)</td>
</tr>
<tr>
<td><img src="hex" alt="GetType" /></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>IsSameTarget(AmsNetId, AmsNetId)</code></td>
<td>Determines whether the AmsNetId's refer to the same target.</td>
</tr>
<tr>
<td><code>IsSameTarget(AmsNetId, AmsNetId, Boolean)</code></td>
<td>Determines whether the AmsNetId's refer to the same target.</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>NetIdsEqual(.Byte.)</code></td>
<td>Compares the netIds</td>
</tr>
<tr>
<td><code>NetIdsEqual(.Byte., .Byte., Boolean)</code></td>
<td></td>
</tr>
<tr>
<td><code>Parse</code></td>
<td>Converts the string representation of the address to AmsNetId.</td>
</tr>
<tr>
<td><code>ToBinHex</code></td>
<td>Converts the AmsNetId to a BinHex string.</td>
</tr>
<tr>
<td><code>ToBinHex(AmsNetId)</code></td>
<td>Converts the specified AmsNetId to a BinHex string.</td>
</tr>
<tr>
<td><code>ToBytes</code></td>
<td>Converts the NetId object to byte array</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Converts the netId to string (Overrides Object.ToString.)</td>
</tr>
<tr>
<td><code>ToString(String, IFormatProvider)</code></td>
<td>Returns a String that represents this instance.</td>
</tr>
<tr>
<td><code>TryParse</code></td>
<td>Converts the string representation of the address to AmsNetId.</td>
</tr>
<tr>
<td><code>TryWriteBytes</code></td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

AmsNetId Class [665]

TwinCAT.Ads Namespace [151]

### 6.2.25.3.1 AmsNetId.AsSpan Method

Returns the AmsNetId as byte Span.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public ReadOnlySpan AsSpan()
```

Return Value

Type: `ReadOnlySpan`<br>`ReadOnlySpan<System.Byte>`.

Reference

AmsNetId Class [665]<br>TwinCAT.Ads Namespace [151]

6.2.25.3.2 AmsNetId.Clone Method

Clones the NetId

Namespace: TwinCAT.Ads [151]<br>Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AmsNetId Clone()
```

Return Value

Type: `AmsNetId` [665]<br>The cloned `AmsNetId` [665]

Reference

AmsNetId Class [665]<br>TwinCAT.Ads Namespace [151]

6.2.25.3.3 AmsNetId.CompareTo Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompareTo(Object)</td>
<td></td>
</tr>
</tbody>
</table>
[678] | Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object. |

| CompareTo(AmsNetId) |
[678] | Compares the current object with another object of the same type. |

Reference

AmsNetId Class [665]
**AmsNetId.CompareTo Method (Object)**

Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public int CompareTo(
    Object obj
)
```

**Parameters**

- `obj` Type: `System.Object`
  An object to compare with this instance.

**Return Value**

Type: `Int32`
A value that indicates the relative order of the objects being compared. The return value has these meanings: Value Meaning Less than zero This instance precedes obj in the sort order. Zero This instance occurs in the same position in the sort order as obj. Greater than zero This instance follows obj in the sort order.

**Implements**

`IComparable.CompareTo(Object)`

**Reference**

- AmsNetId Class [› 665]
- CompareTo Overload [› 677]
- TwinCAT.Ads Namespace [› 151]

**AmsNetId.CompareTo Method (AmsNetId)**

Compares the current object with another object of the same type.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public int CompareTo(
    AmsNetId other
)
```

Parameters

other Type: TwinCAT.Ads.AmsNetId
An object to compare with this object.

Return Value

Type: Int32
A value that indicates the relative order of the objects being compared. The return value has the following meanings: Value Meaning Less than zero This object is less than the other parameter. Zero This object is equal to other. Greater than zero This object is greater than other.

Implements

IComparable.T..CompareTo(T)

Reference

AmsNetId Class
CompareTo Overload
TwinCAT.Ads Namespace

6.2.25.3.4 AmsNetId.Equals Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals(Object)</td>
<td>Equals (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Equals(AmsNetId, AmsNetId)</td>
<td>Determines whether the specified AmsNetId is equal.</td>
</tr>
</tbody>
</table>

Reference

AmsNetId Class
TwinCAT.Ads Namespace

AmsNetId.Equals Method (Object)

Equals

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
public override bool Equals(
    Object obj
)
```

Parameters

- **obj**
  *Type: System.Object*
  The object to compare with the current object.

Return Value

- **Type: Boolean**
  *true if the specified Object is equal to this instance; otherwise, false.*

Reference

- AmsNetId Class [665]
- Equals Overload [679]
- TwinCAT.Ads Namespace [151]

### AmsNetId.Equals Method (AmsNetId, AmsNetId)

Determines whether the specified AmsNetId[s] are equal.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static bool Equals(
    AmsNetId netIDA,
    AmsNetId netIDB
)
```

Parameters

- **netIDA**
  *Type: TwinCAT.Ads.AmsNetId [665]*
  The net IDA.

- **netIDB**
  *Type: TwinCAT.Ads.AmsNetId [665]*
  The net IDB.

Return Value

- **Type: Boolean**
  *true if the specified net IDA is equal; otherwise, false.*

Reference

- AmsNetId Class [665]
- Equals Overload [679]
6.2.25.3.5 AmsNetId.FromBinHexString Method

Creates the AmsNetId from bin hex string.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static AmsNetId FromBinHexString(
    string str
)
```

Parameters

- `str`: Type: System.String
  The BinHex string.

Return Value

- Type: AmsNetId
  AmsNetId.

Reference

AmsNetId Class
TwinCAT.Ads Namespace

6.2.25.3.6 AmsNetId.GetHashCode Method

Gets the HashCode of the Address

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public override int GetHashCode()
```

Return Value

- Type: Int32
  A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

Reference

AmsNetId Class
TwinCAT.Ads Namespace
6.2.25.3.7 AmsNetId.IsSameTarget Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>IsSameTarget(AmsNetId, AmsNetId)</td>
</tr>
<tr>
<td>S</td>
<td>IsSameTarget(AmsNetId, AmsNetId, Boolean)</td>
</tr>
</tbody>
</table>

Determines whether the AmsNetId\[665\]s refer to the same target.

Reference

AmsNetId Class [665]

AmsNetId.IsSameTarget Method (AmsNetId, AmsNetId)

Determines whether the AmsNetId\[665\]s refer to the same target.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static bool IsSameTarget(
    AmsNetId netIDA,
    AmsNetId netIDB
)
```

Parameters

- **netIDA**
  - Type: TwinCAT.Ads.AmsNetId [665]
  - NetID of target system A

- **netIDB**
  - Type: TwinCAT.Ads.AmsNetId [665]
  - NetID of target system B

Return Value

- Type: Boolean
  - true if the target systems are the same, otherwise false.

Remarks

In comparison to the Equals(Object) [679] or Equals(AmsNetId, AmsNetId) [680] methods, this Method also checks against the LocalHost ID, which means that LocalHost [675] is the same target as Local [674]

Reference

AmsNetId Class [665]
IsSameTarget Overload [682]

TwinCAT.Ads Namespace [151]

AmsNetId.Equals(Object) [679]

AmsNetId.Equals(AmsNetId, AmsNetId) [680]

AmsNetId.IsSameTarget Method (AmsNetId, AmsNetId, Boolean)

Determines whether the AmsNetId [665]s refer to the same target.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

**C#**

```csharp
public static bool IsSameTarget(
    AmsNetId netIDA,
    AmsNetId netIDB,
    bool ignoreSubId
)
```

Parameters

- **netIDA**
  Type: TwinCAT.Ads.AmsNetId [665]
  NetID of target system A

- **netIDB**
  Type: TwinCAT.Ads.AmsNetId [665]
  NetID of target system B

- **ignoreSubId**
  Type: System.Boolean
  Indicates only to check the 4 parts of the address (for SubId check).

Return Value

Type: Boolean
true if the target systems are the same, otherwise false.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>netIDA</td>
</tr>
<tr>
<td>ArgumentNullException</td>
<td>netIDB</td>
</tr>
</tbody>
</table>

Remarks

In comparison to the Equals(Object) [679] or Equals(AmsNetId, AmsNetId) [680] methods, this Method also checks against the LocalHost ID, which means that LocalHost [675] is the same target as Local [674].

Reference

AmsNetId Class [665]

IsSameTarget Overload [682]

TwinCAT.Ads Namespace [151]
6.25.3.8 AmsNetId.NetIdsEqual Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="684" alt="NetIdsEqual(Byte)" /></td>
<td>Compares the netIds</td>
</tr>
<tr>
<td><img src="685" alt="NetIdsEqual(Byte, Byte)" /></td>
<td>Compares the NetIds</td>
</tr>
<tr>
<td><img src="685" alt="NetIdsEqual(Byte, Byte, Boolean)" /></td>
<td>Compares the NetIds for SubID equality.</td>
</tr>
</tbody>
</table>

**Reference**

AmsNetId Class [665]

AsmCAT.Ads Namespace [151]

**AmsNetId.NetIdsEqual Method (Byte.)**

Compares the netIds

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool NetIdsEqual(
    byte[] netId
)
```

**Parameters**

| netId | Type: System.Byte.  
| NetId in bytes. |

**Return Value**

Type: Boolean  
true if XXXX, false otherwise.

**Reference**

AmsNetId Class [665]  
NetIdsEqual Overload [684]
AmsNetId.NetIdsEqual Method (.Byte., .Byte.)

Compares the NetIds

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public static bool NetIdsEqual(
    byte[] netId1,
    byte[] netId2
)

Parameters

netId1 Type: System.Byte
NetID1

netId2 Type: System.Byte
NetId2

Return Value

Type: Boolean
ture if XXXX, false otherwise.

Reference

AmsNetId Class

NetIdsEqual Overload

TwinCAT.Ads Namespace


Compares the NetIds for SubID equality.

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public static bool NetIdsEqual(
    byte[] netId1,
    byte[] netId2,
    bool ignoreSubID
)
Parameters

netId1  
Type: System.Byte
NetID1

netId2  
Type: System.Byte
NetId2

ignoreSubID  
Type: System.Boolean
Ignores the SubId part of the NetId (last 2 numbers/bytes).

Return Value

Type: Boolean
true if the SubIds are equal, false otherwise.

Remarks

The first 4 numbers indicate the global ID of the AmsNetId. The last 2 numbers indicate the (local) submodule.

Reference

AmsNetId Class [665]
NetIdsEqual Overload [684]
TwinCAT.Ads Namespace [151]

6.25.3.9 AmsNetId.Parse Method

Converts the string representation of the address to AmsNetId [665].

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static AmsNetId Parse(
    string str
)     

Parameters

str  
Type: System.String
The string to parse.

Return Value

Type: AmsNetId [665]
AmsNetId.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormatException</td>
<td>Format of AmsNetId is not valid!</td>
</tr>
</tbody>
</table>
Reference

AmsNetId Class [665]
TwinCAT.Ads Namespace [151]

6.2.25.3.10 AmsNetId.ToBinHex Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToBinHex()</td>
<td>Converts the AmsNetId [665] to a BinHex string.</td>
</tr>
<tr>
<td>ToBinHex(AmsNetId)</td>
<td>Converts the specified AmsNetId [665] to a BinHex string.</td>
</tr>
</tbody>
</table>

Reference

AmsNetId Class [665]
TwinCAT.Ads Namespace [151]

AmsNetId.ToBinHex Method

Converts the AmsNetId [665] to a BinHex string.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public string ToBinHex()
```

Return Value

Type: String
System.String.

Reference

AmsNetId Class [665]
ToBinHex Overload [687]
TwinCAT.Ads Namespace [151]

AmsNetId.ToBinHex Method (AmsNetId)

Converts the specified AmsNetId [665] to a BinHex string.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
## Syntax

**C#**

```csharp
public static string ToBinHex(
    AmsNetId netId
)
```

### Parameters

- **netId**
  - Type: `TwinCAT.Ads.AmsNetId` ([665])
  - NetId to convert

### Return Value

- Type: `String`

**Reference**

- `AmsNetId Class` ([665])
- `ToBinHex Overload` ([687])
- `TwinCAT.Ads Namespace` ([151])

---

### 6.2.25.3.11 AmsNetId.ToBytes Method

Converts the NetId object to byte array

**Namespace:** `TwinCAT.Ads` ([151])

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public byte[] ToBytes()
```

### Return Value

- Type: `Byte`

**Reference**

- `AmsNetId Class` ([665])
- `TwinCAT.Ads Namespace` ([151])

---

### 6.2.25.3.12 AmsNetId.ToString Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToString</td>
<td>Converts the netId to string (Overrides <code>Object.ToString</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ToString(String, IFormatProvider)</td>
<td>Returns a String that represents this instance.</td>
</tr>
</tbody>
</table>

### Reference

AmsNetId Class [› 665]

TwinCAT.Ads Namespace [› 151]

#### AmsNetId.ToString Method

Converts the netId to string

**Namespace:** TwinCAT.Ads [› 151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public override string ToString()
```

**Return Value**

Type: String

A String that represents this instance.

**Reference**

AmsNetId Class [› 665]

ToString Overload [› 688]

TwinCAT.Ads Namespace [› 151]

#### AmsNetId.ToString Method (String, IFormatProvider)

Returns a String that represents this instance.

**Namespace:** TwinCAT.Ads [› 151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public string ToString(
    string format,
    IFormatProvider formatProvider
)
```

**Parameters**

format

Type: System.String

The format.
formatProvider

The format provider.

Return Value

Type: String
A String that represents this instance.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormatException</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

<table>
<thead>
<tr>
<th>Formatting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>g</td>
<td>Standard formatting</td>
</tr>
<tr>
<td>x</td>
<td>Formatting as Hexadecimal (small letters)</td>
</tr>
<tr>
<td>X</td>
<td>Formatting as Hexadecimal (big letters)</td>
</tr>
</tbody>
</table>

Reference

AmsNetId Class [665]

ToString Overload [688]

TwinCAT.Ads Namespace [151]

6.2.25.3.13 AmsNetId.TryParse Method

Converts the string representation of the address to AmsNetId [665].

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static bool TryParse(
    string str,
    out AmsNetId netId
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>str</td>
<td>System.String</td>
<td>The string to parse.</td>
</tr>
<tr>
<td>netId</td>
<td>TwinCAT.Ads.AmsNetId [665]</td>
<td>The parsed AmsNetId [665].</td>
</tr>
</tbody>
</table>

Return Value

Type: Boolean
true if parsed, false otherwise.
Reference

AmsNetId Class [665]

TwinCAT.Ads Namespace [151]

### 6.2.25.3.14 AmsNetId.TryWriteBytes Method

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public bool TryWriteBytes(Span span)
```

**Parameters**

| span | Type: Span |

**Return Value**

Type: Boolean

Reference

AmsNetId Class [665]

TwinCAT.Ads Namespace [151]

### 6.2.25.4 AmsNetId Operators

The AmsNetId [665] type exposes the following members.

<table>
<thead>
<tr>
<th>Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>(=)</td>
</tr>
<tr>
<td>&lt; (&gt;)</td>
</tr>
</tbody>
</table>

Reference

AmsNetId Class [665]

TwinCAT.Ads Namespace [151]
6.2.25.4.1  AmsNetId.Equality Operator

Operator==

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public static bool operator ==(AmsNetId o1, AmsNetId o2)
```

**Parameters**

- **o1**
  - Type: TwinCAT.Ads.AmsNetId
  - The o1.

- **o2**
  - Type: TwinCAT.Ads.AmsNetId
  - The o2.

**Return Value**

Type: Boolean
The result of the operator.

**Reference**

AmsNetId Class
TwinCAT.Ads Namespace

6.2.25.4.2  AmsNetId.Inequality Operator

Implements the != operator.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public static bool operator !=(AmsNetId o1, AmsNetId o2)
```

**Parameters**

- **o1**
  - Type: TwinCAT.Ads.AmsNetId
  - The o1.

- **o2**
  - Type: TwinCAT.Ads.AmsNetId
  - The o2.
Return Value

Type: Boolean
The result of the operator.

Reference

AmsNetId Class [› 665]
TwinCAT.Ads Namespace [› 151]

6.2.26 AmsPort Enumeration

AmsPorts

Namespace: TwinCAT.Ads [› 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public enum AmsPort
```

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router</td>
<td>1</td>
<td>AMS Router (Port 1)</td>
</tr>
<tr>
<td>Debugger</td>
<td>2</td>
<td>AMS Debugger (Port 2)</td>
</tr>
<tr>
<td>R0_TComServer</td>
<td>10</td>
<td>The TCom Server. Dpc or passive level.</td>
</tr>
<tr>
<td>R0_TComServerTask</td>
<td>11</td>
<td>TCom Server Task. RT context.</td>
</tr>
<tr>
<td>R0_TComServer_PL</td>
<td>12</td>
<td>TCom Serve Task. Passive level.</td>
</tr>
<tr>
<td>R0_TcDebugger</td>
<td>20</td>
<td>TwinCAT Debugger</td>
</tr>
<tr>
<td>R0_TcDebuggerTask</td>
<td>21</td>
<td>TwinCAT Debugger Task</td>
</tr>
<tr>
<td>R0_LicenseServer</td>
<td>30</td>
<td>The License Server (Port 30)</td>
</tr>
<tr>
<td>Logger</td>
<td>100</td>
<td>Logger (Port 100)</td>
</tr>
<tr>
<td>EventLog</td>
<td>110</td>
<td>Event Logger (Port 110)</td>
</tr>
<tr>
<td>DeviceApplication</td>
<td>120</td>
<td>application for coupler (EK), gateway (EL), etc.</td>
</tr>
<tr>
<td>EventLog_UM</td>
<td>130</td>
<td>Event Logger UM</td>
</tr>
<tr>
<td>EventLog_RT</td>
<td>131</td>
<td>Event Logger RT</td>
</tr>
<tr>
<td>EventLogPublisher</td>
<td>132</td>
<td>Event Logger Publisher</td>
</tr>
<tr>
<td>R0_Realtime</td>
<td>200</td>
<td>R0 Realtime (Port 200)</td>
</tr>
<tr>
<td>R0_Trace</td>
<td>290</td>
<td>R0 Trace (Port 290)</td>
</tr>
<tr>
<td>R0_IO</td>
<td>300</td>
<td>R0 IO (Port 300)</td>
</tr>
<tr>
<td>R0_NC</td>
<td>500</td>
<td>NC (R0) (Port 500)</td>
</tr>
<tr>
<td>R0_NCSAF</td>
<td>501</td>
<td>R0 Satzausführung (Port 501)</td>
</tr>
<tr>
<td>R0_NCSVB</td>
<td>511</td>
<td>R0 Satzvorbereitung (Port 511)</td>
</tr>
<tr>
<td>R0_NCINSTANCE</td>
<td>520</td>
<td>Preconfigured Nc2-Nc3-Instance</td>
</tr>
<tr>
<td>R0_ISG</td>
<td>550</td>
<td>R0 ISG (Port 550)</td>
</tr>
<tr>
<td>R0_CNC</td>
<td>600</td>
<td>R0 CNC (Port 600)</td>
</tr>
<tr>
<td>R0_LINE</td>
<td>700</td>
<td>R0 Line (Port 700)</td>
</tr>
<tr>
<td>Member name</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>R0_PLC</td>
<td>800</td>
<td>R0 PLC (Port 800)</td>
</tr>
<tr>
<td>Tc2_Plc1</td>
<td>801</td>
<td>Tc2 PLC RuntimeSystem 1 (Port 801)</td>
</tr>
<tr>
<td>Tc2_Plc2</td>
<td>811</td>
<td>Tc2 PLC RuntimeSystem 2 (Port 811)</td>
</tr>
<tr>
<td>Tc2_Plc3</td>
<td>821</td>
<td>Tc2 PLC RuntimeSystem 3 (Port 821)</td>
</tr>
<tr>
<td>Tc2_Plc4</td>
<td>831</td>
<td>Tc2 PLC RuntimeSystem 4 (Port 831)</td>
</tr>
<tr>
<td>R0_RTS</td>
<td>850</td>
<td>R0 RTS (Port 850)</td>
</tr>
<tr>
<td>CamshaftController</td>
<td>900</td>
<td>Camshaft Controller (R0) (Port 900)</td>
</tr>
<tr>
<td>R0_CAMTOOL</td>
<td>950</td>
<td>R0 CAM Tool (Port 950)</td>
</tr>
<tr>
<td>R0_USER</td>
<td>2000</td>
<td>R0 User (Port 2000)</td>
</tr>
<tr>
<td>SystemService</td>
<td>10000</td>
<td>System Service (AMSPORT_R3_SYSSERV, 10000)</td>
</tr>
<tr>
<td>R3_CTRLPROG</td>
<td>10000</td>
<td>(Port 10000)</td>
</tr>
<tr>
<td>R3_SYSCTRL</td>
<td>10001</td>
<td>(Port 10001)</td>
</tr>
<tr>
<td>R3_SYSSAMPLER</td>
<td>10100</td>
<td>Port 10100</td>
</tr>
<tr>
<td>R3_TCPRAWCONN</td>
<td>10200</td>
<td>Port 10200</td>
</tr>
<tr>
<td>R3_TCPIPSERVER</td>
<td>10201</td>
<td>Port 10201</td>
</tr>
<tr>
<td>R3_SYSMANAGER</td>
<td>10300</td>
<td>Port 10300</td>
</tr>
<tr>
<td>R3_SMSSERVER</td>
<td>10400</td>
<td>Port 10400</td>
</tr>
<tr>
<td>R3_MODBUSERVER</td>
<td>10500</td>
<td>Port 10500</td>
</tr>
<tr>
<td>R3_AMSLOGGER</td>
<td>10502</td>
<td>Port 10502</td>
</tr>
<tr>
<td>R3_S7SERVER</td>
<td>10600</td>
<td>Obsolete, Port 10600</td>
</tr>
<tr>
<td>R3_XMLDATASERVER</td>
<td>10600</td>
<td>Port 10600</td>
</tr>
<tr>
<td>R3_AUTOCONFIG</td>
<td>10700</td>
<td>Port 10700</td>
</tr>
<tr>
<td>R3_PLCCONTROL</td>
<td>10800</td>
<td>Port 10800</td>
</tr>
<tr>
<td>R3_FTPCLIENT</td>
<td>10900</td>
<td>Port 10900</td>
</tr>
<tr>
<td>R3_NCTRL</td>
<td>11000</td>
<td>Port 11000</td>
</tr>
<tr>
<td>R3_NCINTERPRETER</td>
<td>11500</td>
<td>Port 11500</td>
</tr>
<tr>
<td>R3_GSTINTERPRETER</td>
<td>11600</td>
<td>Port 11600</td>
</tr>
<tr>
<td>R3_STRECKECTRL</td>
<td>12000</td>
<td>Port 12000</td>
</tr>
<tr>
<td>R3_CAMCTRL</td>
<td>13000</td>
<td>Port 13000</td>
</tr>
<tr>
<td>R3_SCOPE</td>
<td>14000</td>
<td>Port 14000</td>
</tr>
<tr>
<td>R3_CONDITIONMON</td>
<td>14100</td>
<td>Port 14100</td>
</tr>
<tr>
<td>R3_SINECH1</td>
<td>15000</td>
<td>Port 15000</td>
</tr>
<tr>
<td>R3_CONTROLNET</td>
<td>16000</td>
<td>Port 16000</td>
</tr>
<tr>
<td>R3_OPSCSERVER</td>
<td>17000</td>
<td>Port 17000</td>
</tr>
<tr>
<td>R3_OPCCLIENT</td>
<td>17500</td>
<td>Port 17500</td>
</tr>
<tr>
<td>R3_MAILSERVER</td>
<td>18000</td>
<td>Port 18000</td>
</tr>
<tr>
<td>R3_EL60XX</td>
<td>19000</td>
<td>Port 19000</td>
</tr>
<tr>
<td>R3_MANAGEMENT</td>
<td>19100</td>
<td>Port 19100</td>
</tr>
<tr>
<td>R3_MIELEHOME</td>
<td>19200</td>
<td>Port 19200</td>
</tr>
<tr>
<td>R3_CPLINK3</td>
<td>19300</td>
<td>Port 19300</td>
</tr>
<tr>
<td>R3_VNSERVICE</td>
<td>19500</td>
<td>Port 19500</td>
</tr>
<tr>
<td>R3_MULTIUSER</td>
<td>19600</td>
<td>Multiuser (Port 19600)</td>
</tr>
<tr>
<td>USEDEFAULT</td>
<td>65535</td>
<td></td>
</tr>
</tbody>
</table>
6.2.27 AmsRouterNotificationEventArgs Class

Arguments for the IRouterNotificationProvider [973] events.

Inheritance Hierarchy

System.Object
  System.EventArgs
  TwinCAT.Ads.AmsRouterNotificationEventArgs

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public sealed class AmsRouterNotificationEventArgs : EventArgs

The AmsRouterNotificationEventArgs type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>赧</td>
<td>AmsRouterNotificationEventArgs [695]</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>魉</td>
<td>State [696]</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>forall</td>
<td>Equals</td>
</tr>
<tr>
<td>forall</td>
<td>GetHashCode</td>
</tr>
<tr>
<td>forall</td>
<td>GetType</td>
</tr>
<tr>
<td>forall</td>
<td>ToString</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]

6.2.27.1 AmsRouterNotificationEventArgs Constructor

Initializes a new instance of the AmsRouterNotificationEventArgs class.
### Syntax

**C#**

```csharp
public AmsRouterNotificationEventArgs(AmsRouterState value)
```

### Parameters

- **value**
  - **Type:** TwinCAT.Ads.AmsRouterState
  - Value of the ADS variable.

### Reference

- **AmsRouterNotificationEventArgs Class [6.95]**
- **TwinCAT.Ads Namespace [6.151]**

### 6.2.27.2 AmsRouterNotificationEventArgs Properties

The `AmsRouterNotificationEventArgs` type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>

### Reference

- **AmsRouterNotificationEventArgs Class [6.95]**
- **TwinCAT.Ads Namespace [6.151]**

### 6.2.27.2.1 AmsRouterNotificationEventArgs.State Property

Current state of the AMS Router.

**Namespace:** TwinCAT.Ads [6.151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AmsRouterState State { get; }
```

### Property Value

- **Type:** AmsRouterState [6.97]
6.2.27.3 **AmsRouterNotificationEventArgs Methods**

The *AmsRouterNotificationEventArgs* type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>= Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <em>Object.</em>)</td>
</tr>
<tr>
<td>= GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <em>Object.</em>)</td>
</tr>
<tr>
<td>= GetType</td>
<td>Gets the <em>Type</em> of the current instance. (Inherited from <em>Object.</em>)</td>
</tr>
<tr>
<td>= ToString</td>
<td>Returns a string that represents the current object. (Inherited from <em>Object.</em>)</td>
</tr>
</tbody>
</table>

Reference

*AmsRouterNotificationEventArgs* Class [695]

TwinCAT.Ads Namespace [151]

### AmsRouterState Enumeration

State of the AMS Router.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public enum AmsRouterState
```

**Members**

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>Unknown Router State (None, Uninitialized)</td>
</tr>
<tr>
<td>Stop</td>
<td>1</td>
<td>AMS Router is stopped.</td>
</tr>
<tr>
<td>Start</td>
<td>2</td>
<td>AMS Router is started.</td>
</tr>
<tr>
<td>Removed</td>
<td>3</td>
<td>AMS Router has been removed.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]
6.2.29 DeviceInfo Class

The structure contains the name and the version information of the device.

Inheritance Hierarchy

System.Object
   TwinCAT.Ads.DeviceInfo

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class DeviceInfo
```

The DeviceInfo type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceInfo</td>
<td>Initializes a new instance of the DeviceInfo class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Empty / Uninitialized DeviceInfo</td>
</tr>
<tr>
<td>IsEmpty</td>
<td>Gets a value indicating whether this instance is empty.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the device.</td>
</tr>
<tr>
<td>Version</td>
<td>Gets the version information.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace
6.2.29.1   **DeviceInfo Constructor**

Initializes a new instance of the `DeviceInfo` class.

**Namespace:**  [TwinCAT.Ads](#)  
**Assembly:**  [TwinCAT.Ads.Abstractions](#)  
**Version:**  5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public DeviceInfo(
    string name,
    AdsVersion version
)
```

**Parameters**

- `name`  
  - Type: `System.String`  
  - The name.

- `version`  
  - Type: `TwinCAT.Ads.AdsVersion`  
  - The version.

**Reference**

- [DeviceInfo Class](#)
- [TwinCAT.Ads Namespace](#)

6.2.29.2   **DeviceInfo Properties**

The `DeviceInfo` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty [699]</td>
<td>Empty / Uninitialized DeviceInfo</td>
</tr>
<tr>
<td>IsEmpty [700]</td>
<td>Gets a value indicating whether this instance is empty.</td>
</tr>
<tr>
<td>Name [700]</td>
<td>Gets the name of the device.</td>
</tr>
</tbody>
</table>

**Reference**

- [DeviceInfo Class](#)
- [TwinCAT.Ads Namespace](#)

6.2.29.2.1   **DeviceInfo.Empty Property**

Empty / Uninitialized DeviceInfo
Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static DeviceInfo Empty { get; }

Property Value

Type: DeviceInfo
The empty.

Reference

DeviceInfo Class

TwinCAT.Ads Namespace

6.2.29.2.2 DeviceInfo.IsEmpty Property

Gets a value indicating whether this instance is empty.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public bool IsEmpty { get; }

Property Value

Type: Boolean
true if this instance is empty; otherwise, false.

Reference

DeviceInfo Class

TwinCAT.Ads Namespace

6.2.29.2.3 DeviceInfo.Name Property

Gets the name of the device.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public string Name { get; }
6.2.29.2.4 DeviceInfo.Version Property

Gets the version information.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AdsVersion Version { get; }
```

### Property Value

**Type:** AdsVersion

### Reference

DeviceInfo Class [698]

TwinCAT.Ads Namespace [151]

### 6.2.29.3 DeviceInfo Methods

The DeviceInfo [698] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Reference

DeviceInfo Class [698]
6.2.30 IAdsAnyAccess Interface

Interface for accessing ADS 'Any' objects.

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IAdsAnyAccess
```

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReadAny(UInt32, Type)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny(UInt32, Type, Int32.)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny(UInt32, UInt32, Type)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny(UInt32, UInt32, Type, Int32.)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny.T.(UInt32)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny.T.(UInt32, Int32.)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny.T.(UInt32, UInt32)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAny.T.(UInt32, UInt32, Int32.)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyAsync(UInt32, Type, CancellationToken)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyAsync(UInt32, Type, Int32., CancellationToken)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [720]</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, Int32, CancellationToken) [721]</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, CancellationToken) [721]</td>
<td>Reads data synchronously from an ADS device.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Int32, CancellationToken) [719]</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, CancellationToken) [719]</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, CancellationToken) [715]</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyString(UInt32, Int32, Encoding) [722]</td>
<td>Reads a string from the specified symbol/variable.</td>
</tr>
<tr>
<td>ReadAnyString(UInt32, UInt32, Int32, Encoding) [723]</td>
<td>Reads as string from a specified address.</td>
</tr>
<tr>
<td>ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [724]</td>
<td>Reads a string asynchronously from the specified symbol/variable</td>
</tr>
<tr>
<td>ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [725]</td>
<td>Reads a string from a specified address asynchronously.</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object) [726]</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object, Int32.)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAny(UInt32, UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAny(UInt32, UInt32, Object, Int32.)</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, CancellationToken)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, Int32., CancellationToken)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, Int32., CancellationToken)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
</tbody>
</table>

Reference
TwinCAT.Ads Namespace [ 151 ]

### 6.2.30.1 IAdsAnyAccess Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAny(UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32) [707]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, .Int32.) [708]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, UInt32) [710]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, UInt32, .Int32.) [711]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, CancellationToken) [716]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [718]</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [720]</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [721]</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync.T.(UInt32, CancellationToken) [715]</td>
<td>Reads data synchronously from an ADS device.</td>
</tr>
<tr>
<td>ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [715]</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [717]</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [719]</td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadAnyString(UInt32, Int32, Encoding)</td>
<td>Reads a string from the specified symbol/variable</td>
</tr>
<tr>
<td>ReadAnyString(UInt32, UInt32, Int32, Encoding)</td>
<td>Reads as string from a specified address.</td>
</tr>
<tr>
<td>ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken)</td>
<td>Reads a string asynchronously from the specified symbol/variable</td>
</tr>
<tr>
<td>ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken)</td>
<td>Reads a string from a specified address asynchronously.</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object, Int32)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object</td>
</tr>
<tr>
<td>WriteAny(UInt32, UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, CancellationToken)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, Int32, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, .Int32, CancellationToken)</td>
</tr>
</tbody>
</table>

**Description**

Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

### Reference

IAdsAnyAccess Interface [➤ 702]

TwinCAT.Ads Namespace [➤ 151]

### 6.2.30.1.1 IAdsAnyAccess.ReadAny Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAny.T.(UInt32)</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, .Int32.)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type)</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, .Int32.)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, .Int32.)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, .Int32.)</td>
</tr>
</tbody>
</table>

**Description**

Reads data synchronously from an ADS device and writes it to an object.

### Reference

IAdsAnyAccess Interface [➤ 702]

TwinCAT.Ads Namespace [➤ 151]

### IAdsAnyAccess.ReadAny.T. Method (UInt32)

Reads data synchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads [➤ 151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```c#
T ReadAny<T>(
    uint variableHandle
)
```

Parameters

variableHandle Type: `System.UInt32`
Handle of the ADS variable.

Type Parameters

T The type of the value to read.

Return Value

Type: `T`
The value of the read symbol.

Reference

IAdsAnyAccess Interface [702]

ReadAny Overload [707]

TwinCAT.Ads Namespace [151]

IAdsAnyAccess.ReadAny.T. Method (UInt32, Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```c#
T ReadAny<T>(
    uint variableHandle,
    int[] args
)
```

Parameters

variableHandle Type: `System.UInt32`
Handle of the ADS variable.

args Type: `System.Int32`
Additional arguments.

Type Parameters

T The type of the value to read.
Return Value

Type: T
The value of the read symbol.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395]</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

IAdsAnyAccess Interface [702]

ReadAny Overload [707]

TwinCAT.Ads Namespace [151]

IAdsAnyAccess.ReadAny Method (UInt32, Type)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

    Object ReadAny(
        uint variableHandle,
        Type type
    )

Parameters

<table>
<thead>
<tr>
<th>parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variableHandle</td>
<td>System.UInt32</td>
<td>Handle of the ADS variable.</td>
</tr>
<tr>
<td>type</td>
<td>System.Type</td>
<td>Type of the object to be read.</td>
</tr>
</tbody>
</table>

Return Value

Type: Object
The read object.

Reference

IAdsAnyAccess Interface [702]

ReadAny Overload [707]
### TwinCAT.Ads Namespace [› 151]

#### IAdsAnyAccess.ReadAny<T>. Method (UInt32, UInt32)

Reads data synchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads [› 151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```
T ReadAny<T>(
    uint indexGroup,
    uint indexOffset
)
```

**Parameters**

- `indexGroup` Type: System.UInt32
  Index group of the ADS variable.

- `indexOffset` Type: System.UInt32
  Index offset of the ADS variable.

**Type Parameters**

- `T` The type of the object to be read.

**Return Value**

- Type: T
  The read value.

**Reference**

- IAdsAnyAccess Interface [› 702]
- ReadAny Overload [› 707]
- TwinCAT.Ads Namespace [› 151]

#### IAdsAnyAccess.ReadAny Method (UInt32, Type, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads [› 151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```
Object ReadAny(
    uint variableHandle,
    Type type,
    int[] args
)
```
Parameters

variableHandle  
Type: System.UInt32  
Handle of the ADS variable.

type  
Type: System.Type  
Type of the object to be read.

args  
Type: System.Int32  
Additional arguments.

Return Value

Type: Object  
The read value.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395]</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

IAdsAnyAccess Interface [702]

ReadStream Overload [707]

TwinCAT.Ads Namespace [151]

IAdsAnyAccess.ReadAny.T. Method (UInt32, UInt32, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  
T ReadAny<T>(
    uint indexGroup,
    uint indexOffset,
    int[] args)

Parameters

indexGroup  
Type: System.UInt32  
Index group of the ADS variable.

indexOffset  
Type: System.UInt32  
Index offset of the ADS variable.
arg

Additional arguments.

Type Parameters

T

The type of the object to be read.

Return Value

Type: T

The read value.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

IAdsAnyAccess Interface [702]

ReadAny Overload [707]

TwinCAT.Ads Namespace [151]

IAdsAnyAccess.ReadAny Method (UInt32, UInt32, Type)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdcf72bc0e15da1c14

Syntax

C#

Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type
)

Parameters

indexGroup

Type: System.UInt32

Index group of the ADS variable.

indexOffset

Type: System.UInt32

Index offset of the ADS variable.

type

Type: System.Type

Type of the object to be read.
Return Value
Type: Object
The read value.

Reference
IAdsAnyAccess Interface [702]
ReadAny Overload [707]
TwinCAT.Ads Namespace [151]

IAdsAnyAccess.ReadAny Method (UInt32, UInt32, Type, .Int32.)
Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args
)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>indexGroup</td>
<td>System.UInt32</td>
<td>Index group of the ADS variable.</td>
</tr>
<tr>
<td>indexOffset</td>
<td>System.UInt32</td>
<td>Index offset of the ADS variable.</td>
</tr>
<tr>
<td>type</td>
<td>System.Type</td>
<td>Type of the object to be read.</td>
</tr>
</tbody>
</table>

Return Value
Type: Object
The read value.

Remarks
As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [2395].</td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>
### 6.2.30.1.2 IAdsAnyAccess.ReadAnyAsync Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReadAnyAsync(UInt32, Type, CancellationToken)</code></td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyAsync(UInt32, UInt32, Type, CancellationToken)</code></td>
<td>Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyAsync(UInt32, Type, CancellationToken)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyAsync(UInt32, UInt32, CancellationToken)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyAsync(UInt32, Int32, CancellationToken)</code></td>
<td>Reads data synchronously from an ADS device.</td>
</tr>
<tr>
<td><code>ReadAnyAsync(UInt32, Type, Int32, CancellationToken)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyAsync(UInt32, Int32, CancellationToken)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyAsync(UInt32, UInt32, Int32, CancellationToken)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><code>ReadAnyAsync(UInt32, Int32, Int32, CancellationToken)</code></td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
</tbody>
</table>
IAdsAnyAccess.ReadAnyAsync<T> Method (UInt32, CancellationToken)

Reads data synchronously from an ADS device.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

Task<ResultValue<T>> ReadAnyAsync<T>(
    uint variableHandle,
    CancellationToken cancel
)

Parameters

variableHandle Type: System.UInt32
The variable/symbol handle.

cancel Type: System.Threading.CancellationToken
The cancellation token.

Type Parameters

T The Type of the value to be read.

Return Value

Type: Task<ResultValue[T]>
A task that represents the asynchronous read operation. The ResultAnyValue parameter contains the read value (Value) and the ErrorCode parameter after execution.

Remarks

As object types only primitive types are supported.

Reference

IAdsAnyAccess Interface [702]
ReadAnyAsync Overload [714]
TwinCAT.Ads Namespace [151]

IAdsAnyAccess.ReadAnyAsync<T>. Method (UInt32, .Int32., CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

```csharp
Task<ResultValue<T>> ReadAnyAsync<T>(
    uint variableHandle,
    int[] args,
    CancellationToken cancel
)
```

Parameters

- **variableHandle**
  - Type: `System.UInt32`
  - The variable handle.

- **args**
  - Type: `System.Int32`
  - Additional arguments.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.

Type Parameters

- **T**
  - Type of the object to be read

Return Value

Type: `Task<ResultValue<T>>`.

A task that represents the asynchronous read operation. The `ResultValue.TValue` parameter contains the read value (`Value`) and the `ErrorCode` after execution.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter `args` specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter `args`.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td><code>args[0]</code>: Number of characters in the string typed as <code>FixedSizeZeroTerminated</code> [2395].</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

- `IAdsAnyAccess Interface` [702]
- `ReadAnyAsync Overload` [714]
- `TwinCAT.Ads Namespace` [151]

**IAdsAnyAccess.ReadAnyAsync Method (UInt32, Type, CancellationToken)**

Reads data synchronously from an ADS device and writes it to an object.

**Namespace:** `TwinCAT.Ads` [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
Task<ResultAnyValue> ReadAnyAsync(
    uint variableHandle,
    Type type,
    CancellationToken cancel
)
```

Parameters

- **variableHandle**
  - Type: `System.UInt32`
  - The variable/symbol handle.

- **type**
  - Type: `System.Type`
  - Type of the object to be read.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.

Return Value

- Type: `Task<ResultAnyValue>`
  - A task that represents the asynchronous read operation. The `ResultAnyValue` parameter contains the read value (`Value`) and the `ErrorCode` after execution.

Remarks

As object types only primitive types are supported.

Reference

- `IAdsAnyAccess Interface`
- `ReadAnyAsync Overload`
- `TwinCAT.Ads Namespace`

`IAdsAnyAccess.ReadAnyAsync<T>. Method (UInt32, UInt32, CancellationToken)`

Reads data asynchronously from an ADS device and writes it to an object.

**Namespace:** `TwinCAT.Ads`

**Assembly:** `TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)`

**Version:**

5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Task<ResultValue<T>> ReadAnyAsync<T>(
    uint indexGroup,
    uint indexOffset,
    CancellationToken cancel
)
```

Parameters

- **indexGroup**
  - Type: `System.UInt32`
  - Index group of the ADS variable.

- **indexOffset**
  - Type: `System.UInt32`
  - Index offset of the ADS variable.
**TwinCAT.Ads Namespaces**

**cancel**

Type: `System.Threading.CancellationToken`

The cancellation token.

**Type Parameters**

T

**Return Value**

Type: `Task<ResultAnyValue>`

The asynchronous result.

**Return Value**

Type: `Task<ResultAnyValue>`

A task that represents the asynchronous read operation. The `ResultValue.TValue` parameter contains the read value (`Value`) and the `ErrorCode` parameter after execution.

**Reference**

IAdsAnyAccess Interface

`ReadAnyAsync Overload` [714]

TwinCAT.Ads Namespace [151]

**IAdsAnyAccess.ReadAnyAsync Method (UInt32, Type, .Int32., CancellationToken)**

Reads data synchronously from an ADS device and writes it to an object.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
Task<ResultAnyValue> ReadAnyAsync(
    uint variableHandle,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

**Parameters**

**variableHandle**

Type: `System.UInt32`

The variable handle.

**type**

Type: `System.Type`

Type of the object to be read.

**args**

Type: `System.Int32`

Additional arguments.

**cancel**

Type: `System.Threading.CancellationToken`

The cancellation token.
Return Value

Type: Task<ResultAnyValue>.
A task that represents the asynchronous read operation. The ResultAnyValue parameter contains the read value (Value) and the ErrorCode after execution.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated.</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference

IAdsAnyAccess Interface
ReadAnyAsync Overload
TwinCAT.Ads Namespace

IAdsAnyAccess.ReadAnyAsync.T. Method (UInt32, UInt32, int[], CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Task<ResultValue<T>> ReadAnyAsync<T>(
    uint indexGroup,
    uint indexOffset,
    int[] args,
    CancellationToken cancel
)
```

Parameters

- **indexGroup**
  Type: System.UInt32
  Index group of the ADS variable.

- **indexOffset**
  Type: System.UInt32
  Index offset of the ADS variable.

- **args**
  Type: System.Int32.
  Additional arguments.

- **cancel**
  Type: System.Threading.CancellationToken
  The cancellation token.
Type Parameters

T
  The type of the result value.

Return Value

Type: `Task<ResultAnyValue>`

A task that represents the asynchronous read operation. The `ResultValue.TValue` parameter contains the read value (`Value`) and the `ErrorCode` after execution.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td><code>args[0]</code>: Number of characters in the string typed as <code>FixedSizeZeroTerminated</code></td>
</tr>
<tr>
<td>string[]</td>
<td><code>args[0]</code>: Array of characters</td>
</tr>
</tbody>
</table>

Reference

IAdsAnyAccess Interface ([702]

ReadAnyAsync Overload ([714]

TwinCAT.Ads Namespace ([151]

IAdsAnyAccess.ReadAnyAsync Method (UInt32, UInt32, Type, CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

**Parameters**

<table>
<thead>
<tr>
<th>indexGroup</th>
<th>Type: <code>System.UInt32</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>indexOffset</td>
<td>Type: <code>System.UInt32</code></td>
</tr>
<tr>
<td>type</td>
<td>Type: <code>System.Type</code></td>
</tr>
</tbody>
</table>
cancel

Type: System.Threading.CancellationToken
The cancellation token.

Return Value
Type: Task<ResultAnyValue [998]].
A task that represents the asynchronous read operation. The ResultAnyValue [998] parameter contains the
read value (Value [1000]) and the ErrorCode [992] after execution.

Reference
IAdsAnyAccess Interface [702]
ReadAnyAsync Overload [714]
TwinCAT.Ads Namespace [151]

IAdsAnyAccess.ReadAnyAsync Method (UInt32, UInt32, Type, .Int32., CancellationToken)
Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [151]
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
Task<ResultAnyValue> ReadAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args,
    CancellationToken cancel
)

Parameters
indexGroup
Type: System.UInt32
Index group of the ADS variable.

indexOffset
Type: System.UInt32
Index offset of the ADS variable.

type
Type: System.Type
Type of the object to be read.

args
Type: .System.Int32.
Additional arguments.

cancel
Type: System.Threading.CancellationToken
The cancellation token.

Return Value
Type: Task<ResultAnyValue [998]].
A task that represents the asynchronous read operation. The ResultAnyValue [998] parameter contains the
read value (Value [1000]) and the ErrorCode [992] after execution.
Remarks
As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [\ref{2395}].</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference
IAdsAnyAccess Interface [\ref{702}]
ReadAnyAsync Overload [\ref{714}]
TwinCAT.Ads Namespace [\ref{151}]

6.2.30.1.3 IAdsAnyAccess.ReadAnyString Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAnyString(UInt32, Int32, Encoding) [\ref{722}]</td>
<td>Reads a string from the specified symbol/variable.</td>
</tr>
<tr>
<td>ReadAnyString(UInt32, UInt32, Int32, Encoding) [\ref{723}]</td>
<td>Reads as string from a specified address.</td>
</tr>
</tbody>
</table>

Reference
IAdsAnyAccess Interface [\ref{702}]
TwinCAT.Ads Namespace [\ref{151}]

IAdsAnyAccess.ReadAnyString Method (UInt32, Int32, Encoding)

Reads a string from the specified symbol/variable.

Namespace: TwinCAT.Ads [\ref{151}]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
string ReadAnyString(
    uint variableHandle,
    int len,
    Encoding encoding
)
```
Parameters

variableHandle Type: System.UInt32
The variable handle.

len Type: System.Int32
The length.

encoding Type: System.Text.Encoding
The encoding.

Return Value

Type: String
The string value.

Reference

IAdsAnyAccess Interface [702]
ReadAnyString Overload [722]
TwinCAT.Ads Namespace [151]

IAdsAnyAccess.ReadAnyString Method (UInt32, UInt32, Int32, Encoding)

Reads as string from a specified address.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

string ReadAnyString(
    uint indexGroup,
    uint indexOffset,
    int len,
    Encoding encoding
)

Parameters

indexGroup Type: System.UInt32
The index group.

indexOffset Type: System.UInt32
The index offset.

len Type: System.Int32
The string length to be read.

encoding Type: System.Text.Encoding
The encoding.

Return Value

Type: String
System.String.
6.2.30.1.4 IAdsAnyAccess.ReadAnyStringAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken)</td>
<td>Reads a string asynchronously from the specified symbol/variable</td>
</tr>
<tr>
<td>ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken)</td>
<td>Reads a string from a specified address asynchronously.</td>
</tr>
</tbody>
</table>

Reference

IAdsAnyAccess Interface [702]
TwinCAT.Ads Namespace [151]

IAdsAnyAccess.ReadAnyStringAsync Method (UInt32, Int32, Encoding, CancellationToken)

Reads a string asynchronously from the specified symbol/variable

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Task<ResultAnyValue> ReadAnyStringAsync(
    uint variableHandle,
    int len,
    Encoding encoding,
    CancellationToken cancel
)
```

Parameters

- variableHandle: Type: System.UInt32
  The variable handle.
- len: Type: System.Int32
  The length.
encoding  
Type: System.Text.Encoding
The encoding.

cancel  
Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultAnyValue> [998].
A task that represents the asynchronous read operation. The ResultAnyValue [998] parameter contains the read value (Value [1000]) and the ErrorCode [992] after execution.

Reference

IAdsAnyAccess Interface [702]

ReadAnyStringAsync Overload [724]

TwinCAT.Ads Namespace [151]

IAdsAnyAccess.ReadAnyStringAsync Method (UInt32, UInt32, Int32, Encoding, CancellationToken)

Reads a string from a specified address asynchronously.

Namespace:  TwinCAT.Ads [151]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e7bc0ea15da1c14

Syntax

C#

Task<ResultAnyValue> ReadAnyStringAsync(
    uint indexGroup,
    uint indexOffset,
    int len,
    Encoding encoding,
    CancellationToken cancel)

Parameters

indexGroup  
Type: System.UInt32
The index group.

indexOffset  
Type: System.UInt32
The index offset.

len  
Type: System.Int32
The string length to be read.

encoding  
Type: System.Text.Encoding
The encoding.

cancel  
Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultAnyValue> [998].
A task that represents the asynchronous read operation. The ResultAnyValue [998] parameter contains the read value (Value [1000]) and the ErrorCode [992] after execution.
### 6.2.30.1.5 IAdsAnyAccess.WriteAny Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>WriteAny(UInt32, Object)</code></td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td><code>WriteAny(UInt32, Object, Int32)</code></td>
<td>Writes an object synchronously to an ADS device. If the Type of the object</td>
</tr>
<tr>
<td></td>
<td>to be written is a string type, the first element of parameter args specifies</td>
</tr>
<tr>
<td></td>
<td>the number of characters of the string.</td>
</tr>
<tr>
<td><code>WriteAny(UInt32, UInt32, Object)</code></td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
<tr>
<td><code>WriteAny(UInt32, UInt32, Object, Int32)</code></td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
</tbody>
</table>

#### Reference

IAdsAnyAccess Interface [702]

TwinCAT.Ads Namespace [151]

---

### IAdsAnyAccess.WriteAny Method (UInt32, Object)

Writes an object synchronously to an ADS device.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
void WriteAny(
    uint variableHandle,
    Object value
)
```

**Parameters**

- **variableHandle**
  - Type: `System.UInt32`
  - Handle of the ADS variable.

- **value**
  - Type: `System.Object`
  - Object to write to the ADS device.
IAdsAnyAccess.WriteAny Method (UInt32, Object, .Int32.)

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#:
```csharp
void WriteAny(
    uint variableHandle,
    Object value,
    int[] args
)
```

**Parameters**

- `variableHandle` Type: `System.UInt32`
  Handle of the ADS variable.

- `value` Type: `System.Object`
  Object to write to the ADS device.

- `args` Type: `System.Int32`.
  Additional arguments.

**Remarks**

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [12395].</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

- IAdsAnyAccess Interface [702]
- WriteAny Overload [726]
- TwinCAT.Ads Namespace [151]
IAdsAnyAccess.WriteAny Method (UInt32, UInt32, Object)

Writes an object synchronously to an ADS device.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value
)

Parameters

indexGroup Type: System.UInt32
Contains the index group number of the requested ADS service.

indexOffset Type: System.UInt32
Contains the index offset number of the requested ADS service.

value Type: System.Object
Object to write to the ADS device.

Reference

IAdsAnyAccess Interface [702]
WriteAny Overload [726]
TwinCAT.Ads Namespace [151]

IAdsAnyAccess.WriteAny Method (UInt32, UInt32, Object, .Int32.)

Writes an object synchronously to an ADS device.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args
)

Parameters

indexGroup Type: System.UInt32
Contains the index group number of the requested ADS service.

indexOffset Type: System.UInt32
Contains the index offset number of the requested ADS service.

value
Type: `System.Object`
Object to write to the ADS device.

args
Type: `System.Int32`
Additional arguments.

Remarks
If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Remarks
As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as <code>FixedLengthZeroTerminated</code></td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

Reference
- `IAdsAnyAccess Interface [702]`
- `WriteAny Overload [726]`
- `TwinCAT.Ads Namespace [151]`

### 6.2.30.1.6 `IAdsAnyAccess.WriteAnyAsync Method`

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>WriteAnyAsync(UInt32, Object, CancellationToken)</code></td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td><code>WriteAnyAsync(UInt32, Object, Int32, CancellationToken)</code></td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td><code>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)</code></td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, Int32, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.</td>
</tr>
</tbody>
</table>

**Reference**

IAdsAnyAccess Interface [702]

TwinCAT.Ads Namespace [151]

**IAdsAnyAccess.WriteAnyAsync Method (UInt32, Object, CancellationToken)**

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
Task<ResultWrite> WriteAnyAsync(  
    uint variableHandle,  
    Object value,  
    CancellationToken cancel)
```

**Parameters**

- **variableHandle**
  - Type: System.UInt32
  - Handle of the ADS variable.

- **value**
  - Type: System.Object
  - Object to write to the ADS device.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

**Return Value**

Type: Task<ResultWrite>[1032].

A task that represents the asynchronous task operation. The result parameter ResultWrite[1032] of the write operation contains the ErrorCode[992].

**Reference**

IAdsAnyAccess Interface [702]

WriteAnyAsync Overload [729]

TwinCAT.Ads Namespace [151]
IAdsAnyAccess.WriteAnyAsync Method (UInt32, Object, .Int32., CancellationToken)

Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
Task<ResultWrite> WriteAnyAsync(
    uint variableHandle,
    Object value,
    int[] args,
    CancellationToken cancel
)
```

**Parameters**

- **variableHandle**  
  Type: `System.UInt32`  
  Handle of the ADS variable.
- **value**  
  Type: `System.Object`  
  Object to write to the ADS device.
- **args**  
  Type: `System.Int32`  
  Additional arguments.
- **cancel**  
  Type: `System.Threading.CancellationToken`  
  The cancellation token.

**Return Value**

Type: `Task<ResultWrite>` [1032].  
A task that represents the asynchronous task operation. The result parameter `ResultWrite` [1032] of the write operation contains the `ErrorCode` [992].

**Remarks**

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>args[0]: Number of characters in the string typed as <code>FixedSizeZeroTerminated</code> [2395].</td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

IAdsAnyAccess Interface [702]  
WriteAnyAsync Overload [729]  
TwinCAT.Ads Namespace [151]
IAdsAnyAccess.WriteAnyAsync Method (UInt32, UInt32, Object, CancellationToken)

Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

**Namespace:** TwinCAT.Ads  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
Task<ResultWrite> WriteAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Object value,
    CancellationToken cancel)
```

**Parameters**

- **indexGroup**  
  Type: `System.UInt32`  
  Contains the index group number of the requested ADS service.

- **indexOffset**  
  Type: `System.UInt32`  
  Contains the index offset number of the requested ADS service.

- **value**  
  Type: `System.Object`  
  Object to write to the ADS device.

- **cancel**  
  Type: `System.Threading.CancellationToken`  
  The cancellation token.

**Return Value**

Type: `Task<ResultWrite>`  
A task that represents the asynchronous task operation. The result parameter `ResultWrite` of the write operation contains the `ErrorCode`.

**Reference**

- `IAdsAnyAccess Interface`  
- `WriteAnyAsync Overload`  
- `TwinCAT.Ads Namespace`  

IAdsAnyAccess.WriteAnyAsync Method (UInt32, UInt32, Object, .Int32., CancellationToken)

Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

**Namespace:** TwinCAT.Ads  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
### Syntax

**C#**

```csharp
Task<ResultWrite> WriteAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args,
    CancellationToken cancel
)
```

### Parameters

- **indexGroup**
  - Type: `System.UInt32`
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - Type: `System.UInt32`
  - Contains the index offset number of the requested ADS service.

- **value**
  - Type: `System.Object`
  - Object to write to the ADS device.

- **args**
  - Type: `System.Int32`
  - Additional arguments.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.

### Return Value

Type: `Task<ResultWrite>`

A task that represents the asynchronous task operation. The result parameter `ResultWrite` of the write operation contains the `ErrorCode`.

### Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter `args` specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter `args`.

<table>
<thead>
<tr>
<th>Type of value Parameter</th>
<th>Necessary Arguments (args)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td><code>args[0]</code>: Number of characters in the string typed as <code>FixedSizeZeroTerminated</code></td>
</tr>
<tr>
<td>string[]</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td></td>
</tr>
</tbody>
</table>

### Reference

- `IAdsAnyAccess Interface` [702]
- `WriteAnyAsync Overload` [729]
- `TwinCAT.Ads Namespace` [151]

#### 6.2.31 **IAdsConnectAddress Interface**

Interface for method to connect the ADS client via AMS Address.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public interface IAdsConnectAddress : IAdsConnection,
    IAdsNotifications, IAdsSymbolicAccess, IAdsAnyAccess,
    IAdsHandle, IAdsReadWrite2, IAdsReadWrite, IAdsStateProvider, IAdsStateControl,
    IAdsSymbolChangedProvider, IAdsRpcInvoke
```

The `IAdsConnectAddress` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address [781]</td>
<td>Gets the <code>AmsAddress [648]</code> of the ADS server. (Inherited from <code>IAdsConnection [765].</code>)</td>
</tr>
<tr>
<td>ClientAddress [781]</td>
<td>Get the <code>AmsAddress [648]</code> of the ADS client. (Inherited from <code>IAdsConnection [765].</code>)</td>
</tr>
<tr>
<td>ConnectionState [85]</td>
<td>Gets the current Connection state of the <code>IConnectionStateProvider [84]</code> (Inherited from <code>IConnectionStateProvider [84].</code>)</td>
</tr>
<tr>
<td>DefaultValueEncoding [76]</td>
<td>Gets the default value encoding. (Inherited from <code>IConnection [74].</code>)</td>
</tr>
<tr>
<td>Id [76]</td>
<td>Gets the Connection Identifier. (Inherited from <code>IConnection [74].</code>)</td>
</tr>
<tr>
<td>IsConnected [76]</td>
<td>Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available. (Inherited from <code>IConnection [74].</code>)</td>
</tr>
<tr>
<td>IsLocal [782]</td>
<td>Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer. (Inherited from <code>IAdsConnection [765].</code>)</td>
</tr>
<tr>
<td>Session [77]</td>
<td>Gets the session that initiated this <code>IConnection [74]</code> (Inherited from <code>IConnection [74].</code>)</td>
</tr>
<tr>
<td>Timeout [77]</td>
<td>Gets the timeout (in milliseconds) (Inherited from <code>IConnection [74].</code>)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotification(String, Int32, NotificationSettings, Object) [844]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotification [866]</code> event. (Inherited from <code>IAdsNotifications [839].</code>)</td>
</tr>
<tr>
<td>AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [845]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from <code>IAdsNotifications [839].</code>)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><img src="847" alt="AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken)" /></td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the <code>AdsNotification</code> event. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><img src="848" alt="AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)" /></td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the <code>AdsNotification</code> event. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><img src="850" alt="AddDeviceNotificationEx(String, NotificationSettings, Object, Type)" /></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotificationEx</code> event. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><img src="852" alt="AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type)" /></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotificationEx</code> event. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><img src="855" alt="AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, CancellationToken)" /></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotificationEx</code> event. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><img src="853" alt="AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, CancellationToken)" /></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotificationEx</code> event. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><img src="855" alt="AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, Int32, CancellationToken)" /></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotificationExAsync</code> event. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, Int32, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>CleanupSymbolTable</td>
<td>Clears the internal symbol / DataTypes cache. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>Connect(Int32)</td>
<td>(Re)Connects the IConnection when disconnected. (Inherited from IConnection.)</td>
</tr>
<tr>
<td>Connect(AmsAddress)</td>
<td>Connects to the local target ADS Device.</td>
</tr>
<tr>
<td>Connect(String, Int32)</td>
<td>Connects to the target ADS Device.</td>
</tr>
<tr>
<td>Connect(AmsNetId, Int32)</td>
<td>Connects to the target ADS Device.</td>
</tr>
<tr>
<td>CreateVariableHandle</td>
<td>Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>CreateVariableHandleAsync</td>
<td>Determines the Symbol handle by its instance path asynchronously. (Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>DeleteDeviceNotification</td>
<td>Deletes a registered notification. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>DeleteDeviceNotificationAsync</td>
<td>Deletes a registered notification asynchronously. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>DeleteVariableHandle</td>
<td>Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>DeleteVariableHandleAsync</td>
<td>Releases the specified symbol/variable handle asynchronously. (Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>Disconnect</td>
<td>Disconnects this IConnection. (Inherited from IConnection.)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke.)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object, Any TypeSpecifier, AnyTypeSpecifier, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke.)</td>
</tr>
<tr>
<td>InvokeRpcMethodA(IAdsRpcInvoke, String, Object, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke.)</td>
</tr>
<tr>
<td>InvokeRpcMethodA(IAdsRpcInvoke, String, Object, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke.)</td>
</tr>
<tr>
<td>InvokeRpcMethodA(IAdsRpcInvoke, IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke.)</td>
</tr>
<tr>
<td>Read(UInt32, Memory)</td>
<td>(Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>Read(UInt32, UInt32, Memory)</td>
<td>(Inherited from IAdsReadWrite2.)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, Int32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, Int32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, .Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, UInt32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, UInt32, .Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, .Int32., CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, .Int32., CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAnyAsync.T.(UInt32, CancellationToken)</td>
<td>Reads data synchronously from an ADS device. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAnyAsync.T.(UInt32, .Int32., CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAnyAsync.T.(UInt32, UInt32, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadAnyString(UInt32, Int32, Encoding)</td>
<td>Reads a string from the specified symbol/variable. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAnyString(UInt32, UInt32, Int32, Encoding)</td>
<td>Reads a string from a specified address. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAnyStringAsycn(UInt32, Int32, Encoding, CancellationToken)</td>
<td>Reads a string asynchronously from the specified symbol/variable (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAnyStringAsycn(UInt32, UInt32, Int32, Encoding, CancellationToken)</td>
<td>Reads a string from a specified address asynchronously. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAsync(UInt32, Memory, Void)</td>
<td>(Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>ReadAsync(UInt32, UInt32, Memory, Void)</td>
<td>(Inherited from IAdsReadWrite.)</td>
</tr>
<tr>
<td>ReadDataType</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadDataTypeAsync</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadDeviceInfo</td>
<td>Reads the identification and version number of an ADS server. (Inherited from IAdsConnection.)</td>
</tr>
<tr>
<td>ReadDeviceInfoAsycn</td>
<td>Reads the identification and version number of an ADS server. (Inherited from IAdsConnection.)</td>
</tr>
<tr>
<td>ReadState</td>
<td>Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider.)</td>
</tr>
<tr>
<td>ReadStateAsync</td>
<td>Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider.)</td>
</tr>
<tr>
<td>ReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadSymbolAsync</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValue(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValue(String, Type)</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValue.T.(String)</td>
<td>Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>ReadValueAsync(ISymbol, CancellationToken)</code></td>
<td>Reads the value of a symbol asynchronously and returns it as an object. (Inherited from <code>IAdsSymbolicAccess</code> [937].)</td>
</tr>
<tr>
<td><code>ReadValueAsync(String, Type, CancellationToken)</code></td>
<td>Reads the value of a symbol asynchronously. (Inherited from <code>IAdsSymbolicAccess</code> [937].)</td>
</tr>
<tr>
<td><code>ReadValueAsync(T, (String, CancellationToken))</code></td>
<td>Reads the value of a symbol asynchronously. (Inherited from <code>IAdsSymbolicAccess</code> [937].)</td>
</tr>
<tr>
<td><code>ReadWrite(UInt32, Memory, Void)</code></td>
<td>(Inherited from <code>IAdsHandle</code> [827].)</td>
</tr>
<tr>
<td><code>ReadWrite(UInt32, UInt32, Memory, Void)</code></td>
<td>(Inherited from <code>IAdsReadWrite2</code> [875].)</td>
</tr>
<tr>
<td><code>ReadWriteAsync(UInt32, Memory, Void, Byte)</code></td>
<td>(Inherited from <code>IAdsHandle</code> [827].)</td>
</tr>
<tr>
<td><code>RegisterAdsStateChangedAsync</code></td>
<td>Registers for <code>AdsStateChanged</code> events as an asynchronous operation. (Inherited from <code>IAdsStateProvider</code> [929].)</td>
</tr>
<tr>
<td><code>RegisterSymbolVersionChangedAsync</code></td>
<td>Registers the symbol version changed asynchronously. (Inherited from <code>IAdsSymbolChangedProvider</code> [934].)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotification</code> event. (Inherited from <code>IAdsNotifications</code> [839].)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotification</code> event. (Inherited from <code>IAdsNotifications</code> [839].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32, UInt32) ![862]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx ![868] event. (Inherited from IAdsNotifications ![839] )</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32, UInt32) ![863]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx ![868] event. (Inherited from IAdsNotifications ![839] )</td>
</tr>
<tr>
<td>TryCreateVariableHandle ![834]</td>
<td>Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle ![827] )</td>
</tr>
<tr>
<td>TryDeleteDeviceNotification ![864]</td>
<td>Deletes a registered notification. (Inherited from IAdsNotifications ![839] )</td>
</tr>
<tr>
<td>TryDeleteVariableHandle ![835]</td>
<td>Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle ![827] )</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object) ![901]</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke ![886] )</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, AnyTypeSpecifier) ![903]</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke ![886] )</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object) ![905]</td>
<td>Invokes the rpc method. (Inherited from IAdsRpcInvoke ![886] )</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object) ![906]</td>
<td>Invokes the rpc method. (Inherited from IAdsRpcInvoke ![886] )</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryRead(UInt32, Memory, Void)</td>
<td>(Inherited from IAdsHandle [836].)</td>
</tr>
<tr>
<td>TryRead(UInt32, UInt32, Memory, Void)</td>
<td>(Inherited from IAdsReadWrite [872].)</td>
</tr>
<tr>
<td>TryReadDataType</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadState</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td>TryReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue(String, Type, Object.)</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue(ISymbol, T.)</td>
<td>(Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, Memory, Void)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, UInt32, Memory, Void, Byte)</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryWrite(UInt32, ReadOnlyMemory)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryWrite(UInt32, UInt32, ReadOnlyMemory)</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryWrite(ReadOnlyMemory)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, ReadOnlyMemory)</td>
<td>(Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>TryWriteValue(String, Object)</strong> [957]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td><strong>TryWriteValue(Symbol, Object)</strong> [959]</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td><strong>TryWriteValue.T.(String, T)</strong> [958]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td><strong>TryWriteValue.T.(ISymbol, T)</strong> [960]</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td><strong>UnregisterAdsStateChangedAsync</strong> [933]</td>
<td>Registers for AdsStateChanged [934] events as an asynchronous operation. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td><strong>UnregisterSymbolVersionChangedAsync</strong></td>
<td>Unregisters the symbol version changed asynchronous. (Inherited from IAdsSymbolChangedProvider [934].)</td>
</tr>
<tr>
<td><strong>Write(UInt32, ReadOnlyMemory)</strong> [838]</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td><strong>Write(UInt32, UInt32)</strong> [878]</td>
<td>Triggers a 'Write' call to the ADS device at the specified address. (Inherited from IAdsReadWrite2 [875].)</td>
</tr>
<tr>
<td><strong>Write(UInt32, UInt32, ReadOnlyMemory)</strong> [878]</td>
<td>(Inherited from IAdsReadWrite2 [875].)</td>
</tr>
<tr>
<td><strong>WriteAny(UInt32, Object)</strong> [726]</td>
<td>Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td><strong>WriteAny(UInt32, Object, Int32)</strong> [727]</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td><strong>WriteAny(UInt32, UInt32, Object)</strong> [728]</td>
<td>Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td><strong>WriteAny(UInt32, UInt32, Object, Int32)</strong> [728]</td>
<td>Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td><strong>WriteAnyAsync(UInt32, Object, CancellationToken)</strong> [730]</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, Int32, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, Int32, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAsync(UInt32, ReadOnlyMemory, Void)</td>
<td>Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void)</td>
<td>Changes the ADS status and device status of the ADS server asynchronously. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteControl(StatesInfo)</td>
<td>Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteControl(StatesInfo, ReadOnlyMemory)</td>
<td>Changes the ADS status and device status of the ADS server asynchronously. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, CancellationToken)</td>
<td>Changes the ADS status and the device status of the ADS server asynchronously. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValue(ISymbol, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValue.T.(String, T)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>WriteValue.T.([ISymbol, T])</code></td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from <code>IAdsSymbolicAccess</code>.)</td>
</tr>
<tr>
<td><code>WriteValueAsync([ISymbol, Object, CancellationToken])</code></td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from <code>IAdsSymbolicAccess</code>.)</td>
</tr>
<tr>
<td><code>WriteValueAsync.T.([String, T, CancellationToken])</code></td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from <code>IAdsSymbolicAccess</code>.)</td>
</tr>
<tr>
<td><code>WriteValueAsync.T.([ISymbol, T, CancellationToken])</code></td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from <code>IAdsSymbolicAccess</code>.)</td>
</tr>
</tbody>
</table>

**Events**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>AdsNotification</code></td>
<td>Occurs when the ADS device sends a notification to the client. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><code>AdsNotificationError</code></td>
<td>Occurs when an exception has occurred during notification management. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><code>AdsNotificationEx</code></td>
<td>Occurs when the ADS device sends a notification to the client. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><code>AdsStateChanged</code></td>
<td>Occurs when the AdsState of the target system has been changed. (Inherited from <code>IAdsStateProvider</code>.)</td>
</tr>
<tr>
<td><code>AdsSumNotification</code></td>
<td>Occurs when Notifications are send (bundled notifications) (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><code>AdsSymbolVersionChanged</code></td>
<td>Occurs when the ADS Symbol Version changed. (Inherited from <code>IAdsSymbolChangedProvider</code>.)</td>
</tr>
<tr>
<td><code>ConnectionStateChanged</code></td>
<td>Occurs when connection status of the <code>IConnectionStateProvider</code> has been changed. (Inherited from <code>IConnectionStateProvider</code>.)</td>
</tr>
</tbody>
</table>

**Extension Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PollAdsState([Observevle.Unit.])</code></td>
<td>Overloaded. Gets an observable sequence of <code>AdsState</code> via Polling. (Defined by <code>AdsClientExtensions</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollAdsState(TimeSpan)</td>
<td>Overloaded. Gets an observable sequence of AdsState's via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollAdsStateAsync(TimeSpan, CancellationToken)</td>
<td>Overloaded. Gets an observable sequence of AdsState's via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, I.Observable.Unit.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32., TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, I.Observable.Unit., Func.Exception, Object.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan, Func.Exception, Object.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32., I.Observable.Unit., Func.Exception, Object.)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, I.Observable.Unit.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PollValues.T.(String, IObservable.Unit., Func.Exception, T.) [1087]</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, TimeSpan, Func.Exception, T.) [1088]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, Int32., IObservable.Unit.) [1085]</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, Int32., TimeSpan) [1086]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, Int32., IObservable.Unit., Func.Exception, T.) [1091]</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, Int32., TimeSpan, Func.Exception, T.) [1092]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>WhenAdsStateChanges [1066]</td>
<td>Gets an observable sequence of AdsState [626]s. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol) [1068]</td>
<td>Overloaded. Gets an observable sequence of Notification [974]s. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection) [1069]</td>
<td>Overloaded. Gets an observable sequence of Notification [974] objects. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol, NotificationSettings) [1104] [1070]</td>
<td>Overloaded. Gets an observable sequence of SymbolValueNotification [1104]s. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection, NotificationSettings) [1071]</td>
<td>Overloaded. Gets an observable sequence of Notification [974] objects. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(String, Type, NotificationSettings) [1099]</td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhenNotification.T. (String, NotificationSettings)</td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>WhenSymbolVersionChanges</td>
<td>Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenSymbolVersionChanges(ISchedule)</td>
<td>Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenValueChanged</td>
<td>Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td>WriteValues.T. (String, IObservable.T.)</td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>WriteValues.T. (String, IObservable.T., Action.Exception.)</td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.Ads Namespace [151]

### 6.2.31.1 IAdsConnectAddress Properties

The IAdsConnectAddress [733] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address [781]</td>
<td>Gets the AmsAddress [648] of the ADS server. (Inherited from IAdsConnection [765].)</td>
</tr>
<tr>
<td>ClientAddress [781]</td>
<td>Get the AmsAddress [648] of the ADS client. (Inherited from IAdsConnection [765].)</td>
</tr>
<tr>
<td>ConnectionState [85]</td>
<td>Gets the current Connection state of the IConnectionStateProvider [84]. (Inherited from IConnectionStateProvider [84].)</td>
</tr>
<tr>
<td>DefaultValueEncoding [76]</td>
<td>Gets the default value encoding. (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>Id [76]</td>
<td>Gets the Connection Identifier . (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsConnected</td>
<td>Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available. (Inherited from IConnection.)</td>
</tr>
<tr>
<td>IsLocal</td>
<td>Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer. (Inherited from IAdsConnection.)</td>
</tr>
<tr>
<td>Session</td>
<td>Gets the session that initiated this IConnection (Inherited from IConnection.)</td>
</tr>
<tr>
<td>Timeout</td>
<td>Gets the timeout (in milliseconds) (Inherited from IConnection.)</td>
</tr>
</tbody>
</table>

**Reference**

IAdsConnectAddress Interface

TwinCAT.Ads Namespace

### 6.2.31.2  IAdsConnectAddress Methods

The IAdsConnectAddress type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotification(String, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from IAdsNotifications[839].)</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type, _Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from IAdsNotifications[839].)</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type. (Inherited from IAdsNotifications[839].)</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, _Int32, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from IAdsNotifications[839].)</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, _Int32, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from IAdsNotifications[839].)</td>
</tr>
<tr>
<td>CleanupSymbolTable</td>
<td>Clears the internal symbol / DataTypes cache. (Inherited from IAdsSymbolicAccess[937].)</td>
</tr>
<tr>
<td>Close</td>
<td>Closes this IConnection[74] (Inherited from IConnection[74].)</td>
</tr>
<tr>
<td>Connect</td>
<td>(Re)Connects the IConnection[74] when disconnected. (Inherited from IConnection[74].)</td>
</tr>
<tr>
<td>Connect(UInt32)</td>
<td>Connects to the local target ADS Device.</td>
</tr>
<tr>
<td>Connect(AmsAddress)</td>
<td>Connects the target ADS Device.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Connect(String, Int32)</td>
<td>Connects to the target ADS Device.</td>
</tr>
<tr>
<td>Connect(AmsNetId, Int32)</td>
<td>Connects to the target ADS Device.</td>
</tr>
<tr>
<td>CreateVariableHandle</td>
<td>Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>CreateVariableHandleAsync</td>
<td>Determines the Symbol handle by its instance path asynchronously. (Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>DeleteDeviceNotification</td>
<td>Deletes a registered notification. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>DeleteDeviceNotificationAsync</td>
<td>Deletes a registered notification asynchronously. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>DeleteVariableHandle</td>
<td>Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>DeleteVariableHandleAsync</td>
<td>Releases the specified symbol/variable handle asynchronously. (Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>Disconnect</td>
<td>Disconnects this IConnection. (Inherited from IConnection.)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke.)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke.)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object, Any TypeSpecifier, AnyTypeSpecifier, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke.)</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, String, Object, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>InvokeRpcMethodA sync(String, String, Object, Any TypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from AdsRpcInvoke)</td>
</tr>
<tr>
<td>InvokeRpcMethodA sync(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from AdsRpcInvoke)</td>
</tr>
<tr>
<td>Read(UInt32, Memory)</td>
<td>(Inherited from AdsHandle)</td>
</tr>
<tr>
<td>Read(UInt32, UInt32, Memory)</td>
<td>(Inherited from AdsReadWrite2)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, Int32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess)</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess)</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Int32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, .Int32., CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, .Int32., CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, .Int32., CancellationToken)</td>
<td>Reads data synchronously from an ADS device. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, .Int32., CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, .Int32., .Int32., CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Int32, Encoding)</td>
<td>Reads a string from the specified symbol/variable. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>ReadAnyString(UInt32, UInt32, Int32, Encoding)</td>
<td>Reads a string from a specified address. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken)</td>
<td>Reads a string asynchronously from the specified symbol/variable. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadAnyStringAsyn&lt;br /&gt;(UInt32, UInt32, Int32, Encoding, CancellationToken)</td>
<td>Reads a string from a specified address asynchronously. (Inherited from AdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAsync&lt;br /&gt;(UInt32, Memory, Void)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>ReadAsync&lt;br /&gt;(UInt32, UInt32, Memory, Void)</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>ReadDataType</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadDataTypeAsync</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadDeviceInfo</td>
<td>Reads the identification and version number of an ADS server. (Inherited from IAdsConnection [765].)</td>
</tr>
<tr>
<td>ReadDeviceInfoAsync</td>
<td>Reads the identification and version number of an ADS server. (Inherited from IAdsConnection [765].)</td>
</tr>
<tr>
<td>ReadState</td>
<td>Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td>ReadStateAsync</td>
<td>Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td>ReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadSymbolAsync</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValue(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValue(String, Type)</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValue.T.(String)</td>
<td>Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValueAsync(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously and returns it as an object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValueAsync(String, Type, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValueAsync.T.(String, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>ReadWrite(UInt32, Memory, Void)</code></td>
<td>(Inherited from <code>IAdsHandle</code>.)</td>
</tr>
<tr>
<td><code>ReadWrite(UInt32, UInt32, Memory, Void)</code></td>
<td>(Inherited from <code>IAdsReadWrite2</code>.)</td>
</tr>
<tr>
<td><code>ReadWriteAsync(UInt32, Memory, Void)</code></td>
<td>(Inherited from <code>IAdsHandle</code>.)</td>
</tr>
<tr>
<td><code>ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte)</code></td>
<td>(Inherited from <code>IAdsReadWrite</code>.)</td>
</tr>
<tr>
<td><code>RegisterAdsStateChangedAsync</code></td>
<td>Registers for <code>AdsStateChanged</code> events as an asynchronous operation. (Inherited from <code>IAdsStateProvider</code>.)</td>
</tr>
<tr>
<td><code>RegisterSymbolVersionChangedAsync</code></td>
<td>Registers the symbol version changed asynchronously. (Inherited from <code>IAdsSymbolChangedProvider</code>.)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotification</code> event. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotification</code> event. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, _Int32., UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotificationEx</code> event. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, _Int32., UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotificationEx</code> event. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><code>TryCreateVariableHandle</code></td>
<td>Determines the Symbol handle by its instance path synchronously. (Inherited from <code>IAdsHandle</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryDeleteDeviceNotification</td>
<td>Deletes a registered notification. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>TryDeleteVariableHandle</td>
<td>Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod</td>
<td>Invokes the rpc method. (Inherited from IAdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod</td>
<td>Invokes the rpc method. (Inherited from IAdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>TryRead(UInt32, Memory, Void)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryRead(UInt32, UInt32, Memory, Void)</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryReadDataType</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadState</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td>TryReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue(IrSym bol, Object)</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryReadValue(String, Type, Object)</td>
<td>Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue.T.(String, T.)</td>
<td>Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue.T.(ISymbol, T.)</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, Memory, Void, Byte)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, UInt32, Memory, Void, Byte)</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryWrite(UInt32, ReadOnlyMemory)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryWrite(UInt32, UInt32, ReadOnlyMemory)</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, ReadOnlyMemory)</td>
<td>(Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>TryWriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue(ISymbol, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue.T.(String, T.)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue.T.(ISymbol, T.)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>UnregisterAdsStateChangedAsync</td>
<td>Registers for AdsStateChanged [934] events as an asynchronous operation. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UnregisterSymbolVersionChangedAsync</td>
<td>Unregisters the symbol version changed asynchronous. (Inherited from IAdsSymbolChangedProvider [934].)</td>
</tr>
<tr>
<td>Write(UInt32, ReadOnlyMemory)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>Write(UInt32, UInt32, ReadOnlyMemory)</td>
<td>Triggers a 'Write' call to the ADS device at the specified address. (Inherited from IAdsReadWrite2 [875].)</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object, Int32)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAny(UInt32, UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, CancellationToken)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, Int32, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, Int32, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAsync(UInt32, ReadOnlyMemory, Void)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void)</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>WriteControl(StateI nfo)</td>
<td>Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteControl(StateI nfo, ReadOnlyMemory)</td>
<td>(Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, CancellationToken)</td>
<td>Changes the ADS status and device status of the ADS server asynchronously. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void)</td>
<td>(Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValue(Symbol, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValue.T.(String, T)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValue.T.(ISymbol, T)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValueAsync(ISymbol, Object, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValueAsync.T.(String, T, CancellationToken)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValueAsync.T.(ISymbol, T, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
</tbody>
</table>
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollAdsState(IObservable&lt;Unit.&gt;)</td>
<td>Overloaded. Gets an observable sequence of AdsState via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollAdsState(TimeSpan)</td>
<td>Overloaded. Gets an observable sequence of AdsState via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollAdsStateAsync(IObservable&lt;Unit., CancellationToken&gt;)</td>
<td>Overloaded. Gets an observable sequence of AdsState via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollAdsStateAsync(TimeSpan, CancellationToken)</td>
<td>Overloaded. Gets an observable sequence of AdsState via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, IObservable&lt;Unit.&gt;)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32., TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32., IObservable&lt;Unit., Func.Exception, Object.&gt;)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32., TimeSpan, Func.Exception, Object.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32., IObservable&lt;Unit., Func.Exception, Object.&gt;)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32., TimeSpan, Func.Exception, Object.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, IObservable&lt;Unit.&gt;)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollValues.T.(String, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, IObserver.Unit, Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, TimeSpan, Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, .Int32., IObserver.Unit.)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, .Int32., TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, .Int32., IObserver.Unit, Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, .Int32., TimeSpan, Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>WhenAdsStateChangess</td>
<td>Gets an observable sequence of AdsState.s. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol)</td>
<td>Overloaded. Gets an observable sequence of Notification.s. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection)</td>
<td>Overloaded. Gets an observable sequence of Notification objects. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol, NotificationSettings)</td>
<td>Overloaded. Gets an observable sequence of SymbolValueNotification objects. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection, NotificationSettings)</td>
<td>Overloaded. Gets an observable sequence of Notification objects. (Defined by AdsClientExtensions.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WhenNotification(String, Type, NotificationSettings) [1099]</strong></td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td><strong>WhenNotification.T(String, NotificationSettings) [1098]</strong></td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td><strong>WhenSymbolVersionChanges [1073]</strong></td>
<td>Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td><strong>WhenSymbolVersionChanges(IScheduler) [1074]</strong></td>
<td>Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td><strong>WhenValueChanged [1114]</strong></td>
<td>Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td><strong>WriteValues.T(String, IObservable.T.) [1101]</strong></td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td><strong>WriteValues.T(String, IObservable.T., Action.Exception.) [1102]</strong></td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
</tbody>
</table>

**Reference**

IAdsConnectAddress Interface [733]

TwinCAT.Ads Namespace [151]

### 6.2.31.2.1 IAdsConnectAddress.Connect Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connect [78]</strong></td>
<td>(Re)Connects the IConnection [74] when disconnected. (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td><strong>Connect(Int32) [763]</strong></td>
<td>Connects to the local target ADS Device.</td>
</tr>
<tr>
<td><strong>Connect(AmsAddress) [763]</strong></td>
<td>Connects the target ADS Device.</td>
</tr>
<tr>
<td><strong>Connect(String, Int32) [764]</strong></td>
<td>Connects to the target ADS Device.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>Connect(AmsNetId, Int32)</code></td>
<td>Connects to the target ADS Device.</td>
</tr>
</tbody>
</table>

**Reference**

IAdsConnectAddress Interface [733]

TwinCAT.Ads Namespace [151]

**IAdsConnectAddress.Connect Method (Int32)**

Connects to the local target ADS Device.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fredca3e72bc0e15da1c14

**Syntax**

**C#**

```csharp
void Connect(
    int port
)
```

**Parameters**

- **port**
  - Type: System.Int32
  - The port number of the local ADS target device to connect to.

**Reference**

IAdsConnectAddress Interface [733]

Connect Overload [762]

TwinCAT.Ads Namespace [151]

**IAdsConnectAddress.Connect Method (AmsAddress)**

Connects the target ADS Device.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fredca3e72bc0e15da1c14

**Syntax**

**C#**

```csharp
void Connect(
    AmsAddress address
)
```

**Parameters**

- **address**
  - Type: TwinCAT.Ads.AmsAddress [648]
  - The address of the target device.
IAdsConnectAddress.Connect Method (String, Int32)

Connects to the target ADS Device.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
void Connect(
    string netId,
    int port
)
```

Parameters

**netId**
Type: `System.String`
The `AmsNetId` of the ADS target device specified as string.

**port**
Type: `System.Int32`
The port number of the ADS target device.

IAdsConnectAddress.Connect Method (AmsNetId, Int32)

Connects to the target ADS Device.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
void Connect(
    AmsNetId netId,
    int port
)
```

Parameters

**netId**
Type: `TwinCAT.Ads.AmsNetId`
The `AmsNetId` of the target device.
port Type: System.Int32
The Ams Port number on the target device to connect to.

Reference
IAdsConnectAddress Interface [733]
Connect Overload[762]
TwinCAT.Ads Namespace [151]

6.2.31.3 IAdsConnectAddress Events
The IAdsConnectAddress [733] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsNotification ❯ 866</td>
<td>Occurs when the ADS device sends a notification to the client. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AdsNotificationError ❯ 867</td>
<td>Occurs when a exception has occurred during notification management. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AdsNotificationEx ❯ 868</td>
<td>Occurs when the ADS devices sends a notification to the client. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AdsStateChanged ❯ 934</td>
<td>Occurs when the AdsState of the target system has been changed. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td>AdsSumNotification ❯ 868</td>
<td>Occurs when Notifications are send (bundled notifications) (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AdsSymbolVersionChanged ❯ 937</td>
<td>Occurs when the ADS Symbol Version changed. (Inherited from IAdsSymbolChangedProvider [934].)</td>
</tr>
<tr>
<td>ConnectionStateChanged ❯ 86</td>
<td>Occurs when connection status of the IConnectionStateProvider [84] has been changed. (Inherited from IConnectionStateProvider [84].)</td>
</tr>
</tbody>
</table>

Reference
IAdsConnectAddress Interface [733]
TwinCAT.Ads Namespace [151]

6.2.32 IAdsConnection Interface
ADS Connection interface

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
The IAdsConnection type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address [781]</td>
<td>Gets the AmsAddress [648] of the ADS server.</td>
</tr>
<tr>
<td>ClientAddress [781]</td>
<td>Get the AmsAddress [648] of the ADS client.</td>
</tr>
<tr>
<td>ConnectionState [85]</td>
<td>Gets the current Connection state of the IConnectionStateProvider [84]</td>
</tr>
<tr>
<td></td>
<td>(Inherited from IConnectionStateProvider [84].)</td>
</tr>
<tr>
<td>DefaultValueEncoding [76]</td>
<td>Gets the default value encoding. (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>Id [76]</td>
<td>Gets the Connection Identifier. (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>IsConnected [76]</td>
<td>Gets a value indicating whether the local ADS port was opened successfully.</td>
</tr>
<tr>
<td></td>
<td>It does not indicate if the target port is available. Use the method</td>
</tr>
<tr>
<td></td>
<td>ReadState to determine if the target port is available. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>IConnection [74].)</td>
</tr>
<tr>
<td>IsLocal [782]</td>
<td>Gets a value indicating whether the ADS client is connected to a ADS Server</td>
</tr>
<tr>
<td></td>
<td>on the local computer.</td>
</tr>
<tr>
<td>Session [77]</td>
<td>Gets the session that initiated this IConnection [74]</td>
</tr>
<tr>
<td></td>
<td>(Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>Timeout [77]</td>
<td>Gets the timeout (in milliseconds) (Inherited from IConnection [74].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotification(String, Int32, NotificationSettings, Object) [844]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [866] event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [845]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [856]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>CleanupSymbolTable [942]</td>
<td>Clears the internal symbol / DataTypes cache. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>Close [78]</td>
<td>Closes this IConnection [74] (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>Connect [78]</td>
<td>(Re)Connects the IConnection [74] when disconnected. (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>CreateVariableHandle [829]</td>
<td>Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>CreateVariableHandleAsync [830]</td>
<td>Determines the Symbol handle by its instance path asynchronously. (Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>DeleteDeviceNotification [857]</td>
<td>Deletes a registered notification. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>DeleteDeviceNotificationAsync [858]</td>
<td>Deletes a registered notification asynchronously. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>DeleteVariableHandle [830]</td>
<td>Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>DeleteVariableHandleAsync [831]</td>
<td>Releases the specified symbol/variable handle asynchronously. (Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>Disconnect [79]</td>
<td>Disconnects this IConnection [74]. (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, .Object.) [890]</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object...) [891]</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Any TypeSpecifier, Any TypeSpecifier, Object...) [893]</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>InvokeRpcMethodA sync(String, String, Object, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from AdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>InvokeRpcMethodA sync(String, String, Object, Any TypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from AdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>InvokeRpcMethodA sync(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from AdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>Read(UInt32, Memory)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>Read(UInt32, UInt32, Memory)</td>
<td>(Inherited from IAdsReadWrite2 [875].)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, UInt32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, UInt32, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Int32, Type, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, <em>Int32</em>, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, <em>Int32</em>, Type, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, <em>Int32</em>, <em>Int32</em>, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, <em>Int32</em>, <em>Int32</em>, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, <em>Int32</em>, <em>Int32</em>, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(T. (UInt32, CancellationToken)</td>
<td>Reads data synchronously from an ADS device. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(T. (UInt32, <em>Int32</em>, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(T. (UInt32, <em>Int32</em>, <em>Int32</em>, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(T. (UInt32, <em>Int32</em>, <em>Int32</em>, <em>Int32</em>, CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadString(UInt32, Int32, Encoding)</td>
<td>Reads a string from the specified symbol/variable. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadString(UInt32, UInt32, Int32, Encoding)</td>
<td>Reads as string from a specified address. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken)</td>
<td>Reads a string asynchronously from the specified symbol/variable. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadAnyStringAsync(UI32, UI32, Int32, Encoding, CancellationToken) [725]</td>
<td>Reads a string from a specified address asynchronously. (Inherited from AdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAsync(UI32, Memory, Void) [833]</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>ReadAsync(UI32, UI32, Memory, Void) [871]</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>ReadDataType [942]</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadDataTypeAsync [943]</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadDeviceInfo [795]</td>
<td>Reads the identification and version number of an ADS server.</td>
</tr>
<tr>
<td>ReadDeviceInfoAsync [796]</td>
<td>Reads the identification and version number of an ADS server.</td>
</tr>
<tr>
<td>ReadState [931]</td>
<td>Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td>ReadStateAsync [931]</td>
<td>Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td>ReadSymbol [943]</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadSymbolAsync [944]</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValue(ISymbol) [946]</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValue(String, Type) [947]</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValue.T.(String) [945]</td>
<td>Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValueAsync(ISymbol, CancellationToken) [949]</td>
<td>Reads the value of a symbol asynchronously and returns it as an object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValueAsync(String, Type, CancellationToken) [951]</td>
<td>Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValueAsync.T.(String, CancellationToken) [948]</td>
<td>Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>ReadWrite(UInt32, Memory, Void)</code></td>
<td>(Inherited from <code>IAdsHandle [827]</code>.)</td>
</tr>
<tr>
<td><code>ReadWrite(UInt32, UInt32, Memory, Void)</code></td>
<td>(Inherited from <code>IAdsReadWrite2 [875]</code>.)</td>
</tr>
<tr>
<td><code>ReadWriteAsync(UInt32, Memory, Void)</code></td>
<td>(Inherited from <code>IAdsHandle [827]</code>.)</td>
</tr>
<tr>
<td><code>ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte)</code></td>
<td>(Inherited from <code>IAdsReadWrite [870]</code>.)</td>
</tr>
<tr>
<td><code>RegisterAdsStateChangedAsync</code></td>
<td>Registers for AdsStateChanged [934] events as an asynchronous operation. (Inherited from <code>IAdsStateProvider [929]</code>.)</td>
</tr>
<tr>
<td><code>RegisterSymbolVersionChangedAsync</code></td>
<td>Registers the symbol version changed asynchronously. (Inherited from <code>IAdsSymbolChangedProvider [934]</code>.)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [866] event. (Inherited from <code>IAdsNotifications [839]</code>.)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [866] event. (Inherited from <code>IAdsNotifications [839]</code>.)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, _Int32., UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from <code>IAdsNotificationsEx [839]</code>.)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, _Int32., UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from <code>IAdsNotificationsEx [839]</code>.)</td>
</tr>
<tr>
<td><code>TryCreateVariableHandle</code></td>
<td>Determines the Symbol handle by its instance path synchronously. (Inherited from <code>IAdsHandle [827]</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryDeleteDeviceNotification</td>
<td>Deletes a registered notification. (Inherited from IAdsNotifications[839])</td>
</tr>
<tr>
<td>TryDeleteVariableHandle</td>
<td>Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle[827])</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke[886])</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke[886])</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object)</td>
<td>Invokes the rpc method. (Inherited from IAdsRpcInvoke[886])</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object)</td>
<td>Invokes the rpc method. (Inherited from IAdsRpcInvoke[886])</td>
</tr>
<tr>
<td>TryRead(UInt32, Memory, Void)</td>
<td>(Inherited from IAdsHandle[827])</td>
</tr>
<tr>
<td>TryRead(UInt32, UInt32, Memory, Void)</td>
<td>(Inherited from IAdsReadWrite[870])</td>
</tr>
<tr>
<td>TryReadDataType</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess[937])</td>
</tr>
<tr>
<td>TryReadState</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful. (Inherited from IAdsStateProvider[929])</td>
</tr>
<tr>
<td>TryReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess[937])</td>
</tr>
<tr>
<td>TryReadValue(ISymbol, Object)</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess[937])</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryReadValue(String, Type, Object)</td>
<td>Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue.T.(String, T.)</td>
<td>Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue.T.(ISymbol, T.)</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, Memory, Void, Byte)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, UInt32, Memory, Void, Byte)</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryWrite(UInt32, ReadOnlyMemory)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryWrite(UInt32, UInt32, ReadOnlyMemory)</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, ReadOnlyMemory)</td>
<td>(Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>TryWriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue.ISymbol, Object]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue.T.(String, T.)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue.T.(ISymbol, T.)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>UnregisterAdsStateChangedAsync</td>
<td>Registers for AdsStateChanged [934] events as an asynchronous operation. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>UnregisterSymbolVersionChangedAsync</strong> ([936])</td>
<td>Unregisters the symbol version changed asynchronous. (Inherited from AdsSymbolChangedProvider.)</td>
</tr>
<tr>
<td><strong>Write(UInt32, ReadOnlyMemory)</strong> ([838])</td>
<td>(Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td><strong>Write(UInt32, UInt32)</strong> ([878])</td>
<td>Triggers a ‘Write’ call to the ADS device at the specified address. (Inherited from IAdsReadWrite2.)</td>
</tr>
<tr>
<td><strong>WriteAny(UInt32, Object)</strong> ([726])</td>
<td>Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td><strong>WriteAny(UInt32, Object, .Int32.)</strong> ([727])</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td><strong>WriteAny(UInt32, UInt32, Object)</strong> ([728])</td>
<td>Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td><strong>WriteAnyAsync(UInt32, Object, CancellationToken)</strong> ([730])</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td><strong>WriteAnyAsync(UInt32, Object, .Int32., CancellationToken)</strong> ([731])</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td><strong>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)</strong> ([732])</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td><strong>WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken)</strong> ([732])</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td><strong>WriteAsync(UInt32, ReadOnlyMemory, Void)</strong> ([838])</td>
<td>(Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void)</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>WriteControl(Stateload)</td>
<td>Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteControl(Stateload, ReadOnlyMemory)</td>
<td>(Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, CancellationToken)</td>
<td>Changes the ADS status and device status of the ADS server asynchronously. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void)</td>
<td>(Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValue(ISymbol, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValue.T.(String, T)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValue.T.(ISymbol, T)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValueAsync(ISymbol, Object, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValueAsync.T.(String, T, CancellationToken)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValueAsync.T.(ISymbol, T, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
</tbody>
</table>
Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsNotification</td>
<td>Occurs when the ADS device sends a notification to the client. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AdsNotificationError</td>
<td>Occurs when an exception has occurred during notification management. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AdsNotificationEx</td>
<td>Occurs when the ADS device sends a notification to the client. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AdsStateChanged</td>
<td>Occurs when the AdsState of the target system has been changed. (Inherited from IAdsStateProvider.)</td>
</tr>
<tr>
<td>AdsSumNotification</td>
<td>Occurs when Notifications are send (bundled notifications) (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AdsSymbolVersionChanged</td>
<td>Occurs when the ADS Symbol Version changed. (Inherited from IAdsSymbolChangedProvider.)</td>
</tr>
<tr>
<td>ConnectionStateChanged</td>
<td>Occurs when connection status of the IConnectionStateProvider has been changed. (Inherited from IConnectionStateProvider.)</td>
</tr>
</tbody>
</table>

Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollAdsState(I Observable.Unit.)</td>
<td>Overloaded. Gets an observable sequence of AdsState via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollAdsState(TimeSpan)</td>
<td>Overloaded. Gets an observable sequence of AdsState via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollAdsStateAsync(I Observable.Unit, CancellationToken)</td>
<td>Overloaded. Gets an observable sequence of AdsState via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollAdsStateAsync(TimeSpan, CancellationToken)</td>
<td>Overloaded. Gets an observable sequence of AdsState via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, I Observable.Unit.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, IObservable.Unit, Func.Exception, Object)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan, Func.Exception, Object)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, IObservable.Unit, Func.Exception, Object)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, IObservable.Unit.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, IObservable.Unit., Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T.(String, TimeSpan, Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T. (String, Int32, IObservable.Unit.)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T. (String, Int32., TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues.T. (String, Int32., IObservable.Unit., Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollValues.T. (String, Int32, TimeSpan, Func.Exception, T.) [1092]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>WhenAdsStateChanges [1066]</td>
<td>Gets an observable sequence of AdsState [626]s. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol) [1068]</td>
<td>Overloaded. Gets an observable sequence of Notification [974]s. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection) [1069]</td>
<td>Overloaded. Gets an observable sequence of Notification [974] objects. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol, NotificationSettings) [1070]</td>
<td>Overloaded. Gets an observable sequence of Notification [974] objects. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenSymbolVersionChanges(ISchedule) [1071]</td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>WhenSymbolVersionChanges(ISchedule) [1074]</td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>WriteValues.T. (String, IObservable.T.) [1101]</td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValues.T. (String, IObservable.T., Action.Exception)</td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
</tbody>
</table>

#### Remarks

The ADS connection interface represents the logical point-to-point exception between Client and Server within an ADS Session.

#### Reference

- TwinCAT.Ads Namespace [151]
- TwinCAT.IConnection [74]
- TwinCAT.Ads.IAdsNotifications [839]
- TwinCAT.Ads.IAdsAnyAccess [702]
- TwinCAT.Ads.IAdsHandle [827]
- TwinCAT.Ads.IAdsReadWrite2 [875]
- TwinCAT.Ads.IAdsStateControl [915]

#### 6.2.32.1 IAdsConnection Properties

The IAdsConnection [765] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address [781]</td>
<td>Gets the AmsAddress [648] of the ADS server.</td>
</tr>
<tr>
<td>ClientAddress [781]</td>
<td>Get the AmsAddress [648] of the ADS client.</td>
</tr>
<tr>
<td>ConnectionState [85]</td>
<td>Gets the current Connection state of the IConnectionStateProvider [84] (Inherited from IConnectionStateProvider [84].)</td>
</tr>
<tr>
<td>DefaultValueEncoding [76]</td>
<td>Gets the default value encoding. (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>Id [76]</td>
<td>Gets the Connection Identifier. (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>IsConnected [76]</td>
<td>Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available. (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>IsLocal [782]</td>
<td>Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.</td>
</tr>
<tr>
<td>Session [77]</td>
<td>Gets the session that initiated this IConnection [74] (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Timeout</td>
<td>Gets the timeout (in milliseconds) (Inherited from IConnection).</td>
</tr>
</tbody>
</table>

**Reference**

IAdsConnection Interface [765]

TwinCAT.Ads Namespace [151]

### 6.2.32.1.1 IAdsConnection.Address Property

Gets the `AmsAddress` of the ADS server.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
AmsAddress Address { get; }
```

**Property Value**

Type: `AmsAddress` [648]

**Reference**

IAdsConnection Interface [765]

TwinCAT.Ads Namespace [151]

### 6.2.32.1.2 IAdsConnection.ClientAddress Property

Get the `AmsAddress` of the ADS client.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
AmsAddress ClientAddress { get; }
```

**Property Value**

Type: `AmsAddress` [648]

**Reference**

IAdsConnection Interface [765]

TwinCAT.Ads Namespace [151]
6.2.32.1.3  IAdsConnection.IsLocal Property

Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.

**Namespace:**  TwinCAT.Ads

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
bool IsLocal { get; }
```

**Property Value**

Type:  Boolean

**Reference**

IAdsConnection Interface

TwinCAT.Ads Namespace

6.2.32.2  IAdsConnection Methods

The IAdsConnection type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![icon]  AddDeviceNotification(String, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>![icon]  AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>![icon]  AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, Int32., CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, Int32., CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>CleanupSymbolTable</td>
<td>Clears the internal symbol / DataTypes cache. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>Close</td>
<td>Closes this IConnection. (Inherited from IConnection.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Connect</strong></td>
<td>(Re)Connects the <strong>IConnection</strong> when disconnected. (Inherited from <strong>IConnection</strong>.)</td>
</tr>
<tr>
<td><strong>CreateVariableHandle</strong></td>
<td>Determines the Symbol handle by its instance path synchronously. (Inherited from <strong>IAdsHandle</strong>.)</td>
</tr>
<tr>
<td><strong>CreateVariableHandleAsync</strong></td>
<td>Determines the Symbol handle by its instance path asynchronously. (Inherited from <strong>IAdsHandle</strong>.)</td>
</tr>
<tr>
<td><strong>DeleteDeviceNotification</strong></td>
<td>Deletes a registered notification. (Inherited from <strong>IAdsNotifications</strong>.)</td>
</tr>
<tr>
<td><strong>DeleteDeviceNotificationAsync</strong></td>
<td>Deletes a registered notification asynchronously. (Inherited from <strong>IAdsNotifications</strong>.)</td>
</tr>
<tr>
<td><strong>DeleteVariableHandle</strong></td>
<td>Releases the specified symbol/variable handle synchronously. (Inherited from <strong>IAdsHandle</strong>.)</td>
</tr>
<tr>
<td><strong>DeleteVariableHandleAsync</strong></td>
<td>Releases the specified symbol/variable handle asynchronously. (Inherited from <strong>IAdsHandle</strong>.)</td>
</tr>
<tr>
<td><strong>Disconnect</strong></td>
<td>Disconnects this <strong>IConnection</strong>. (Inherited from <strong>IConnection</strong>.)</td>
</tr>
<tr>
<td><strong>InvokeRpcMethod</strong></td>
<td>Invokes the specified RPC Method. (Inherited from <strong>IAdsRpcInvoke</strong>.)</td>
</tr>
<tr>
<td><strong>InvokeRpcMethodAsync</strong></td>
<td>Invokes the specified RPC Method asynchronously. (Inherited from <strong>IAdsRpcInvoke</strong>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>InvokeRpcMethodA</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from AdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>InvokeRpcMethodA</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from AdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>Read(UInt32, Memory)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>Read(UInt32, UInt32, Memory)</td>
<td>(Inherited from IAdsReadWrite2 [875].)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, Int32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type, Int32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, .Int32., CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken)</td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, .Int32., CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, .Int32., CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, .Int32., CancellationToken)</td>
<td>Reads data synchronously from an ADS device. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, .Int32., CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, .Int32., CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, .Int32., CancellationToken)</td>
<td>Reads a string asynchronously from the specified symbol/variable. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, .Int32., CancellationToken)</td>
<td>Reads as string from a specified address. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, .Int32., CancellationToken)</td>
<td>Reads a string asynchronously from the specified symbol/variable. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadAnyStringAsynC(UInt32, UInt32, Int32, Encoding, CancellationToken) [725]</td>
<td>Reads a string from a specified address asynchronously. (Inherited from AdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAsync(UInt32, Memory, Void) [833]</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>ReadAsync(UInt32, UInt32, Memory, Void) [871]</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>ReadDataType [942]</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadDataTypeAsync [943]</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadDeviceInfo [795]</td>
<td>Reads the identification and version number of an ADS server.</td>
</tr>
<tr>
<td>ReadDeviceInfoAsync [796]</td>
<td>Reads the identification and version number of an ADS server.</td>
</tr>
<tr>
<td>ReadState [931]</td>
<td>Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td>ReadStateAsync [931]</td>
<td>Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td>ReadSymbol [943]</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadSymbolAsync [944]</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValue(ISymbol) [946]</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValue(String, Type) [947]</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValue.T.(String) [945]</td>
<td>Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValueAsync(ISymbol, CancellationToken) [949]</td>
<td>Reads the value of a symbol asynchronously and returns it as an object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValueAsync(String, Type, CancellationToken) [951]</td>
<td>Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>ReadValueAsync.T.(String, CancellationToken) [948]</td>
<td>Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>ReadWrite(UInt32, Memory, Void)</code></td>
<td>(Inherited from <code>IAdsHandle [827].</code>)</td>
</tr>
<tr>
<td><code>ReadWrite(UInt32, UInt32, Memory, Void)</code></td>
<td>(Inherited from <code>IAdsReadWrite2 [875].</code>)</td>
</tr>
<tr>
<td><code>ReadWriteAsync(UInt32, Memory, Void)</code></td>
<td>(Inherited from <code>IAdsHandle [827].</code>)</td>
</tr>
<tr>
<td><code>ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte)</code></td>
<td>(Inherited from <code>IAdsReadWrite [870].</code>)</td>
</tr>
<tr>
<td><code>RegisterAdsStateChangedAsync</code></td>
<td>Registers for AdsStateChanged [934] events as an asynchronous operation. (Inherited from <code>IAdsStateProvider [929].</code>)</td>
</tr>
<tr>
<td><code>RegisterSymbolVersionChangedAsync</code></td>
<td>Registers the symbol version changed asynchronously. (Inherited from <code>IAdsSymbolChangedProvider [934].</code>)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [866] event. (Inherited from <code>IAdsNotifications [839].</code>)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [866] event. (Inherited from <code>IAdsNotifications [839].</code>)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from <code>IAdsNotifications [839].</code>)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from <code>IAdsNotifications [839].</code>)</td>
</tr>
<tr>
<td><code>TryCreateVariableHandle</code></td>
<td>Determines the Symbol handle by its instance path synchronously. (Inherited from <code>IAdsHandle [827].</code>)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryDeleteDeviceNotification</td>
<td>Deletes a registered notification. (Inherited from IAdsNotifications[839].)</td>
</tr>
<tr>
<td>TryDeleteVariableHandle</td>
<td>Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle[827].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke[886].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke[886].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object)</td>
<td>Invokes the rpc method. (Inherited from IAdsRpcInvoke[886].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object)</td>
<td>Invokes the rpc method. (Inherited from IAdsRpcInvoke[886].)</td>
</tr>
<tr>
<td>TryRead(UInt32, Memory, Void)</td>
<td>(Inherited from IAdsHandle[827].)</td>
</tr>
<tr>
<td>TryRead(UInt32, UInt32, Memory, Void)</td>
<td>(Inherited from IAdsReadWrite[870].)</td>
</tr>
<tr>
<td>TryReadDataType</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess[937].)</td>
</tr>
<tr>
<td>TryReadState</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful. (Inherited from IAdsStateProvider[929].)</td>
</tr>
<tr>
<td>TryReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess[937].)</td>
</tr>
<tr>
<td>TryReadValue(ISymbol, Object)</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess[937].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryReadValue(String, Type, Object) [956]</td>
<td>Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue_T.(String, T.) [954]</td>
<td>Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue_T.(ISymbol, T.) [955]</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, Memory, Void, Byte) [837]</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, UInt32, Memory, Void, Byte) [873]</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryWrite(UInt32, ReadOnlyMemory) [837]</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryWrite(UInt32, UInt32, ReadOnlyMemory) [873]</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo) [917]</td>
<td>Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, ReadOnlyMemory) [917]</td>
<td>(Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>TryWriteValue(String, Object) [957]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue(ISymbol, Object) [959]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue_T.(String, T.) [958]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue_T.(ISymbol, T.) [960]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>UnregisterAdsStateChangedAsync [933]</td>
<td>Registers for AdsStateChanged [934] events as an asynchronous operation. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UnregisterSymbolVersionChangedAsync</td>
<td>Unregisters the symbol version changed asynchronous. (Inherited from IAdsSymbolChangedProvider [934].)</td>
</tr>
<tr>
<td>Write(UInt32, ReadOnlyMemory)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>Write(UInt32, UInt32)</td>
<td>Triggers a 'Write' call to the ADS device at the specified address. (Inherited from IAdsReadWrite2 [875].)</td>
</tr>
<tr>
<td>Write(UInt32, UInt32, ReadOnlyMemory)</td>
<td>(Inherited from IAdsReadWrite2 [875].)</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object, Int32)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAny(UInt32, UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, CancellationToken)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, Int32, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [702].)</td>
</tr>
<tr>
<td>WriteAsync(UInt32, ReadOnlyMemory, Void)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void)</td>
<td>Changes the ADS status and the device status of an ADS server. (Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>WriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteControl(StateInfo, ReadOnlyMemory)</td>
<td>(Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, CancellationToken)</td>
<td>Changes the ADS status and device status of the ADS server asynchronously. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void)</td>
<td>(Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>WriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValue(ISymbol, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValue.T.(String, T)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValue.T.(ISymbol, T)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValueAsync(ISymbol, Object, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValueAsync.T.(String, T, CancellationToken)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>WriteValueAsync.T.(ISymbol, T, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
</tbody>
</table>
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollAdsState(IObservable&lt;Unit.&gt;[])</td>
<td>Overloaded. Gets an observable sequence of AdsState[] via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollAdsState(TimeSpan)</td>
<td>Overloaded. Gets an observable sequence of AdsState[] via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollAdsStateAsync(IObservable&lt;Unit., CancellationToken&gt;)</td>
<td>Overloaded. Gets an observable sequence of AdsState[] via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, IObservable&lt;Unit.&gt;)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, IObservable&lt;Unit., Func.Exception, Object.&gt;)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, TimeSpan, Func.Exception, Object.)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, IObservable&lt;Unit., Func.Exception, Object.&gt;)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, TimeSpan, Func.Exception, Object.)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, IObservable&lt;Unit.&gt;)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollValues.T.(String, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, IObservable.Unit, Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, TimeSpan, Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, Int32, IObservable.Unit)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, Int32, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, Int32, IObservable.Unit, Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues.T.(String, Int32, TimeSpan, Func.Exception, T.)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>WhenAdsStateChanges</td>
<td>Gets an observable sequence of AdsState.s. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol)</td>
<td>Overloaded. Gets an observable sequence of Notification.s. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection)</td>
<td>Overloaded. Gets an observable sequence of Notification objects. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol, NotificationSettings)</td>
<td>Overloaded. Gets an observable sequence of SymbolValueNotification.s. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection, NotificationSettings)</td>
<td>Overloaded. Gets an observable sequence of Notification objects. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WhenNotification(String, Type, NotificationSettings...)</td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>WhenNotification(T, String, NotificationSettings...)</td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>WhenSymbolVersionChanges</td>
<td>Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenSymbolVersionChanges(ISchedule)</td>
<td>Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>WhenValueChanged</td>
<td>Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td>WriteValues(T, String, IObservable.T...)</td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>WriteValues(T, String, IObservable.T, Action.Exception...)</td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
</tbody>
</table>

Reference

IAdsConnection Interface [765]
TwinCAT.Ads Namespace [151]

6.2.32.2.1  IAdsConnection.ReadDeviceInfo Method

Reads the identification and version number of an ADS server.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
DeviceInfo ReadDeviceInfo()
```

Return Value

Type: DeviceInfo [698]
DeviceInfo struct containing the name of the device and the version information.
TwinCAT.Ads Namespaces

Reference

IAdsConnection Interface [765]
TwinCAT.Ads Namespace [151]

6.2.32.2 IAdsConnection.ReadDeviceInfoAsync Method

Reads the identification and version number of an ADS server.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

Task<ResultDeviceInfo> ReadDeviceInfoAsync(
    CancellationToken cancel
)

Parameters

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultDeviceInfo> [1001].
A task that represents the asynchronous 'ReadDeviceState' operation. The ResultDeviceInfo [1001] parameter contains the value DeviceInfo [1003] and the ErrorCode [992] of the ADS communication after execution.

Reference

IAdsConnection Interface [765]
TwinCAT.Ads Namespace [151]

6.2.32.3 IAdsConnection Events

The IAdsConnection [765] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsNotification [866]</td>
<td>Occurs when the ADS device sends a notification to the client. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AdsNotificationError [867]</td>
<td>Occurs when a exception has occurred during notification management. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AdsNotificationEx [868]</td>
<td>Occurs when the ADS device sends a notification to the client. (Inherited from IAdsNotifications [839].)</td>
</tr>
</tbody>
</table>
TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsStateChanged</td>
<td>Occurs when the AdsState of the target system has been changed. (Inherited from IAdsStateProvider.)</td>
</tr>
<tr>
<td>AdsSumNotification</td>
<td>Occurs when Notifications are send (bundled notifications) (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AdsSymbolVersionChanged</td>
<td>Occurs when the ADS Symbol Version changed. (Inherited from IAdsSymbolChangedProvider.)</td>
</tr>
<tr>
<td>ConnectionStateChanged</td>
<td>Occurs when connection status of the IConnectionStateProvider has been changed. (Inherited from IConnectionStateProvider.)</td>
</tr>
</tbody>
</table>

Reference

IAdsConnection Interface [765]

TwinCAT.Ads Namespace [151]

6.2.33 IAdsDisposableConnection Interface

Interface IAdsDisposableConnection Implements the IAdsConnectAddress [733] Implements the IRouterNotificationProvider [973] Implements the IAdsAnyAccess, IAdsHandle, IAdsReadWrite2, IAdsReadWrite, IAdsStateControl, IAdsSymbolChangedProvider, IAdsRpcInvoke, IRouterNotificationProvider, IDisposable


Syntax

C#

```csharp
public interface IAdsDisposableConnection : IAdsConnectAddress,
    IAdsConnection, IConnection, IConnectionStateProvider, IAdsNotifications, IAdsSymbolicAccess,
    IAdsAnyAccess, IAdsHandle, IAdsReadWrite2, IAdsReadWrite, IAdsStateProvider,
    IAdsStateControl, IAdsSymbolChangedProvider, IAdsRpcInvoke, IRouterNotificationProvider, IDisposable
```

The IAdsDisposableConnection type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address [781]</td>
<td>Gets the AmsAddress [648] of the ADS server. (Inherited from IAdsConnection [765].)</td>
</tr>
<tr>
<td>ClientAddress [781]</td>
<td>Get the AmsAddress [648] of the ADS client. (Inherited from IAdsConnection [765].)</td>
</tr>
<tr>
<td>ConnectionState [85]</td>
<td>Gets the current Connection state of the IConnectionStateProvider [84] (Inherited from IConnectionStateProvider [84].)</td>
</tr>
<tr>
<td>DefaultValueEncoding [76]</td>
<td>Gets the default value encoding. (Inherited from IConnection [74].)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id [76]</td>
<td>Gets the Connection Identifier. (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>IsConnected [76]</td>
<td>Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available. (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>IsDisposed [812]</td>
<td>Gets a value indicating whether this instance is disposed.</td>
</tr>
<tr>
<td>IsLocal [782]</td>
<td>Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer. (Inherited from IAdsConnection [765].)</td>
</tr>
<tr>
<td>Session [77]</td>
<td>Gets the session that initiated this IConnection [74] (Inherited from IConnection [74].)</td>
</tr>
<tr>
<td>Timeout [77]</td>
<td>Gets the timeout (in milliseconds) (Inherited from IConnection [74].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotification(String, Int32, NotificationSettings, Object) [844]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [866] event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [845]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [847]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [866] event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [848]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [866] event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [850]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>CleanupSymbolTable</td>
<td>Clears the internal symbol / DataTypes cache. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>Close</td>
<td>Closes this [Connection [74]. (Inherited from [Connection [74].)</td>
</tr>
<tr>
<td>Connect(Int32)</td>
<td>(Re)Connects the [Connection [74] when disconnected. (Inherited from [Connection [74].)</td>
</tr>
<tr>
<td>Connect(AmsAddress)</td>
<td>Connects to the local target ADS Device. (Inherited from [AdsConnectAddress [733].)</td>
</tr>
<tr>
<td>Connect(AmsNetId, Int32)</td>
<td>Connects the target ADS Device. (Inherited from [AdsConnectAddress [733].)</td>
</tr>
<tr>
<td>Connect(String, Int32)</td>
<td>Connects to the target ADS Device. (Inherited from [AdsConnectAddress [733].)</td>
</tr>
<tr>
<td>Connect(AmsNetId, Int32)</td>
<td>Connects to the target ADS Device. (Inherited from [AdsConnectAddress [733].)</td>
</tr>
<tr>
<td>CreateVariableHandle</td>
<td>Determines the Symbol handle by its instance path synchronously. (Inherited from [AdsHandle [827].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>CreateVariableHandleAsync</code></td>
<td>Determines the Symbol handle by its instance path asynchronously. (Inherited from <code>IAdsHandle</code>.)</td>
</tr>
<tr>
<td><code>DeleteDeviceNotification</code></td>
<td>Deletes a registered notification. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><code>DeleteDeviceNotificationAsync</code></td>
<td>Deletes a registered notification asynchronously. (Inherited from <code>IAdsNotifications</code>.)</td>
</tr>
<tr>
<td><code>DeleteVariableHandle</code></td>
<td>Releases the specified symbol/variable handle synchronously. (Inherited from <code>IAdsHandle</code>.)</td>
</tr>
<tr>
<td><code>DeleteVariableHandleAsync</code></td>
<td>Releases the specified symbol/variable handle asynchronously. (Inherited from <code>IAdsHandle</code>.)</td>
</tr>
<tr>
<td><code>Disconnect</code></td>
<td>Disconnects this <code>IConnection</code>. (Inherited from <code>IConnection</code>.)</td>
</tr>
<tr>
<td><code>Dispose</code></td>
<td>Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from <code>IDisposable</code>.)</td>
</tr>
<tr>
<td><code>InvokeRpcMethod(String, String, Object)</code></td>
<td>Invokes the specified RPC Method (Inherited from <code>IAdsRpcInvoke</code>.)</td>
</tr>
<tr>
<td><code>InvokeRpcMethod(String, String, Object, Object)</code></td>
<td>Invokes the specified RPC Method (Inherited from <code>IAdsRpcInvoke</code>.)</td>
</tr>
<tr>
<td><code>InvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object)</code></td>
<td>Invokes the specified RPC Method (Inherited from <code>IAdsRpcInvoke</code>.)</td>
</tr>
<tr>
<td><code>InvokeRpcMethodAsync(String, String, Object, CancellationToken)</code></td>
<td>Invokes the specified RPC Method asynchronously (Inherited from <code>IAdsRpcInvoke</code>.)</td>
</tr>
<tr>
<td><code>InvokeRpcMethodAsync(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</code></td>
<td>Invokes the specified RPC Method asynchronously (Inherited from <code>IAdsRpcInvoke</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from AdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>Read(UInt32, Memory)</td>
<td>(Inherited from AdsHandle [827].)</td>
</tr>
<tr>
<td>Read(UInt32, UInt32, Memory)</td>
<td>(Inherited from AdsReadWrite2 [875].)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess [702].)</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess [707].)</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess [708].)</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, UInt32)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess [710].)</td>
</tr>
<tr>
<td>ReadAny.T.(UInt32, UInt32, Int32.)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess [711].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess [716].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, Int32., CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess [718].)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, UInt32, Type, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from AdsAnyAccess [720].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, UInt32, Type, Int32, CancellationToken)</strong></td>
<td>(Inherited from IAdsAnyAccess) Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><strong>ReadAnyAsync_T.(UInt32, CancellationToken)</strong></td>
<td>(Inherited from IAdsAnyAccess) Reads data synchronously from an ADS device.</td>
</tr>
<tr>
<td><strong>ReadAnyAsync_T.(UInt32, Int32, CancellationToken)</strong></td>
<td>(Inherited from IAdsAnyAccess) Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><strong>ReadAnyAsync_T.(UInt32, UInt32, Int32, CancellationToken)</strong></td>
<td>(Inherited from IAdsAnyAccess) Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><strong>ReadAnyAsync_T.(UInt32, UInt32, CancellationToken)</strong></td>
<td>(Inherited from IAdsAnyAccess) Reads data asynchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td><strong>ReadAnyString(UInt32, Int32, Encoding)</strong></td>
<td>(Inherited from IAdsAnyAccess) Reads a string from the specified symbol/variable.</td>
</tr>
<tr>
<td><strong>ReadAnyString(UInt32, UInt32, Int32, Encoding)</strong></td>
<td>(Inherited from IAdsAnyAccess) Reads a string from a specified address.</td>
</tr>
<tr>
<td><strong>ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken)</strong></td>
<td>(Inherited from IAdsAnyAccess) Reads a string asynchronously from the specified symbol/variable.</td>
</tr>
<tr>
<td><strong>ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken)</strong></td>
<td>(Inherited from IAdsAnyAccess) Reads a string from a specified address asynchronously.</td>
</tr>
<tr>
<td><strong>ReadAsync(UInt32, Memory, Void)</strong></td>
<td>(Inherited from IAdsHandle)</td>
</tr>
<tr>
<td><strong>ReadAsync(UInt32, UInt32, Memory, Void)</strong></td>
<td>(Inherited from IAdsReadWrite)</td>
</tr>
<tr>
<td><strong>ReadDataType</strong></td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadDataTypeAsync</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadDeviceInfo</td>
<td>Reads the identification and version number of an ADS server. (Inherited from IAdsConnection.)</td>
</tr>
<tr>
<td>ReadDeviceInfoAsync</td>
<td>Reads the identification and version number of an ADS server. (Inherited from IAdsConnection.)</td>
</tr>
<tr>
<td>ReadState</td>
<td>Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider.)</td>
</tr>
<tr>
<td>ReadStateAsync</td>
<td>Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider.)</td>
</tr>
<tr>
<td>ReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadSymbolAsync</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValue(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValue(String, Type)</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValue.T.(String)</td>
<td>Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValueAsync(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously and returns it as an object. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValueAsync(String, Type, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValueAsync.T.(String, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadWrite(UInt32, Memory, Void)</td>
<td>(Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>ReadWrite(UInt32, UInt32, Memory, Void)</td>
<td>(Inherited from IAdsReadWrite2.)</td>
</tr>
<tr>
<td>ReadWriteAsync(UInt32, Memory, Void, Byte)</td>
<td>(Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte)</strong></td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td><strong>RegisterAdsStateChangedAsync [932]</strong></td>
<td>Registers for AdsStateChanged [934] events as an asynchronous operation. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td><strong>RegisterSymbolVersionChangedAsync [934]</strong></td>
<td>Registers the symbol version changed asynchronously. (Inherited from AdsSymbolChangedProvider [934].)</td>
</tr>
<tr>
<td><strong>TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.)</strong></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [866] event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td><strong>TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)</strong></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [866] event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td><strong>TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32., UInt32.)</strong></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td><strong>TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32., UInt32.)</strong></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td><strong>TryCreateVariableHandle [834]</strong></td>
<td>Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td><strong>TryDeleteDeviceNotification [864]</strong></td>
<td>Deletes a registered notification. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td><strong>TryDeleteVariableHandle [835]</strong></td>
<td>Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td><strong>TryInvokeRpcMethod(String, String, Object, Object.)</strong></td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object)</td>
<td>Invokes the rpc method. (Inherited from IAdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object)</td>
<td>Invokes the rpc method. (Inherited from IAdsRpcInvoke [886].)</td>
</tr>
<tr>
<td>TryRead(UInt32, Memory, Void)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryRead(UInt32, UInt32, Memory, Void)</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryReadDataType</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadState</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful. (Inherited from IAdsStateProvider [929].)</td>
</tr>
<tr>
<td>TryReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue(ISymbol, Object)</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue(String, Type, Object)</td>
<td>Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue.T.(String, T.)</td>
<td>Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue.T.(ISymbol, T.)</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryReadWriter(UInt32, Memory, Void, Byte) [837]</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryReadWriter(UInt32, Uint32, Memory, Void, Byte) [873]</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryWrite(UInt32, ReadOnlyMemory) [837]</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryWrite(UInt32, UInt32, ReadOnlyMemory) [873]</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo) [917]</td>
<td>Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, ReadOnlyMemory) [917]</td>
<td>(Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>TryWriteValue(String, Object) [957]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue(ISymbol, Object) [959]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue(T, String, T) [958]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue(T, ISymbol, T) [960]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>UnregisterAdsStateChangedAsync [933]</td>
<td>Registers for AdsStateChanged [934] events as an asynchronous operation. (Inherited from IAdsStateChangedProvider [929].)</td>
</tr>
<tr>
<td>UnregisterSymbolVersionChangedAsync [933]</td>
<td>Unregisters the symbol version changed asynchronous. (Inherited from IAdsSymbolChangedProvider [934].)</td>
</tr>
<tr>
<td>Write(UInt32, ReadOnlyMemory) [838]</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>Write(UInt32, UInt32) [878]</td>
<td>Triggers a 'Write' call to the ADS device at the specified address. (Inherited from IAdsReadWrite2 [875].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Write(UInt32, UInt32, ReadOnlyMemory)</td>
<td>(Inherited from IAdsReadWrite2)</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object, Int32)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object)</td>
<td>Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, CancellationToken)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, Int32, CancellationToken)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, Int32, CancellationToken)</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess)</td>
</tr>
<tr>
<td>WriteAsync(UInt32, ReadOnlyMemory, Void)</td>
<td>(Inherited from IAdsHandle)</td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void)</td>
<td>(Inherited from IAdsReadWrite)</td>
</tr>
<tr>
<td>WriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>WriteControl(StateInfo, ReadOnlyMemory)</code></td>
<td>(Inherited from <code>IAdsStateControl</code> [915].)</td>
</tr>
<tr>
<td><code>WriteControlAsync(AdsState, UInt16, CancellationToken)</code></td>
<td>Changes the ADS status and device status of the ADS server asynchronously. (Inherited from <code>IAdsStateControl</code> [915].)</td>
</tr>
<tr>
<td><code>WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void)</code></td>
<td>(Inherited from <code>IAdsStateControl</code> [915].)</td>
</tr>
<tr>
<td><code>WriteValue(String, Object)</code></td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from <code>IAdsSymbolicAccess</code> [937].)</td>
</tr>
<tr>
<td><code>WriteValue(ISymbol, Object)</code></td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from <code>IAdsSymbolicAccess</code> [937].)</td>
</tr>
<tr>
<td><code>WriteValue(ISymbol, T)</code></td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from <code>IAdsSymbolicAccess</code> [937].)</td>
</tr>
<tr>
<td><code>WriteValue(ISymbol, String, T)</code></td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from <code>IAdsSymbolicAccess</code> [937].)</td>
</tr>
<tr>
<td><code>WriteValueAsync(ISymbol, Object, CancellationToken)</code></td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from <code>IAdsSymbolicAccess</code> [937].)</td>
</tr>
<tr>
<td><code>WriteValueAsync(ISymbol, T, CancellationToken)</code></td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from <code>IAdsSymbolicAccess</code> [937].)</td>
</tr>
<tr>
<td><code>WriteValueAsync(ISymbol, String, T, CancellationToken)</code></td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from <code>IAdsSymbolicAccess</code> [937].)</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>AdsNotification</code></td>
<td>Occurs when the ADS device sends a notification to the client. (Inherited from <code>IAdsNotifications</code> [839].)</td>
</tr>
<tr>
<td><code>AdsNotificationError</code></td>
<td>Occurs when a exception has occurred during notification management. (Inherited from <code>IAdsNotifications</code> [839].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsNotificationEx</td>
<td>Occurs when the ADS devices sends a notification to the client. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AdsStateChanged</td>
<td>Occurs when the AdsState of the target system has been changed. (Inherited from IAdsStateProvider.)</td>
</tr>
<tr>
<td>AdsSumNotification</td>
<td>Occurs when Notifications are send (bundled notifications) (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AdsSymbolVersionChanged</td>
<td>Occurs when the ADS Symbol Version changed. (Inherited from IAdsSymbolChangedProvider.)</td>
</tr>
<tr>
<td>ConnectionStateChanged</td>
<td>Occurs when connection status of the IConnectionStateProvider has been changed. (Inherited from IConnectionStateProvider.)</td>
</tr>
<tr>
<td>RouterStateChanged</td>
<td>Router state changed event. (Inherited from IRouterNotificationProvider.)</td>
</tr>
</tbody>
</table>

### Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollAdsState(TimeSpan) [1062]</td>
<td>Overloaded. Gets an observable sequence of AdsState via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollAdsStateAsync(TimeSpan, CancellationToken) [1065]</td>
<td>Overloaded. Gets an observable sequence of AdsState via Polling. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, I.Observable.Unit.) [1089]</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan) [1090]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, TimeSpan) [1093]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollValues(String, Type, IObservable&lt;Unit, Func.Exception, Object&gt;) [1094]</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075]).</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan, Func.Exception, Object) [1095]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075]).</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, IObservable&lt;Unit, Func.Exception, Object&gt;) [1096]</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075]).</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, TimeSpan, Func.Exception, Object) [1097]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075]).</td>
</tr>
<tr>
<td>PollValues.T[String, IObservable&lt;Unit&gt;] [1083]</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075]).</td>
</tr>
<tr>
<td>PollValues.T[String, TimeSpan] [1084]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075]).</td>
</tr>
<tr>
<td>PollValues.T[String, Int32, IObservable&lt;Unit, Func.Exception, T&gt;] [1087]</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075]).</td>
</tr>
<tr>
<td>PollValues.T[String, TimeSpan, Func.Exception, T] [1088]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075]).</td>
</tr>
<tr>
<td>PollValues.T[String, Int32, IObservable&lt;Unit&gt;] [1085]</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075]).</td>
</tr>
<tr>
<td>PollValues.T[String, Int32, TimeSpan] [1086]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075]).</td>
</tr>
<tr>
<td>PollValues.T[String, Int32, IObservable&lt;Unit, Func.Exception, T&gt;] [1091]</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075]).</td>
</tr>
<tr>
<td>PollValues.T[String, Int32, TimeSpan, Func.Exception, T] [1092]</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075]).</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WhenAdsStateChanges</td>
<td>Gets an observable sequence of AdsState objects. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol)</td>
<td>Overloaded. Gets an observable sequence of Notification objects. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection)</td>
<td>Overloaded. Gets an observable sequence of Notification objects. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbol, NotificationSettings)</td>
<td>Overloaded. Gets an observable sequence of SymbolValueNotification objects. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(ISymbolCollection, NotificationSettings)</td>
<td>Overloaded. Gets an observable sequence of Notification objects. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenNotification(String, Type, NotificationSettings)</td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>WhenSymbolVersionChanges</td>
<td>Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenSymbolVersionChanges(ISchedule)</td>
<td>Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions.)</td>
</tr>
<tr>
<td>WhenValueChanged</td>
<td>Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions.)</td>
</tr>
<tr>
<td>WriteValues(String, IObservable)</td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions.)</td>
</tr>
<tr>
<td>WriteValues(String, IObservable, Action.Exception)</td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions.)</td>
</tr>
</tbody>
</table>
6.2.33.1 IAdsDisposableConnection Properties

The IAdsDisposableConnection type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address [781]</td>
<td>Gets the AmsAddress of the ADS server. (Inherited from IAdsConnection.)</td>
</tr>
<tr>
<td>ClientAddress [781]</td>
<td>Gets the AmsAddress of the ADS client. (Inherited from IAdsConnection.)</td>
</tr>
<tr>
<td>ConnectionState [85]</td>
<td>Gets the current Connection state of the IConnectionStateProvider. (Inherited from IConnectionStateProvider.)</td>
</tr>
<tr>
<td>DefaultValueEncoding [76]</td>
<td>Gets the default value encoding. (Inherited from IConnection.)</td>
</tr>
<tr>
<td>Id [76]</td>
<td>Gets the Connection Identifier. (Inherited from IConnection.)</td>
</tr>
<tr>
<td>IsConnected [76]</td>
<td>Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available. (Inherited from IConnection.)</td>
</tr>
<tr>
<td>IsDisposed [812]</td>
<td>Gets a value indicating whether this instance is disposed.</td>
</tr>
<tr>
<td>IsLocal [782]</td>
<td>Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer. (Inherited from IAdsConnection.)</td>
</tr>
<tr>
<td>Session [77]</td>
<td>Gets the session that initiated this IConnection. (Inherited from IConnection.)</td>
</tr>
<tr>
<td>Timeout [77]</td>
<td>Gets the timeout (in milliseconds) (Inherited from IConnection.)</td>
</tr>
</tbody>
</table>

Reference

IAdsDisposableConnection Interface [797]
TwinCAT.Ads Namespace [151]
Syntax

C#

```csharp
bool IsDisposed { get; }
```

**Property Value**

Type: Boolean
true if this instance is disposed; otherwise, false.

**Reference**

[IAdsDisposableConnection Interface](#797)
[TwinCAT.Ads Namespace](#151)

### 6.2.33.2 IAdsDisposableConnection Methods

The [IAdsDisposableConnection](#797) type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotification(String, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from <a href="#839">IAdsNotifications</a>.)</td>
</tr>
<tr>
<td>AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from <a href="#839">IAdsNotifications</a>.)</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event. (Inherited from <a href="#839">IAdsNotifications</a>.)</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event. (Inherited from <a href="#839">IAdsNotifications</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32., CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, Int32., CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, Int32., CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx event. (Inherited from IAdsNotifications.)</td>
</tr>
<tr>
<td>CleanupSymbolTable</td>
<td>Clears the internal symbol / DataTypes cache. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>Close</td>
<td>Closes this IConnection. (Inherited from IConnection.)</td>
</tr>
<tr>
<td>Connect</td>
<td>(Re)Connects the IConnection when disconnected. (Inherited from IConnection.)</td>
</tr>
<tr>
<td>Connect(Int32)</td>
<td>Connects to the local target ADS Device. (Inherited from IAdsConnectAddress.)</td>
</tr>
<tr>
<td>Connect(AmsAddress)</td>
<td>Connects the target ADS Device. (Inherited from IAdsConnectAddress.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Connect(String, Int32)</td>
<td>Connects to the target ADS Device. (Inherited from IAdsConnectAddress)</td>
</tr>
<tr>
<td>Connect(AmsNetId, Int32)</td>
<td>Connects to the target ADS Device. (Inherited from IAdsConnectAddress)</td>
</tr>
<tr>
<td>CreateVariableHandle</td>
<td>Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle)</td>
</tr>
<tr>
<td>CreateVariableHandleAsync</td>
<td>Determines the Symbol handle by its instance path asynchronously. (Inherited from IAdsHandle)</td>
</tr>
<tr>
<td>DeleteDeviceNotification</td>
<td>Deletes a registered notification. (Inherited from IAdsNotifications)</td>
</tr>
<tr>
<td>DeleteDeviceNotificationAsync</td>
<td>Deletes a registered notification asynchronously. (Inherited from IAdsNotifications)</td>
</tr>
<tr>
<td>DeleteVariableHandle</td>
<td>Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle)</td>
</tr>
<tr>
<td>DeleteVariableHandleAsync</td>
<td>Releases the specified symbol/variable handle asynchronously. (Inherited from IAdsHandle)</td>
</tr>
<tr>
<td>Disconnect</td>
<td>Disconnects this IConnection. (Inherited from IConnection)</td>
</tr>
<tr>
<td>Dispose</td>
<td>Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from IDisposable)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IAdsRpcInvoke)</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, String, Object, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>InvokeRpcMethodA sync(String, String, Object, Any TypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from AdsRpcInvoke.)</td>
</tr>
<tr>
<td>InvokeRpcMethodA sync(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from AdsRpcInvoke.)</td>
</tr>
<tr>
<td>Read(UInt32, Memory)</td>
<td>(Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>Read(UInt32, UInt32, Memory)</td>
<td>(Inherited from IAdsReadWrite2.)</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, Type, Int32.)</td>
<td>(Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAny(UInt32, UInt32, Type, Int32.)</td>
<td>(Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, CancellationToken)</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadAnyAsync(UInt32, Type, CancellationToken)</td>
<td>(Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, Type, .Int32., CancellationToken)</strong></td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, UInt32, Type, Int32, CancellationToken)</strong></td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken)</strong></td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, Type, .Int32., CancellationToken)</strong></td>
<td>Reads data synchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken)</strong></td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data synchronously from an ADS device. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data synchronously from an ADS device. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data synchronously from an ADS device. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data synchronously from an ADS device. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data synchronously from an ADS device. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data synchronously from an ADS device. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data synchronously from an ADS device. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data synchronously from an ADS device. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data synchronously from an ADS device. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data synchronously from an ADS device. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data asynchronously from an ADS device and writes it to an object. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td><strong>ReadAnyAsync(UInt32, .Int32., CancellationToken)</strong></td>
<td>Reads data synchronously from an ADS device. (Inherited from <code>IAdsAnyAccess</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadAnyStringAsynct(UInt32, UInt32, Int32, Encoding, CancellationToken)</td>
<td>Reads a string from a specified address asynchronously. (Inherited from IAdsAnyAccess.)</td>
</tr>
<tr>
<td>ReadAsync(UInt32, Memory, Void)</td>
<td>(Inherited from IAdsHandle.)</td>
</tr>
<tr>
<td>ReadAsync(UInt32, UInt32, Memory, Void)</td>
<td>(Inherited from IAdsReadWrite.)</td>
</tr>
<tr>
<td>ReadDataType</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadDataTypeAsync</td>
<td>Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadDeviceInfo</td>
<td>Reads the identification and version number of an ADS server. (Inherited from IAdsConnection.)</td>
</tr>
<tr>
<td>ReadDeviceInfoAsync</td>
<td>Reads the identification and version number of an ADS server. (Inherited from IAdsConnection.)</td>
</tr>
<tr>
<td>ReadState</td>
<td>Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider.)</td>
</tr>
<tr>
<td>ReadStateAsync</td>
<td>Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider.)</td>
</tr>
<tr>
<td>ReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadSymbolAsync</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValue(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValue(String, Type)</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValueT(String)</td>
<td>Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValueAsync(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously and returns it as an object. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValueAsync(String, Type, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>ReadValueAsyncT(String, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>ReadWrite(UInt32, Memory, Void)</code></td>
<td>(Inherited from <code>IAdsHandle</code>)</td>
</tr>
<tr>
<td><code>ReadWrite(UInt32, UInt32, Memory, Void)</code></td>
<td>(Inherited from <code>IAdsReadWrite2</code>)</td>
</tr>
<tr>
<td><code>ReadWriteAsync(UInt32, Memory, Void)</code></td>
<td>(Inherited from <code>IAdsHandle</code>)</td>
</tr>
<tr>
<td><code>ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte)</code></td>
<td>(Inherited from <code>IAdsReadWrite</code>)</td>
</tr>
<tr>
<td><code>RegisterAdsStateChangedAsync</code></td>
<td>Registers for <code>AdsStateChanged</code> events as an asynchronous operation. (Inherited from <code>IAdsStateProvider</code>)</td>
</tr>
<tr>
<td><code>RegisterSymbolVersionChangedAsync</code></td>
<td>Registers the symbol version changed asynchronously. (Inherited from <code>IAdsSymbolChangedProvider</code>)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotification</code> event. (Inherited from <code>IAdsNotifications</code>)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotification</code> event. (Inherited from <code>IAdsNotifications</code>)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32, UInt32)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotificationEx</code> event. (Inherited from <code>IAdsNotifications</code>)</td>
</tr>
<tr>
<td><code>TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32, UInt32)</code></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotificationEx</code> event. (Inherited from <code>IAdsNotifications</code>)</td>
</tr>
<tr>
<td><code>TryCreateVariableHandle</code></td>
<td>Determines the Symbol handle by its instance path synchronously. (Inherited from <code>IAdsHandle</code>)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryDeleteDeviceNotification</td>
<td>Deletes a registered notification. (Inherited from <code>IAdsNotifications</code>[839].)</td>
</tr>
<tr>
<td>TryDeleteVariableHandle</td>
<td>Releases the specified symbol/variable handle synchronously. (Inherited from <code>IAdsHandle</code>[827].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object.)</td>
<td>Invokes the specified RPC Method (Inherited from <code>IAdsRpcInvoke</code>[886].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object.)</td>
<td>Invokes the specified RPC Method (Inherited from <code>IAdsRpcInvoke</code>[886].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object.)</td>
<td>Invokes the rpc method. (Inherited from <code>IAdsRpcInvoke</code>[886].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object.)</td>
<td>Invokes the rpc method. (Inherited from <code>IAdsRpcInvoke</code>[886].)</td>
</tr>
<tr>
<td>TryRead(UInt32, Memory, Void)</td>
<td>(Inherited from <code>IAdsHandle</code>[827].)</td>
</tr>
<tr>
<td>TryRead(UInt32, UInt32, Memory, Void)</td>
<td>(Inherited from <code>IAdsReadWrite</code>[870].)</td>
</tr>
<tr>
<td>TryReadDataType</td>
<td>Call this method to obtain information about the specified data type. (Inherited from <code>IAdsSymbolicAccess</code>[937].)</td>
</tr>
<tr>
<td>TryReadState</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorcode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful. (Inherited from <code>IAdsStateProvider</code>[929].)</td>
</tr>
<tr>
<td>TryReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from <code>IAdsSymbolicAccess</code>[937].)</td>
</tr>
<tr>
<td>TryReadValue(ISymbol, Object)</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from <code>IAdsSymbolicAccess</code>[937].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryReadValue(String, Type, Object)</td>
<td>Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue.T.(String, T.) [954]</td>
<td>Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadValue.T.(ISymbol, T.) [955]</td>
<td>Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, Memory, Void, Byte)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryReadWrite(UInt32, UInt32, Memory, Void, Byte)</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryWrite(UInt32, ReadOnlyMemory)</td>
<td>(Inherited from IAdsHandle [827].)</td>
</tr>
<tr>
<td>TryWrite(UInt32, UInt32, ReadOnlyMemory)</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, ReadOnlyMemory)</td>
<td>(Inherited from IAdsStateControl [915].)</td>
</tr>
<tr>
<td>TryWriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue(ISymbol, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue.T.(String, T) [958]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>TryWriteValue.T.(ISymbol, T) [960]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [937].)</td>
</tr>
<tr>
<td>UnregisterAdsStateChangedAsync</td>
<td>Registers for AdsStateChanged [934] events as an asynchronous operation. (Inherited from IAdsStateChangedProvider [929].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UnregisterSymbolVersionChangedAsync</td>
<td>Unregisters the symbol version changed asynchronous. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>AdsSymbolChangedProvider [6934].)</td>
</tr>
<tr>
<td>Write(UInt32, ReadOnlyMemory) [6838]</td>
<td>(Inherited from AdsHandle [6827].)</td>
</tr>
<tr>
<td>Write(UInt32, UInt32) [6875]</td>
<td>Triggers a 'Write' call to the ADS device at the specified address. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from AdsReadWrite2 [6875].)</td>
</tr>
<tr>
<td>Write(UInt32, UInt32, ReadOnlyMemory) [6878]</td>
<td>(Inherited from AdsReadWrite2 [6875].)</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object) [6726]</td>
<td>Writes an object synchronously to an ADS device. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>AdsAnyAccess [6702].)</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object, Int32) [6727]</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object</td>
</tr>
<tr>
<td></td>
<td>to be written is a string type, the first element of parameter args specifies</td>
</tr>
<tr>
<td></td>
<td>the number of characters of the string. (Inherited from AdsAnyAccess</td>
</tr>
<tr>
<td></td>
<td>[6702].)</td>
</tr>
<tr>
<td>WriteAny(UInt32, Object) [6728]</td>
<td>Writes an object synchronously to an ADS device. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>AdsAnyAccess [6702].)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, CancellationToken) [6730]</td>
<td>Writes an object synchronously to an ADS device. If the Type of the object</td>
</tr>
<tr>
<td></td>
<td>to be written is a string type, the first element of parameter args specifies</td>
</tr>
<tr>
<td></td>
<td>the number of characters of the string. (Inherited from AdsAnyAccess</td>
</tr>
<tr>
<td></td>
<td>[6702].)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, Object, Int32, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object</td>
</tr>
<tr>
<td></td>
<td>to be written is a string type, the first element of parameter args specifies</td>
</tr>
<tr>
<td></td>
<td>the number of characters of the string. (Inherited from AdsAnyAccess</td>
</tr>
<tr>
<td></td>
<td>[6702].)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object</td>
</tr>
<tr>
<td></td>
<td>to be written is a string type, the first element of parameter args specifies</td>
</tr>
<tr>
<td></td>
<td>the number of characters of the string. (Inherited from AdsAnyAccess</td>
</tr>
<tr>
<td></td>
<td>[6702].)</td>
</tr>
<tr>
<td>WriteAnyAsync(UInt32, UInt32, Object, CancellationToken)</td>
<td>Writes an object asynchronously to an ADS device. If the Type of the object</td>
</tr>
<tr>
<td></td>
<td>to be written is a string type, the first element of parameter args specifies</td>
</tr>
<tr>
<td></td>
<td>the number of characters of the string. (Inherited from AdsAnyAccess</td>
</tr>
<tr>
<td></td>
<td>[6702].)</td>
</tr>
<tr>
<td>WriteAsync(UInt32, Void) [6838]</td>
<td>(Inherited from AdsHandle [6827].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void)</td>
<td>(Inherited from IAdsReadWrite.)</td>
</tr>
<tr>
<td>WriteControl(Statelnfo)</td>
<td>Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl.)</td>
</tr>
<tr>
<td>WriteControl(Statelnfo, ReadOnlyMemory)</td>
<td>(Inherited from IAdsStateControl.)</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, CancellationToken)</td>
<td>Changes the ADS status and device status of the ADS server asynchronously. (Inherited from IAdsStateControl.)</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void)</td>
<td>(Inherited from IAdsStateControl.)</td>
</tr>
<tr>
<td>WriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>WriteValue(ISymbol, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>WriteValue&lt;T&gt;(String, T)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>WriteValue&lt;T&gt;(ISymbol, T, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>WriteValueAsync(ISymbol, Object, CancellationToken)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>WriteValueAsync&lt;T&gt;(String, T, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
<tr>
<td>WriteValueAsync&lt;T&gt;(ISymbol, T, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess.)</td>
</tr>
</tbody>
</table>
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollAdsStateAsync(I Observable&lt;Unit&gt;, CancellationToken)</td>
<td>Overloaded. Gets an observable sequence of AdsState [626]s via Polling. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td>PollValues(String, Type, IObservable&lt;Unit&gt;)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, TimeSpan)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, IObservable&lt;Unit&gt;, Func.Exception, Object)</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan, Func.Exception, Object)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, IObservable&lt;Unit&gt;, Func.Exception, Object)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, TimeSpan, Func.Exception, Object)</td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues(String, Type, Int32, TimeSpan, Func.Exception, Object)</td>
<td>Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td>PollValues&lt;T&gt;(String, IObservable&lt;Unit&gt;) [1083]</td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [1075].)</td>
</tr>
</tbody>
</table>

---

**Version:** 1.1
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PollValues.T.(String, TimeSpan)</code></td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues.T.(String, Iobservable.Unit, Func.Exception, T.)</code></td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues.T.(String, TimeSpan, Func.Exception, T.)</code></td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues.T.(String, Int32, Iobservable.Unit)</code></td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues.T.(String, Int32, TimeSpan)</code></td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues.T.(String, Int32, Iobservable.Unit, Func.Exception, T.)</code></td>
<td>Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>PollValues.T.(String, Int32, TimeSpan)</code></td>
<td>Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by <code>AnyTypeExtensions</code>.)</td>
</tr>
<tr>
<td><code>WhenAdsStateChanges</code></td>
<td>Gets an observable sequence of <code>AdsState</code> objects. (Defined by <code>AdsClientExtensions</code>.)</td>
</tr>
<tr>
<td><code>WhenNotification(ISymbol)</code></td>
<td>Overloaded. Gets an observable sequence of <code>Notification</code> objects. (Defined by <code>AdsClientExtensions</code>.)</td>
</tr>
<tr>
<td><code>WhenNotification(ISymbolCollection)</code></td>
<td>Overloaded. Gets an observable sequence of <code>Notification</code> objects. (Defined by <code>AdsClientExtensions</code>.)</td>
</tr>
<tr>
<td><code>WhenNotification(ISymbol, NotificationSettings)</code></td>
<td>Overloaded. Gets an observable sequence of <code>SymbolValueNotification</code> objects. (Defined by <code>AdsClientExtensions</code>.)</td>
</tr>
<tr>
<td><code>WhenNotification(ISymbolCollection, NotificationSettings)</code></td>
<td>Overloaded. Gets an observable sequence of <code>Notification</code> objects. (Defined by <code>AdsClientExtensions</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>WhenNotification(String, Type, NotificationSettings)</strong></td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td><strong>WhenNotification_T(String, NotificationSettings)</strong></td>
<td>Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td><strong>WhenSymbolVersionChanges</strong></td>
<td>Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td><strong>WhenSymbolVersionChanges(ISchedule)</strong></td>
<td>Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [1056].)</td>
</tr>
<tr>
<td><strong>WhenValueChanged</strong></td>
<td>Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td><strong>WriteValues_T(String, IObservable.T.)</strong></td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
<tr>
<td><strong>WriteValues_T(String, IObservable.T., Action.Exception.)</strong></td>
<td>Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [1075].)</td>
</tr>
</tbody>
</table>

**Reference**

IAdsDisposableConnection Interface [797]

TwinCAT.Ads Namespace [151]

### 6.2.33.3 IAdsDisposableConnection Events

The IAdsDisposableConnection interface type exposes the following members.

**Events**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsNotification</td>
<td>Occurs when the ADS device sends a notification to the client. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AdsNotificationError</td>
<td>Occurs when a exception has occurred during notification management. (Inherited from IAdsNotifications [839].)</td>
</tr>
<tr>
<td>AdsNotificationEx</td>
<td>Occurs when the ADS devices sends a notification to the client. (Inherited from IAdsNotifications [839].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsStateChanged</td>
<td>Occurs when the AdsState of the target system has been changed. (Inherited from IAdsStateProvider)</td>
</tr>
<tr>
<td>AdsSumNotification</td>
<td>Occurs when Notifications are send (bundled notifications) (Inherited from IAdsNotifications)</td>
</tr>
<tr>
<td>AdsSymbolVersionChanged</td>
<td>Occurs when the ADS Symbol Version changed. (Inherited from IAdsSymbolChangedProvider)</td>
</tr>
<tr>
<td>ConnectionStateChanged</td>
<td>Occurs when connection status of the IConnectionStateProvider has been changed. (Inherited from IConnectionStateProvider)</td>
</tr>
<tr>
<td>RouterStateChanged</td>
<td>Router state changed event. (Inherited from IRouterNotificationProvider)</td>
</tr>
</tbody>
</table>

**Reference**

IAdsDisposableConnection Interface

TwinCAT.Ads Namespace

### 6.2.34 IAdsHandle Interface

Interface for ADS access via variable handle

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public interface IAdsHandle
```

The IAdsHandle type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateVariableHandle</td>
<td>Determines the Symbol handle by its instance path synchronously.</td>
</tr>
<tr>
<td>CreateVariableHandleAsync</td>
<td>Determines the Symbol handle by its instance path asynchronously.</td>
</tr>
<tr>
<td>DeleteVariableHandle</td>
<td>Releases the specified symbol/variable handle synchronously.</td>
</tr>
<tr>
<td>DeleteVariableHandleAsync</td>
<td>Releases the specified symbol/variable handle asynchronously.</td>
</tr>
<tr>
<td>Read</td>
<td></td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### 6.2.34.1 IAdsHandle Methods

The `IAdsHandle` type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateVariableHandle</td>
<td>Determines the Symbol handle by its instance path synchronously.</td>
</tr>
<tr>
<td>CreateVariableHandleAsync</td>
<td>Determines the Symbol handle by its instance path asynchronously.</td>
</tr>
<tr>
<td>DeleteVariableHandle</td>
<td>Releases the specified symbol/variable handle synchronously.</td>
</tr>
<tr>
<td>DeleteVariableHandleAsync</td>
<td>Releases the specified symbol/variable handle asynchronously.</td>
</tr>
<tr>
<td>Read</td>
<td></td>
</tr>
<tr>
<td>ReadAsync</td>
<td></td>
</tr>
<tr>
<td>ReadWrite</td>
<td></td>
</tr>
<tr>
<td>ReadWriteAsync</td>
<td></td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryCreateVariableHandle</td>
<td>Determines the Symbol handle by its instance path synchronously.</td>
</tr>
<tr>
<td>TryDeleteVariableHandle</td>
<td>Releases the specified symbol/variable handle synchronously.</td>
</tr>
<tr>
<td>TryRead</td>
<td></td>
</tr>
<tr>
<td>TryReadWrite</td>
<td></td>
</tr>
<tr>
<td>TryWrite</td>
<td></td>
</tr>
<tr>
<td>Write</td>
<td></td>
</tr>
<tr>
<td>WriteAsync</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

IAdsHandle Interface [827]

TwinCAT.Ads Namespace [151]

### 6.2.34.1.1 IAdsHandle.CreateVariableHandle Method

Determines the Symbol handle by its instance path synchronously.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
uint CreateVariableHandle(
    string symbolPath
)
```

**Parameters**

- `symbolPath` Type: `System.String` SymbolName / InstancePath.

**Return Value**

Type: `UInt32` The symbols/variable handle

**Remarks**

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this CreateVariableHandle(String) is the `DeleteVariableHandle(UInt32)` [830]

**Reference**

IAdsHandle Interface [827]
TwinCAT.Ads Namespace [151]

IAdsHandle.DeleteVariableHandle(UInt32) [830]

IAdsHandle.CreateVariableHandleAsync(String, CancellationToken) [830]

IAdsHandle.TryCreateVariableHandle(String, UInt32.) [834]

6.2.34.1.2 IAdsHandle.CreateVariableHandleAsync Method

Determines the Symbol handle by its instance path asynchronously.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Task<ResultHandle> CreateVariableHandleAsync(
    string symbolPath,
    CancellationToken cancel
)
```

Parameters

- **symbolPath**: Type: System.String
  SymbolName / InstancePath.
- **cancel**: Type: System.Threading.CancellationToken
  The cancellation token.

Return Value

Type: Task<ResultHandle> [1005].
A task that represents the asynchronous 'CreateVariableHandle' operation. The ResultHandle [1005] parameter contains the variable handle (Handle [1007]) and the ErrorCode [992] after execution.

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this CreateVariableHandleAsync(String, CancellationToken) is the DeleteVariableHandleAsync(UInt32, CancellationToken) [831]

Reference

IAdsHandle Interface [827]

TwinCAT.Ads Namespace [151]

IAdsHandle.DeleteVariableHandleAsync(UInt32, CancellationToken) [831]

IAdsHandle.TryCreateVariableHandle(String, UInt32.) [834]

IAdsHandle.CreateVariableHandle(String) [829]

6.2.34.1.3 IAdsHandle.DeleteVariableHandle Method

Releases the specified symbol/variable handle synchronously.
### Namespace: TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
void DeleteVariableHandle(
    uint variableHandle
)
```

### Parameters

- **variableHandle**
  - Type: System.UInt32
  - Handle of the ADS variable

### Return Value

- Type: The ADS error code.

### Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this `TryDeleteVariableHandle(UInt32)` is the `TryCreateVariableHandle(String, UInt32.)`

### Reference

- IAdsHandle Interface
- TwinCAT.Ads Namespace
- IAdsHandle.CreateVariableHandle(String)
- IAdsHandle.TryDeleteVariableHandle(UInt32)
- IAdsHandle.DeleteVariableHandleAsync(UInt32, CancellationToken)

### 6.2.34.1.4 IAdsHandle.DeleteVariableHandleAsync Method

Releases the specified symbol/variable handle asynchronously.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
Task<ResultAds> DeleteVariableHandleAsync(
    uint variableHandle,
    CancellationToken cancel
)
```

### Parameters

- **variableHandle**
  - Type: System.UInt32
  - Handle of the ADS variable
cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value
Type: Task<ResultAds>
A task that represents the asynchronous 'DeleteVariableHandle' operation. The ResultAds parameter contains the ErrorCode after execution.

Remarks
It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this DeleteVariableHandleAsync(UInt32, CancellationToken) is the CreateVariableHandleAsync(String, CancellationToken)

Reference
IAdsHandle Interface
TwinCAT.Ads Namespace
IAdsHandle.CreateVariableHandleAsync(String, CancellationToken)
IAdsHandle.TryDeleteVariableHandle(UInt32)
IAdsHandle.DeleteVariableHandle(UInt32)

6.2.34.1.5 IAdsHandle.Read Method

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#

```
int Read(
    uint variableHandle,
    Memory readBuffer
)
```

Parameters
variableHandle Type: System.UInt32
readBuffer Type: Memory

Return Value
Type: Int32

Reference
IAdsHandle Interface
TwinCAT.Ads Namespace
6.2.34.1.6   IAdsHandle.ReadAsync Method

Namespace:   TwinCAT.Ads [151]
Assembly:   TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

Task<ResultRead> ReadAsync(
    uint variableHandle,
    Memory readBuffer,
    void cancel
)

Parameters

variableHandle     Type: System.UInt32
readBuffer         Type: Memory
cancel             Type: System.Void

Return Value

Type: Task<ResultRead> [1008].

Reference

IAdsHandle Interface [827]
TwinCAT.Ads Namespace [151]

6.2.34.1.7   IAdsHandle.ReadWrite Method

Namespace:   TwinCAT.Ads [151]
Assembly:   TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

int ReadWrite(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer
)

Parameters

variableHandle     Type: System.UInt32
readBuffer         Type: Memory
writeBuffer        Type: System.Void
Return Value
Type: Int32

Reference
IAdsHandle Interface [827]
TwinCAT.Ads Namespace [151]

6.2.34.1.8 IAdsHandle.ReadWriteAsync Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstracts (in TwinCAT.Ads.Abstracts.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax
C#

```csharp
Task<ResultReadWrite> ReadWriteAsync(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer,
    byte cancel
)
```

Parameters

- `variableHandle` Type: System.UInt32
- `readBuffer` Type: Memory
- `writeBuffer` Type: System.Void
- `cancel` Type: System.Byte

Return Value
Type: Task<ResultReadWrite> [1019].

Reference
IAdsHandle Interface [827]
TwinCAT.Ads Namespace [151]

6.2.34.1.9 IAdsHandle.TryCreateVariableHandle Method

Determines the Symbol handle by its instance path synchronously.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstracts (in TwinCAT.Ads.Abstracts.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

```csharp
AdsErrorCode TryCreateVariableHandle(
    string symbolPath,
    out uint variableHandle
)
```

Parameters

<table>
<thead>
<tr>
<th>symbolPath</th>
<th>Type: System.String</th>
<th>SymbolName / InstancePath.</th>
</tr>
</thead>
<tbody>
<tr>
<td>variableHandle</td>
<td>Type: System.UInt32</td>
<td>The symbols handle.</td>
</tr>
</tbody>
</table>

Return Value

Type: AdsErrorCode [575]
The ADS error code.

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this TryCreateVariableHandle(String, UInt32.) is the TryDeleteVariableHandle(UInt32) [835]

Reference

IAdsHandle Interface [827]

TwinCAT.Ads Namespace [151]

IAdsHandle.TryDeleteVariableHandle(UInt32) [835]

IAdsHandle.CreateVariableHandleAsync(String, CancellationToken) [830]

IAdsHandle.CreateVariableHandle(String) [829]

6.2.34.1.10 IAdsHandle.TryDeleteVariableHandle Method

Releases the specified symbol/variable handle synchronously.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
AdsErrorCode TryDeleteVariableHandle(
    uint variableHandle
)
```

Parameters

| variableHandle | Type: System.UInt32 | Handle of the ADS variable |
Return Value

Type: AdsErrorCode [575]
The ADS error code.

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this TryDeleteVariableHandle(UInt32) is the TryCreateVariableHandle(String, UInt32.) [834]

Reference

IAdsHandle Interface [827]
TwinCAT.Ads Namespace [151]
IAdsHandle.TryCreateVariableHandle(String, UInt32.) [834]
IAdsHandle.DeleteVariableHandleAsync(UInt32, CancellationToken) [831]
IAdsHandle.DeleteVariableHandle(UInt32) [830]

6.2.34.1.11 IAdsHandle.TryRead Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstracts (in TwinCAT.Ads.Abstracts.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
AdsErrorCode TryRead(
    uint variableHandle,
    Memory readBuffer,
    void readBytes
)
```

Parameters

| variableHandle | Type: System.UInt32 |
| readBuffer     | Type: Memory       |
| readBytes      | Type: System.Void  |

Return Value

Type: AdsErrorCode [575]

Reference

IAdsHandle Interface [827]
TwinCAT.Ads Namespace [151]
6.2.34.1.12 IAdsHandle.TryReadWrite Method

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
AdsErrorCode TryReadWrite(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer,
    byte readBytes
)
```

Parameters

- `variableHandle` Type: System.UInt32
- `readBuffer` Type: Memory
- `writeBuffer` Type: System.Void
- `readBytes` Type: System.Byte

Return Value

Type: AdsErrorCode

Reference

IAdsHandle Interface
TwinCAT.Ads Namespace

6.2.34.1.13 IAdsHandle.TryWrite Method

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
AdsErrorCode TryWrite(
    uint variableHandle,
    ReadOnlyMemory writeBuffer
)
```

Parameters

- `variableHandle` Type: System.UInt32
- `writeBuffer` Type: ReadOnlyMemory
Return Value
Type: AdsErrorCode [575]

Reference
IAdsHandle Interface [827]
TwinCAT.Ads Namespace [151]

6.2.34.1.14 IAdsHandle.Write Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#:

```csharp
void Write(
    uint variableHandle,
    ReadOnlyMemory writeBuffer
)
```

Parameters

variableHandle Type: System.UInt32
writeBuffer Type: ReadOnlyMemory

Reference

IAdsHandle Interface [827]
TwinCAT.Ads Namespace [151]

6.2.34.1.15 IAdsHandle.WriteAsync Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#:

```csharp
Task<ResultWrite> WriteAsync(
    uint variableHandle,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

Parameters

variableHandle Type: System.UInt32
writeBuffer Type: ReadOnlyMemory
cancel | Type: System.Void

Return Value
Type: Task.ResultWrite [1032].

Reference

IAdsHandle Interface [827]

TwinCAT.Ads Namespace [151]

6.2.35 IAdsNotifications Interface

Interface for Notification management.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public interface IAdsNotifications

The IAdsNotifications type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotification(String, Int32, NotificationSettings, Object) [844]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [866] event.</td>
</tr>
<tr>
<td>AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [845]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [847]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [866] event.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [¶ 848]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [¶ 850]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32) [¶ 851]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [¶ 852]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32) [¶ 853]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, Int32, CancellationToken) [¶ 855]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, Int32, CancellationToken) [¶ 856]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>DeleteDeviceNotification [¶ 857]</td>
<td>Deletes a registered notification.</td>
</tr>
</tbody>
</table>
### Name

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeleteDeviceNotificationAsync</td>
<td>Deletes a registered notification asynchronously.</td>
</tr>
<tr>
<td>TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32., UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32., UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>TryDeleteDeviceNotification</td>
<td>Deletes a registered notification.</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsNotification</td>
<td>Occurs when the ADS device sends a notification to the client.</td>
</tr>
<tr>
<td>AdsNotificationError</td>
<td>Occurs when a exception has occurred during notification management.</td>
</tr>
<tr>
<td>AdsNotificationEx</td>
<td>Occurs when the ADS device sends a notification to the client.</td>
</tr>
<tr>
<td>AdsSumNotification</td>
<td>Occurs when Notifications are send (bundled notifications)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads Namespace [151]
### 6.2.35.1 IAdsNotifications Methods

The *IAdsNotifications* type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="844" alt="AddDeviceNotification(String, Int32, NotificationSettings, Object)" /></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <em>AdsNotification</em> event.</td>
</tr>
<tr>
<td><img src="845" alt="AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object)" /></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <em>AdsNotification</em> event.</td>
</tr>
<tr>
<td><img src="847" alt="AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken)" /></td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the <em>AdsNotification</em> event.</td>
</tr>
<tr>
<td><img src="848" alt="AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)" /></td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the <em>AdsNotification</em> event.</td>
</tr>
<tr>
<td><img src="850" alt="AddDeviceNotificationEx(String, NotificationSettings, Object, Type)" /></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <em>AdsNotificationEx</em> event.</td>
</tr>
<tr>
<td><img src="851" alt="AddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32.)" /></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <em>AdsNotificationEx</em> event.</td>
</tr>
<tr>
<td><img src="852" alt="AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type)" /></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <em>AdsNotificationEx</em> event.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32.) [853]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotification</code> event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive (<code>AnyType</code>) types are allowed for the parameter type.</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, Int32., CancellationToken) [855]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the <code>AdsNotificationEx</code> event.</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, Int32., CancellationToken) [856]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the <code>AdsNotificationEx</code> event.</td>
</tr>
<tr>
<td>DeleteDeviceNotification [857]</td>
<td>Deletes a registered notification.</td>
</tr>
<tr>
<td>DeleteDeviceNotificationAsync [858]</td>
<td>Deletes a registered notification asynchronously.</td>
</tr>
<tr>
<td>TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [859]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotification</code> event.</td>
</tr>
<tr>
<td>TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [861]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotification</code> event.</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32., UInt32.) [862]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the <code>AdsNotificationEx</code> event.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32., UInt32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>TryDeleteDeviceNotification</td>
<td>Deletes a registered notification.</td>
</tr>
</tbody>
</table>

Reference

IAdsNotifications Interface [839]
TwinCAT.Ads Namespace [151]

6.2.35.1.1 IAdsNotifications.AddDeviceNotification Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotification(String, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
<tr>
<td>AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.</td>
</tr>
</tbody>
</table>

Reference

IAdsNotifications Interface [839]
TwinCAT.Ads Namespace [151]

IAdsNotifications.AddDeviceNotification Method (String, Int32, NotificationSettings, Object)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
uint AddDeviceNotification(
    string symbolPath,
    int dataSize,
    NotificationSettings settings,
    Object userData)
```

Parameters

- **symbolPath**
  - Type: `System.String`
  - Symbol / Instance path of the ADS variable.

- **dataSize**
  - Type: `System.Int32`
  - Maximum amount of data in bytes to receive with this ADS Notification

- **settings**
  - Type: `TwinCAT.Ads.NotificationSettings`
  - The settings.

- **userData**
  - Type: `System.Object`
  - This object can be used to store user specific data (tag data)

Return Value

- Type: `UInt32`
  - The notification handle.

Remarks

The `dataSize` Parameter defines the amount of bytes, that will be attached to the `AdsNotification` as value. Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotification(UInt32)` should always called when the notification is not used anymore.

Reference

- `IAdsNotifications Interface`
- `AddDeviceNotification Overload`
- `TwinCAT.Ads Namespace`
- `IAdsNotifications.AdsNotification`
- `IAdsNotifications.DeleteDeviceNotification(UInt32)`
- `AddDeviceNotification Overload`
- `AddDeviceNotificationAsync Overload`
- `TryAddDeviceNotification Overload`

**IAdsNotifications.AddDeviceNotification Method (UInt32, UInt32, Int32, NotificationSettings, Object)**

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.
Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
uint AddDeviceNotification(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData)
```

Parameters

- **indexGroup**
  - Type: System.UInt32
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - Type: System.UInt32
  - Contains the index offset number of the requested ADS service.

- **dataSize**
  - Type: System.Int32
  - Maximum amount of data in bytes to receive with this ADS Notification

- **settings**
  - Type: TwinCAT.Ads.NotificationSettings [979]
  - The Notification settings.

- **userData**
  - Type: System.Object
  - This object can be used to store user specific data (tag data)

Return Value

- Type: UInt32
  - The notification handle.

Remarks

The `dataSize` Parameter defines the amount of bytes, that will be attached to the `AdsNotification` [866] as value. Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotification(UInt32)` [857] should always called when the notification is not used anymore.

Reference

- IAdsNotifications Interface [839]
- AddDeviceNotification Overload [844]
- TwinCAT.Ads Namespace [151]
- IAdsNotifications.DeleteDeviceNotification(UInt32) [857]
- IAdsNotifications.AdsNotification [866]
- IAdsNotifications.AdsNotificationError [867]
- AddDeviceNotification Overload [844]
- TryAddDeviceNotification Overload [859]
### 6.2.35.1.2 IAdsNotifications.AddDeviceNotificationAsync Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) ![847]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification ![866] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) ![848]</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification ![866] event.</td>
</tr>
</tbody>
</table>

#### Reference

- [IAdsNotifications Interface](#) ![839]
- [TwinCAT.Ads Namespace](#) ![151]

### IAdsNotifications.AddDeviceNotificationAsync Method (String, Int32, NotificationSettings, Object, CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the [AdsNotification ![866]](# event.

**Namespace:** TwinCAT.Ads ![151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
Task<ResultHandle> AddDeviceNotificationAsync(  
    string symbolPath,  
    int dataSize,  
    NotificationSettings settings,  
    Object userData,  
    CancellationToken cancel)
```

#### Parameters

- **symbolPath**: Type: `System.String`
  The symbol/instance path of the ADS variable.
TwinCAT.Ads Namespaces

### dataSize
Type: System.Int32
Maximum amount of data in bytes to receive with this ADS Notification

### settings
Type: TwinCAT.Ads.NotificationSettings
The notification settings.

### userData
Type: System.Object
This object can be used to store user specific data (tag data)

### cancel
Type: System.Threading.CancellationToken
The Cancellation token.

## Return Value
Type: Task<
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle type parameter contains the created handle (Handle) and the ErrorCode after execution.

## Remarks
The `dataSize` Parameter defines the amount of bytes, that will be attached to the `AdsNotification` as value. Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotificationAsync(UInt32, CancellationToken)` should always be called when the notification is not used anymore.

## Reference
- IAdsNotifications Interface
- AddDeviceNotificationAsync Overload
- TwinCAT.Ads Namespace
- IAdsNotifications.AdsNotification
- IAdsNotifications.DeleteDeviceNotificationAsync(UInt32, CancellationToken)
- AddDeviceNotification Overload
- AddDeviceNotificationAsync Overload
- TryAddDeviceNotification Overload

### IAdsNotifications.AddDeviceNotificationAsync Method (UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)
Connects a variable to the ADS client asynchronously. The ADS client will be notified by the `AdsNotification` event.

**Namespace:** TwinCAT.Ads
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
Task<ResultHandle> AddDeviceNotificationAsync(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    CancellationToken cancel
)
```

Parameters

- **indexGroup**: Type: `System.UInt32`  
  The index group number of the requested ADS service.

- **indexOffset**: Type: `System.UInt32`  
  The index offset number of the requested ADS service.

- **dataSize**: Type: `System.Int32`  
  Maximum amount of data in bytes to receive with this ADS Notification

- **settings**: Type: `TwinCAT.Ads.NotificationSettings`  
  The notification settings.

- **userData**: Type: `System.Object`  
  This object can be used to store user specific data (tag data)

- **cancel**: Type: `System.Threading.CancellationToken`  
  The Cancellation token.

Return Value

Type: `Task<ResultHandle>`  
A task that represents the asynchronous 'AddDeviceNotification' operation. The `ResultHandle` type parameter contains the created handle (`Handle`) and the `ErrorCode` after execution.

Remarks

Parameter `dataSize` defines the amount of bytes, that will be attached to the `AdsNotification` as value. Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotificationAsync(UInt32, CancellationToken)` should always be called when the notification is not used anymore.

Reference

- `IAdsNotifications` Interface
- `AddDeviceNotificationAsync Overload`
- `TwinCAT.Ads Namespace`
- `IAdsNotifications.AdsNotification`
- `IAdsNotifications.DeleteDeviceNotificationAsync(UInt32, CancellationToken)`
6.2.35.1.3 IAdsNotifications.AddDeviceNotificationEx Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.</td>
</tr>
<tr>
<td>AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32.)</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.</td>
</tr>
</tbody>
</table>

Reference

IAdsNotifications Interface
TwinCAT.Ads Namespace

IAdsNotifications.AddDeviceNotificationEx Method (String, NotificationSettings, Object, Type)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ddca3e72bc0ea15da1c14

Syntax

C#

```csharp
uint AddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type
)
```
Parameters

symbolPath  
Type: System.String  
Symbol/Instance path of the ADS variable.

settings  
Type: TwinCAT.Ads.NotificationSettings  
The Notification settings.

userData  
Type: System.Object  
This object can be used to store user specific data (tag data)

type  
Type: System.Type  
Type of the object stored in the event argument ('AnyType')

Return Value

Type: UInt32  
The notification handle.

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to  
DeleteDeviceNotification(UInt32) should always called when the notification is not used anymore.

Reference

IAdsNotifications Interface  
AddDeviceNotificationEx Overload  
TwinCAT.Ads Namespace  
IAdsNotifications.AdsNotificationEx  
IAdsNotifications.DeleteDeviceNotification(UInt32)  
AddDeviceNotificationEx Overload  
AddDeviceNotificationExAsync Overload  
TryAddDeviceNotificationEx Overload

IAdsNotifications.AddDeviceNotificationEx Method (String, NotificationSettings, Object, Type, .Int32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.

Namespace: TwinCAT.Ads  
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  

uint AddDeviceNotificationEx(  
    string symbolPath,  
    NotificationSettings settings,  
    Object userData,  
    Type type,  
    int[] args  
)
TwinCAT.Ads Namespaces

Parameters

symbolPath  Type: System.String
Symbol/Instance path of the ADS variable.

settings  Type: TwinCAT.Ads.NotificationSettings
The Notification settings.

userData  Type: System.Object
This object can be used to store user specific data (tag data)

type  Type: System.Type
Type of the object stored in the event argument ('AnyType')

args  Type: System.Int32
Additional arguments (for 'AnyType')

Return Value

Type: UInt32
The notification handle.

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to
DeleteDeviceNotification(UInt32) should always called when the notification is not used anymore.

Reference

IAdsNotifications Interface

AddDeviceNotificationEx Overload
TwinCAT.Ads Namespace
IAdsNotifications.AdsNotificationEx
IAdsNotifications.DeleteDeviceNotification(UInt32)
AddDeviceNotificationEx Overload
AddDeviceNotificationExAsync Overload
TryAddDeviceNotificationEx Overload

IAdsNotifications.AddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
uint AddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type
)
```
Parameters

indexGroup  
Type: System.UInt32  
Contains the index group number of the requested ADS service.

indexOffset  
Type: System.UInt32  
Contains the index offset number of the requested ADS service.

settings  
Type: TwinCAT.Ads.NotificationSettings  
The Notification settings.

userData  
Type: System.Object  
This object can be used to store user specific data (tag data)

type  
Type: System.Type  
Type of the object stored in the event argument ('AnyType')

Return Value

Type: UInt32  
The notification handle.

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to  
DeleteDeviceNotification(UInt32) should always called when the notification is not used anymore.  
AddDeviceNotificationEx Overload

Reference

IAdsNotifications Interface

AddDeviceNotificationEx Overload

TwinCAT.Ads Namespace

IAdsNotifications.DeleteDeviceNotification(UInt32)  
IAdsNotifications.AdsNotificationEx  
IAdsNotifications.AdsNotificationError  
AddDeviceNotificationEx Overload  
TryAddDeviceNotificationEx Overload  
AddDeviceNotificationExAsync Overload

IAdsNotifications.AddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a  
string type, the first element of the parameter args specifies the number of characters of the string. If type is  
an array type, the number of elements for each dimension has to be specified in the parameter args. Only  
primitive ('AnyType') types are allowed for the parameter type.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
uint AddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args
)
```

Parameters

- **indexGroup**
  - Type: System.UInt32
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - Type: System.UInt32
  - Contains the index offset number of the requested ADS service.

- **settings**
  - Type: TwinCAT.Ads.NotificationSettings
  - The Notification settings.

- **userData**
  - Type: System.Object
  - This object can be used to store user specific data.

- **type**
  - Type: System.Type
  - Type of the object stored in the event argument.

- **args**
  - Type: System.Int32
  - Additional arguments for 'AnyType' types.

Return Value

- Type: UInt32
  - The notification handle.

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotification(UInt32)` should always be called when the notification is not used anymore.

Reference

- `IAdsNotifications.Interface` [839]
- `AddDeviceNotificationEx Overload` [850]
- `TwinCAT.Ads Namespace` [151]
- `IAdsNotifications.AdsNotificationEx` [868]
- `AddDeviceNotificationEx Overload` [850]
- `AddDeviceNotificationExAsync Overload` [855]
- `TryAddDeviceNotificationEx Overload` [862]
6.2.35.1.4 IAdsNotifications.AddDeviceNotificationExAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [868] event.</td>
</tr>
<tr>
<td>AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken)</td>
<td>Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [868] event.</td>
</tr>
</tbody>
</table>

Reference

IAdsNotifications Interface [839]
TwinCAT.Ads Namespace [151]

IAdsNotifications.AddDeviceNotificationExAsync Method (String, NotificationSettings, Object, Type, .Int32., CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [868] event.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0e0a15da1c14

Syntax

C#

```csharp
Task<ResultHandle> AddDeviceNotificationExAsync(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

Parameters

- **symbolPath**: Type: `System.String`
The symbol/instance path of the ADS variable.
- **settings**: Type: `TwinCAT.Ads.NotificationSettings [979]`
The notification settings.
- **userData**: Type: `System.Object`
  This object can be used to store user specific data (tag data)
**TwinCAT.Ads Namespaces**

**AddDeviceNotificationExAsync**

```
Task<ResultHandle> AddDeviceNotificationExAsync(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

**Version:** 1.1

**Remarks**

Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotificationAsync(UInt32, CancellationToken)` should always be called when the notification is not used anymore.

**Return Value**

Type: `Task<ResultHandle>`

A task that represents the asynchronous 'AddDeviceNotification' operation. The `ResultHandle` type parameter contains the created handle (`Handle`) after execution.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Parameters

- **indexGroup**
  Type: `System.UInt32`
  Contains the index group number of the requested ADS service.

- **indexOffset**
  Type: `System.UInt32`
  Contains the index offset number of the requested ADS service.

- **settings**
  Type: `TwinCAT.Ads.NotificationSettings`
  The settings.

- **userData**
  Type: `System.Object`
  This object can be used to store user specific data.

- **type**
  Type: `System.Type`
  Type of the object stored in the event argument, only Primitive 'AnyTypes' allowed.

- **args**
  Type: `System.Int32`
  Additional arguments (for 'AnyType')

- **cancel**
  Type: `System.Threading.CancellationToken`
  The Cancellation token.

Return Value

Type: `Task.ResultHandle`
A task that represents the asynchronous 'AddDeviceNotification' operation. The `ResultHandle` type parameter contains the created handle (`Handle`) and the `ErrorCode` after execution.

Remarks

If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive types (AnyType) are supported by this method. Because notifications allocate TwinCAT system resources, a complementary call to `DeleteDeviceNotificationAsync(UInt32, CancellationToken)` should always called when the notification is not used anymore.

Reference

- `IAdsNotifications.Interface`
- `AddDeviceNotificationExAsync Overload`
- `TwinCAT.Ads Namespace`
- `IAdsNotifications.DeleteDeviceNotificationAsync(UInt32, CancellationToken)`
- `IAdsNotifications.AdsNotificationEx`
- `IAdsNotifications.AdsNotificationError`
- `AddDeviceNotificationEx Overload`
- `TryAddDeviceNotificationEx Overload`
- `AddDeviceNotificationExAsync Overload`

6.2.35.1.5 `IAdsNotifications.DeleteDeviceNotification Method`

Deletes a registered notification.

Namespace: `TwinCAT.Ads`
Assembly: `TwinCAT.Ads.Abstractions` (in `TwinCAT.Ads.Abstractions.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
void DeleteDeviceNotification(
    uint notificationHandle
)
```

Parameters

- **notificationHandle**
  - Type: `System.UInt32`
  - Notification handle.

Remarks

This is the complementary method to `AddDeviceNotification Overload [844]` overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

- `IAdsNotifications Interface [839]`
- `TwinCAT.Ads Namespace [151]`
- `AddDeviceNotification Overload [844]`
- `IAdsNotifications.AdsNotification [866]`
- `TryAddDeviceNotification Overload [859]`
- `AddDeviceNotificationAsync Overload [847]`

### 6.2.35.1.6 IAdsNotifications.DeleteDeviceNotificationAsync Method

Deletes a registered notification asynchronously.

**Namespace:** `TwinCAT.Ads [151]`

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdac3e72bc0ea15da1c14

Syntax

C#

```csharp
Task<ResultAds> DeleteDeviceNotificationAsync(
    uint notificationHandle,
    CancellationToken cancel
)
```

Parameters

- **notificationHandle**
  - Type: `System.UInt32`
  - Notification handle.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.

Return Value

Type: `Task<ResultAds>[989]`.

A task that represents the asynchronous 'DeleteDeviceNotification' operation. The `ErrorCode [992]` property contains the ADS error code after execution.
Remarks

This is the complementary method to AddDeviceNotificationAsync Overload \(847\) overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

IAdsNotifications Interface \(839\)

TwinCAT.Ads Namespace \(151\)

AddDeviceNotificationAsync Overload \(847\)

IAdsNotifications.AdsNotification \(866\)

TryAddDeviceNotification Overload \(859\)

AddDeviceNotification Overload \(844\)

6.2.35.1.7 IAdsNotifications.TryAddDeviceNotification Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="859" alt="TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.)" /></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification (866) event.</td>
</tr>
<tr>
<td><img src="861" alt="TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)" /></td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification (866) event.</td>
</tr>
</tbody>
</table>

Reference

IAdsNotifications Interface \(839\)

TwinCAT.Ads Namespace \(151\)

IAdsNotifications.TryAddDeviceNotification Method (String, Int32, NotificationSettings, Object, UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification \(866\) event.

Namespace: TwinCAT.Ads \(151\)

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
AdsErrorCode TryAddDeviceNotification(
    string symbolPath,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    out uint handle
)
```

Parameters

- **symbolPath**
  - Type: `System.String`
  - The symbol/instance path of the ADS variable.

- **dataSize**
  - Type: `System.Int32`
  - Maximum amount of data in bytes to receive with this ADS Notification

- **settings**
  - Type: `TwinCAT.Ads.NotificationSettings` [979]
  - The notification settings.

- **userData**
  - Type: `System.Object`
  - This object can be used to store user specific data.

- **handle**
  - Type: `System.UInt32`
  - The notification handle.

Return Value

- Type: `AdsErrorCode` [575]
  - The ADS ErrorCode.

Remarks

The `dataSize` Parameter defines the amount of bytes, that will be attached to the `AdsNotification` [866] as value. Because notifications allocate TwinCAT system resources, a complementary call to `TryDeleteDeviceNotification(UInt32)` [864] should always be called when the notification is not used anymore.

Reference

- `IAdsNotifications.Interface` [839]
- `TryAddDeviceNotification Overload` [859]
- `TwinCAT.Ads Namespace` [151]
- `IAdsNotifications.AdsNotification` [866]
- `IAdsNotifications.TryDeleteDeviceNotification(UInt32)` [864]
- `AddDeviceNotification Overload` [844]
- `AddDeviceNotificationAsync Overload` [847]
- `TryAddDeviceNotification Overload` [859]
IAdsNotifications.TryAddDeviceNotification Method (UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification[866] event.

Namespace: TwinCAT.Ads[151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6096934fddca3e72bc0ea15da1c14

Syntax

C#

AdsErrorCode TryAddDeviceNotification(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    out uint handle
)

Parameters

indexGroup Type: System.UInt32
The index group number of the requested ADS service.

indexOffset Type: System.UInt32
The index offset number of the requested ADS service.

dataSize Type: System.Int32
Maximum amount of data in bytes to receive with this ADS Notification.

settings Type: TwinCAT.Ads.NotificationSettings[979]
The Notification settings.

userData Type: System.Object
This object can be used to store user specific data (tag data)

handle Type: System.UInt32
The notification handle.

Return Value

Type: AdsErrorCode[575]
The ADS error code.

Remarks

The dataSize Parameter defines the amount of bytes, that will be attached to the AdsNotification[866] as value. Because notifications allocate TwinCAT system resources, a complementary call to TryDeleteDeviceNotification(UInt32)[864] should always called when the notification is not used anymore.

Reference

IAdsNotifications Interface[839]
TryAddDeviceNotification Overload[859]
TwinCAT.Ads Namespace[151]
IAdsNotifications.TryDeleteDeviceNotification(UInt32)[864]
### TwinCAT.Ads Namespaces

- IAdsNotifications.AdsNotification [866]
- IAdsNotifications.AdsNotificationError [867]
- AddDeviceNotification Overload [844]
- TryAddDeviceNotification Overload [859]
- AddDeviceNotificationAsync Overload [847]

### 6.2.35.1.8 IAdsNotifications.TryAddDeviceNotificationEx Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, Int32, UInt32) [862]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event.</td>
</tr>
<tr>
<td>TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, Int32, UInt32) [863]</td>
<td>Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event.</td>
</tr>
</tbody>
</table>

#### Reference

- IAdsNotifications Interface [839]
- TwinCAT.Ads Namespace [151]

### IAdsNotifications.TryAddDeviceNotificationEx Method (String, NotificationSettings, Object, Type, Int32, UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [868] event.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
AdsErrorCode TryAddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    out uint handle
)
```
Parameters
symbolPath Type: System.String
Symbol/Instance path of the ADS variable.
settings Type: TwinCAT.Ads.NotificationSettings
The Notification settings.
userData Type: System.Object
This object can be used to store user specific data (tag data)
type Type: System.Type
Type of the object stored in the event argument ('AnyType')
args Type: System.Int32
Additional arguments (for 'AnyType')
handle Type: System.UInt32
The notification handle

Return Value
Type: AdsErrorCode
The ADS error code.

Remarks
Because notifications allocate TwinCAT system resources, a complementary call to
TryDeleteDeviceNotification(UInt32) should always called when the notification is not used anymore.

Reference
IAdsNotifications Interface
TryAddDeviceNotificationEx Overload
TwinCAT.Ads Namespace
IAdsNotifications.AdsNotificationEx
IAdsNotifications.DeleteDeviceNotification(UInt32)
AddDeviceNotificationEx Overload
AddDeviceNotificationExAsync Overload
TryAddDeviceNotificationEx Overload

IAdsNotifications.TryAddDeviceNotificationEx Method (UInt32, UInt32,
NotificationSettings, Object, Type, .Int32., UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx event.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
AdsErrorCode TryAddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
Object userData,
    Type type,
    int[] args,
    out uint handle
)

Parameters

indexGroup Type: System.UInt32
Contains the index group number of the requested ADS service.

indexOffset Type: System.UInt32
Contains the index offset number of the requested ADS service.

settings Type: TwinCAT.Ads.NotificationSettings
The Notification settings.

userData Type: System.Object
This object can be used to store user specific data (tag data)

type Type: System.Type
Type of the object stored in the event argument ('AnyType')

args Type: System.Int32
The 'AnyType' arguments.

handle Type: System.UInt32
The notification handle.

Return Value

Type: AdsErrorCode
The ADS Error code.

Remarks

If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive types (AnyType) are supported by this method. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) should always called when the notification is not used anymore.

Reference

IAdsNotifications Interface
TryAddDeviceNotificationEx Overload
TwinCAT.Ads Namespace
IAdsNotifications.DeleteDeviceNotification(UInt32)
IAdsNotifications.AdsNotificationEx
IAdsNotifications.AdsNotificationError
AddDeviceNotificationEx Overload
TryAddDeviceNotificationEx Overload
AddDeviceNotificationExAsync Overload

6.2.35.1.9 IAdsNotifications.TryDeleteDeviceNotification Method

Deletes a registered notification.
Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
AdsErrorCode TryDeleteDeviceNotification(uint notificationHandle)
```

Parameters

- `notificationHandle`  
  Type: System.UInt32  
  Notification handle.

Return Value

Type: AdsErrorCode

The ADS error code.

Remarks

This is the complementary method to TryAddDeviceNotification Overload overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

- IAdsNotifications Interface
- TwinCAT.Ads Namespace
- AddDeviceNotification Overload
- IAdsNotifications.AdsNotification
- TryAddDeviceNotification Overload
- AddDeviceNotificationAsync Overload

6.2.35.2 IAdsNotifications Events

The IAdsNotifications type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsNotification</td>
<td>Occurs when the ADS device sends a notification to the client.</td>
</tr>
<tr>
<td>AdsNotificationError</td>
<td>Occurs when a exception has occurred during notification management.</td>
</tr>
<tr>
<td>AdsNotificationEx</td>
<td>Occurs when the ADS devices sends a notification to the client.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsSumNotification</td>
<td>Occurs when Notifications are send (bundled notifications)</td>
</tr>
</tbody>
</table>

### Reference

- [IAdsNotifications Interface](#)
- [TwinCAT.Ads Namespace](#)

### 6.2.35.2.1 IAdsNotifications.AdsNotification Event

Occurs when the ADS device sends a notification to the client.

**Namespace**: TwinCAT.Ads [151]

**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
event EventHandler<AdsNotificationEventArgs> AdsNotification
```

**Value**

Type: System.EventHandler<AdsNotificationEventArgs> [597].

**Remarks**

The Event Argument contains the raw data value of the notification, not marshalled to .NET types.

**Examples**

Example of receiving AdsNotification events.

**Trigger on changed values by ADS Notifications**

```csharp
private async Task RegisterNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;

    using (AdsClient client = new AdsClient())
    {
        // Add the Notification event handler
        client.AdsNotification += Client_AdsNotification2;

        // Connect to target
        client.Connect(AmsNetId.Local, 851);
        uint notificationHandle = 0;

        // Notification to a DINT Type (UINT32)
        // Check for change every 200 ms
        byte[] notificationBuffer = new byte[sizeof(UInt32)];
        int size = sizeof(UInt32);
        ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", size, new NotificationSettings(AdsTransModeOnChange, 200, 0), null, cancel);
        if (result.Succeeded)
        {
            notificationHandle = result.Handle;
            await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
            // Unregister the Event / Handle
        }
    }
}
```
```csharp
ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
}
client.AdsNotification -= Client_AdsNotification2;
}

private void Client_AdsNotification2(object sender, AdsNotificationEventArgs e)
{
    // Or here we know about UDINT type --> can be marshalled as UINT32
    uint nCounter = BinaryPrimitives.ReadUInt32LittleEndian(e.Data.Span);
    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread
    /*SynchronizationContext syncContext = SynchronizationContext.Current;
     _context.Post(status => someLabel.Text = nCounter.ToString(), null); // Non-blocking post */
}
```

### Reference

**IAdsNotifications Interface [839]**

**TwinCAT.Ads Namespace [151]**

#### 6.2.35.2.2 IAdsNotifications.AdsNotificationError Event

Occurs when a exception has occurred during notification management.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#:

```csharp
event EventHandler<AdsNotificationErrorEventArgs> AdsNotificationError
```

**Value**

Type: `System.EventHandle<AdsNotificationErrorEventArgs> AdsNotificationError [595].`

**Remarks**

The occurrence of this event can have two different reasons:

1. Indicates an internal error occurred during Notification management.
2. The registered notification becomes invalid on the server, e.g. after a PLC Download / Online Change.
   
   If the ADS Server detects that the (still registered) Notification Sender is getting invalid, it sends an error notification so that the client will be informed about detached notifications. The event arguments contains the 'AdsInvalidNotificationException' which describes the invalid notification handle by its 'AdsInvalidNotificationException.Handle' property.
### 6.2.35.2.3 IAdsNotifications.AdsNotificationEx Event

Occurs when the ADS devices sends a notification to the client.

**Namespace:** TwinCAT.Ads namespace

**Assembly:** TwinCAT.Ads.Abstractions.dll (version 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14)

**Syntax**

C#  
```csharp
event EventHandler<AdsNotificationExEventArgs> AdsNotificationEx
```

**Value**

Type: `System.EventHandler<AdsNotificationExEventArgs>`

**Remarks**

The Notification event arguments marshals the data value automatically to the specified .NET Type with ANY_TYPE marshalers.

**Examples**

Example of receiving AdsNotificationEx events.

**Trigger on changed values by ADS Notifications**

```csharp
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
   client.AdsNotificationEx += Client_AdsNotificationEx;
   client.Connect(AmsNetId.Local, 851);
   // Add UDINT
   ResultHandle resultHandle = await client.AddDeviceNotificationExAsync("MAIN.udint", new NotificationSettings(AdsTransMode.OnChange, 200, 200), null, typeof(uint),null, cancel);
   await Task.Delay(5000, cancel); // Wait ....
   ResultAds resultHandleDelete = await client.DeleteDeviceNotificationAsync(resultHandle.Handle,cancel); // Unregister Event
}
```

**Reference**

IAdsNotifications Interface

TwinCAT.Ads Namespace

### 6.2.35.2.4 IAdsNotifications.AdsSumNotification Event

Occurs when Notifications are send (bundled notifications)

**Namespace:** TwinCAT.Ads namespace

**Assembly:** TwinCAT.Ads.Abstractions.dll (version 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14)

**Syntax**

C#  
```csharp
event EventHandler<AdsSumNotificationEventArgs> AdsSumNotification
```
Value

Type: `System.EventHandler<AdsSumNotificationEventArgs>`.

Remarks

As an optimization, this event receives all ADS Notifications that occurred at one point in time together. As a consequence, the overhead of handler code is reduced, what can be important if notifications are triggered in a high frequency and the event has to be synchronized to the UI thread context. Because multiple notifications are bound together, less thread synchronization is necessary. The `AdsNotification` and `AdsNotificationEx` events shouldn't be used when SumNotifications are registered, because they have an performance side effect to this AdsSumNotification event. The full performance is reached only, when all notifications are handled on this event.

Examples

Example of receiving AdsSumNotification events.

Trigger on changed values by ADS Notifications

```csharp
private async Task RegisterSumNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;
    using (AdsClient client = new AdsClient())
    {
        // Add the Notification event handler
        client.AdsSumNotification += Client_SumNotification;

        // Connect to target
        client.Connect(AmsNetId.Local, 851);
        uint notificationHandle = 0;

        // Notification to a DINT Type (UINT32)
        // Check for change every 200 ms
        byte[] notificationBuffer = new byte[sizeof(UInt32)];
        ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", sizeof(UInt32), new NotificationSettings(AdsTransModeOnChange, 200, 0), null, cancel);
        if (result.Succeeded)
        {
            notificationHandle = result.Handle;
            await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
            ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
        }
        client.AdsNotification -= Client_AdsNotification2;
    }
}

private void Client_SumNotification(object sender, AdsSumNotificationEventArgs e)
{
    // Timestamp of the Notification List
    DateTimeOffset dateTime = e.TimeStamp;

    // List of Raw ADS Notifications
    IList<Notification> notifications = e.Notifications;
    foreach (Notification notification in notifications)
    {
        // Notifications can be handled more efficiently, because they occur together
        // handler and can be transformed/synchronized in one step compared to AdsClient.AdsNotification events.
    }
}
```

Reference

`IAdsNotifications Interface` [839]
### 6.2.36 IAdsReadWrite Interface

Interface for ADS Read/Write access via IndexGroup / IndexOffset

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0e9a15da1c14

**Syntax**

**C#**

```csharp
public interface IAdsReadWrite
```

The `IAdsReadWrite` type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAsync</td>
<td>![ReadAsync][871]</td>
</tr>
<tr>
<td>ReadWriteAsync</td>
<td>![ReadWriteAsync][871]</td>
</tr>
<tr>
<td>TryRead</td>
<td>![TryRead][872]</td>
</tr>
<tr>
<td>TryReadWrite</td>
<td>![TryReadWrite][873]</td>
</tr>
<tr>
<td>TryWrite</td>
<td>![TryWrite][873]</td>
</tr>
<tr>
<td>WriteAsync</td>
<td>![WriteAsync][874]</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.Ads Namespace [151]

### 6.2.36.1 IAdsReadWrite Methods

The `IAdsReadWrite` ![870] type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAsync</td>
<td>![ReadAsync][871]</td>
</tr>
<tr>
<td>ReadWriteAsync</td>
<td>![ReadWriteAsync][871]</td>
</tr>
<tr>
<td>TryRead</td>
<td>![TryRead][872]</td>
</tr>
<tr>
<td>TryReadWrite</td>
<td>![TryReadWrite][873]</td>
</tr>
<tr>
<td>TryWrite</td>
<td>![TryWrite][873]</td>
</tr>
<tr>
<td>WriteAsync</td>
<td>![WriteAsync][874]</td>
</tr>
</tbody>
</table>
Reference

IAdsReadWrite Interface [870]
TwinCAT.Ads Namespace [151]

6.2.36.1.1 IAdsReadWrite.ReadAsync Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Task<ResultRead> ReadAsync(
    uint indexGroup,
    uint indexOffset,
    Memory buffer,
    void cancel
)
```

Parameters

indexGroup Type: System.UInt32
indexOffset Type: System.UInt32
buffer Type: Memory
cancel Type: System.Void

Return Value

Type: Task.ResultRead [1008].

Reference

IAdsReadWrite Interface [870]
TwinCAT.Ads Namespace [151]

6.2.36.1.2 IAdsReadWrite.ReadWriteAsync Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Task<ResultReadWrite> ReadWriteAsync(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte cancel
)
```
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>indexGroup</td>
<td>System.UInt32</td>
</tr>
<tr>
<td>indexOffset</td>
<td>System.UInt32</td>
</tr>
<tr>
<td>readBuffer</td>
<td>Memory</td>
</tr>
<tr>
<td>writeBuffer</td>
<td>System.Void</td>
</tr>
<tr>
<td>cancel</td>
<td>System.Byte</td>
</tr>
</tbody>
</table>

Return Value

Type: `Task.ResultReadWrite` [1019].

Reference

[IAdsReadWrite Interface] [870]

TwinCAT.Ads Namespace [151]

6.2.36.1.3  IAdsReadWrite.TryRead Method

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
AdsErrorCode TryRead(
    uint indexGroup,
    uint indexOffset,
    Memory buffer,
    void readBytes
)
```

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>indexGroup</td>
<td>System.UInt32</td>
</tr>
<tr>
<td>indexOffset</td>
<td>System.UInt32</td>
</tr>
<tr>
<td>buffer</td>
<td>Memory</td>
</tr>
<tr>
<td>readBytes</td>
<td>System.Void</td>
</tr>
</tbody>
</table>

Return Value

Type: `AdsErrorCode` [575]

Reference

[IAdsReadWrite Interface] [870]
6.2.36.1.4  IAdsReadWrite.TryReadWrite Method

Namespace:  TwinCAT.Ads
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
AdsErrorCode TryReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte readBytes
)
```

Parameters

- `indexGroup`: Type: `System.UInt32`
- `indexOffset`: Type: `System.UInt32`
- `readBuffer`: Type: `Memory`
- `writeBuffer`: Type: `System.Void`
- `readBytes`: Type: `System.Byte`

Return Value

Type: `AdsErrorCode`

Reference

- `IAdsReadWrite Interface`  
- `TwinCAT.Ads Namespace`

6.2.36.1.5  IAdsReadWrite.TryWrite Method

Namespace:  TwinCAT.Ads
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
AdsErrorCode TryWrite(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer
)
```
Parameters

- **indexGroup**: Type: `System.UInt32`
- **indexOffset**: Type: `System.UInt32`
- **writeBuffer**: Type: `ReadOnlyMemory`

Return Value

Type: `AdsErrorCode` [575]

Reference

- `IAdsReadWrite Interface` [870]
- `TwinCAT.Ads Namespace` [151]

### 6.2.36.1.6 IAdsReadWrite.WriteAsync Method

**Namespace**: `TwinCAT.Ads` [151]

**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
Task<ResultWrite> WriteAsync(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

Parameters

- **indexGroup**: Type: `System.UInt32`
- **indexOffset**: Type: `System.UInt32`
- **writeBuffer**: Type: `ReadOnlyMemory`
- **cancel**: Type: `System.Void`

Return Value

Type: `Task<ResultWrite>` [1032]

Reference

- `IAdsReadWrite Interface` [870]
- `TwinCAT.Ads Namespace` [151]
6.2.37  IAdsReadWrite2 Interface

Interface for ADS Read/Write access via IndexGroup / IndexOffset

Namespace:  TwinCAT.Ads
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public interface IAdsReadWrite2 : IAdsReadWrite

The IAdsReadWrite2 type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>(Inherited from IAdsReadWrite)</td>
</tr>
<tr>
<td>ReadAsync</td>
<td>(Inherited from IAdsReadWrite)</td>
</tr>
<tr>
<td>ReadWrite</td>
<td>(Inherited from IAdsReadWrite)</td>
</tr>
<tr>
<td>ReadWriteAsync</td>
<td>(Inherited from IAdsReadWrite)</td>
</tr>
<tr>
<td>TryRead</td>
<td>(Inherited from IAdsReadWrite)</td>
</tr>
<tr>
<td>TryReadWrite</td>
<td>(Inherited from IAdsReadWrite)</td>
</tr>
<tr>
<td>TryWrite</td>
<td>(Inherited from IAdsReadWrite)</td>
</tr>
<tr>
<td>Write(UInt32, UInt32)</td>
<td>Triggers a 'Write' call to the ADS device at the specified address.</td>
</tr>
<tr>
<td>Write(UInt32, UInt32, ReadOnlyMemory)</td>
<td></td>
</tr>
<tr>
<td>WriteAsync</td>
<td>(Inherited from IAdsReadWrite)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace

6.2.37.1  IAdsReadWrite2 Methods

The IAdsReadWrite2 type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>(Inherited from IAdsReadWrite)</td>
</tr>
<tr>
<td>ReadAsync</td>
<td>(Inherited from IAdsReadWrite)</td>
</tr>
<tr>
<td>ReadWrite</td>
<td>(Inherited from IAdsReadWrite)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadWriteAsync</td>
<td>(Inherited from IAdsReadWrite [871].)</td>
</tr>
<tr>
<td>TryRead [872]</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryReadWrite</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>TryWrite [873]</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
<tr>
<td>Write(UInt32, UInt32) [878]</td>
<td>Triggers a 'Write' call to the ADS device at the specified address.</td>
</tr>
<tr>
<td>Write(UInt32, UInt32, ReadOnlyMemory) [878]</td>
<td></td>
</tr>
<tr>
<td>WriteAsync [874]</td>
<td>(Inherited from IAdsReadWrite [870].)</td>
</tr>
</tbody>
</table>

Reference

IAdsReadWrite2 Interface [875]
TwinCAT.Ads Namespace [151]

6.2.37.1.1 IAdsReadWrite2.Read Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
int Read(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer
)
```

Parameters

- `indexGroup` Type: System.UInt32
- `indexOffset` Type: System.UInt32
- `readBuffer` Type: Memory

Return Value

Type: Int32

Reference

IAdsReadWrite2 Interface [875]
TwinCAT.Ads Namespace [151]
### 6.2.37.1.2 IAdsReadWrite2.ReadWrite Method

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
int ReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer
)
```

**Parameters**

- `indexGroup`  
  Type: System.UInt32
- `indexOffset`  
  Type: System.UInt32
- `readBuffer`  
  Type: Memory
- `writeBuffer`  
  Type: System.Void

**Return Value**

Type: Int32

**Reference**

- IAdsReadWrite2 Interface [875]
- TwinCAT.Ads Namespace [151]

### 6.2.37.1.3 IAdsReadWrite2.Write Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="write" /> Write(UInt32, UInt32) [878]</td>
<td>Triggers a 'Write' call to the ADS device at the specified address.</td>
</tr>
<tr>
<td><img src="image" alt="write" /> Write(UInt32, UInt32, ReadOnlyMemory) [878]</td>
<td><strong>Reference</strong></td>
</tr>
</tbody>
</table>

- IAdsReadWrite2 Interface [875]
- TwinCAT.Ads Namespace [151]
IAdsReadWrite2.Write Method (UInt32, UInt32)

Triggers a ‘Write’ call to the ADS device at the specified address.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
void Write(
    uint indexGroup,
    uint indexOffset
)
```

**Parameters**

- **indexGroup**
  - Type: System.UInt32
  - The index group.

- **indexOffset**
  - Type: System.UInt32
  - The index offset.

**Reference**

- IAdsReadWrite2 Interface [875]
- Write Overload [877]
- TwinCAT.Ads Namespace [151]

IAdsReadWrite2.Write Method (UInt32, UInt32, ReadOnlyMemory`1)

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
void Write(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer
)
```

**Parameters**

- **indexGroup**
  - Type: System.UInt32

- **indexOffset**
  - Type: System.UInt32

- **writeBuffer**
  - Type: ReadOnlyMemory

**Reference**

- IAdsReadWrite2 Interface [875]
6.2.38  IAdsReadWriteTimeoutAccess Interface

Interface IAdsReadWriteTimeoutAccess

Namespace:  TwinCAT.Ads [151]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IAdsReadWriteTimeoutAccess
```

The IAdsReadWriteTimeoutAccess type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read [880]</td>
<td></td>
</tr>
<tr>
<td>ReadAny [880]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadWrite [881]</td>
<td></td>
</tr>
<tr>
<td>TryRead [882]</td>
<td></td>
</tr>
<tr>
<td>TryReadWrite [883]</td>
<td></td>
</tr>
<tr>
<td>TryWrite [883]</td>
<td></td>
</tr>
<tr>
<td>Write [884]</td>
<td></td>
</tr>
<tr>
<td>WriteAny [885]</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
</tbody>
</table>

**Remarks**

For internal use only.

**Reference**

TwinCAT.Ads Namespace [151]

6.2.38.1  IAdsReadWriteTimeoutAccess Methods

The IAdsReadWriteTimeoutAccess [879] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read [880]</td>
<td></td>
</tr>
<tr>
<td>ReadAny [880]</td>
<td>Reads data synchronously from an ADS device and writes it to an object.</td>
</tr>
<tr>
<td>ReadWrite [881]</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryRead</td>
<td></td>
</tr>
<tr>
<td>TryReadWrite</td>
<td></td>
</tr>
<tr>
<td>TryWrite</td>
<td></td>
</tr>
<tr>
<td>Write</td>
<td></td>
</tr>
<tr>
<td>WriteAny</td>
<td>Writes an object synchronously to an ADS device.</td>
</tr>
</tbody>
</table>

**Reference**

IAdsReadWriteTimeoutAccess Interface [879]

TwinCAT.Ads Namespace [151]

### 6.2.38.1.1 IAdsReadWriteTimeoutAccess.Read Method

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
int Read(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void timeout
)
```

**Parameters**

- `indexGroup` Type: `System.UInt32`
- `indexOffset` Type: `System.UInt32`
- `readBuffer` Type: `Memory`
- `timeout` Type: `System.Void`

**Return Value**

Type: `Int32`

**Reference**

IAdsReadWriteTimeoutAccess Interface [879]

TwinCAT.Ads Namespace [151]

### 6.2.38.1.2 IAdsReadWriteTimeoutAccess.ReadAny Method

Reads data synchronously from an ADS device and writes it to an object.
Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args,
    int timeout
)
```

Parameters

- **indexGroup**
  - Type: System.UInt32
  - Index group of the ADS variable.

- **indexOffset**
  - Type: System.UInt32
  - Index offset of the ADS variable.

- **type**
  - Type: System.Type
  - Type of the object to be read.

- **args**
  - Type: System.Int32
  - Additional arguments.

- **timeout**
  - Type: System.Int32
  - The timeout.

Return Value

- Type: Object
  - The Value of the data marshalled to the specified type.

Remarks

If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args. The type is limited to Primitive types ("AnyType").

Reference

IAdsReadWriteTimeoutAccess Interface [879]
TwinCAT.Ads Namespace [151]

6.2.38.1.3 IAdsReadWriteTimeoutAccess.ReadWrite Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
int ReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
```
void writeBuffer, byte timeout
)

Parameters

indexGroup                Type: System.UInt32
indexOffset               Type: System.UInt32
readBuffer                Type: Memory
writeBuffer               Type: System.Void
timeout                   Type: System.Byte

Return Value

Type: Int32

Reference

IAdsReadWriteTimeoutAccess Interface [879]
TwinCAT.Ads Namespace [151]

6.2.38.1.4 IAdsReadWriteTimeoutAccess.TryRead Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

AdsErrorCode TryRead(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void timeout,
    byte readBytes
)

Parameters

indexGroup                Type: System.UInt32
indexOffset               Type: System.UInt32
readBuffer                Type: Memory
timeout                   Type: System.Void
readBytes                 Type: System.Byte
Return Value
Type: AdsErrorCode [575]

Reference
IAdsReadWriteTimeoutAccess Interface [879]
TwinCAT.Ads Namespace [151]

6.2.38.1.5 IAdsReadWriteTimeoutAccess.TryReadWrite Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#

AdsErrorCode TryReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte timeout,
    ReadOnlyMemory readBytes
)

Parameters
indexGroup Type: System.UInt32
indexOffset Type: System.UInt32
readBuffer Type: Memory
writeBuffer Type: System Void
timeout Type: System.Byte
readBytes Type: ReadOnlyMemory

Return Value
Type: AdsErrorCode [575]

Reference
IAdsReadWriteTimeoutAccess Interface [879]
TwinCAT.Ads Namespace [151]

6.2.38.1.6 IAdsReadWriteTimeoutAccess.TryWrite Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

AdsErrorCode TryWrite(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer,
    void timeout
)

Parameters

indexGroup Type: System.UInt32
indexOffset Type: System.UInt32
writeBuffer Type: ReadOnlyMemory
timeout Type: System.Void

Return Value

Type: AdsErrorCode

Reference

IAdsReadWriteTimeoutAccess Interface

TwinCAT.Ads Namespace

6.2.38.1.7 IAdsReadWriteTimeoutAccess.Write Method

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

void Write(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer,
    void timeout
)

Parameters

indexGroup Type: System.UInt32
indexOffset Type: System.UInt32
writeBuffer Type: ReadOnlyMemory
timeout Type: System.Void
6.2.38.1.8  IAdsReadWriteTimeoutAccess.WriteAny Method

Writes an object synchronously to an ADS device.

**Namespace:**  TwinCAT.Ads [151]

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args,
    int timeout
)
```

**Parameters**

- **indexGroup**
  - Type: `System.UInt32`
  - Contains the index group number of the requested ADS service.

- **indexOffset**
  - Type: `System.UInt32`
  - Contains the index offset number of the requested ADS service.

- **value**
  - Type: `System.Object`
  - Object to write to the ADS device (Primitive type, 'AnyType')

- **args**
  - Type: `System.Int32`
  - Additional arguments.

- **timeout**
  - Type: `System.Int32`
  - The timeout.

**Return Value**

Type: `System.Int32`.

**Remarks**

If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. If the Type of the object to be written is an array type, the number of elements for each dimension has to be specified in the parameter args. The value is limited to Primitive types ('AnyType').

**Reference**

IAdsReadWriteTimeoutAccess Interface [879]

TwinCAT.Ads Namespace [151]
### 6.2.39 IAdsRpcInvoke Interface

Interface IAdsRpcInvoke

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public interface IAdsRpcInvoke
```

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvokeRpcMethod(String, String, Object.)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object., Object.)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object., AnyTypeSpecifier., AnyTypeSpecifier., Object.)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, String, Object., CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, String, Object., AnyTypeSpecifier., AnyTypeSpecifier., CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| InvokeRpcMethodA  

sync(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken) | Invokes the specified RPC Method asynchronously |
| TryInvokeRpcMethod(String, String, Object, Object) | Invokes the specified RPC Method |
| TryInvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object) | Invokes the rpc method. |
| TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object) | Invokes the rpc method. |

**Remarks**

This interface is used to invoke ADS RPC Method calls. This can be done 'symbolic' via the Method name or the MethodID of the method on the specified symbol. To activate the RPC Access within the PLC environment, its dataType (Structure, FB) must be marked with the PlcAttribute 'TcRpcEnable' to enable RpcMethods.

**Reference**

TwinCAT.Ads Namespace [151]
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvokeRpcMethod(String, String, Object.)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object., Object.)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object., Any TypeSpecifier, AnyTypeSpecifier, Object.)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, String, Object., CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, String, Object., Any TypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, Object., AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object.)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, Object) ▶ 903</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object) ▶ 905</td>
<td>Invokes the rpc method.</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object) ▶ 906</td>
<td>Invokes the rpc method.</td>
</tr>
</tbody>
</table>

Reference

IAdsRpcInvoke Interface ▶ 886

TwinCAT.Ads Namespace ▶ 151

6.2.39.1.1 IAdsRpcInvoke.InvokeRpcMethod Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvokeRpcMethod(String, String, Object) ▶ 890</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object, Object) ▶ 891</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object) ▶ 893</td>
<td>Invokes the specified RPC Method</td>
</tr>
</tbody>
</table>
IAdsRpcInvoke.InvokeRpcMethod Method (String, String, .Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters
)
```

Parameters

- **symbolPath**
  - Type: `System.String`
  - The symbol path.

- **methodName**
  - Type: `System.String`
  - The method name.

- **inParameters**
  - Type: `System.Object`
  - The input parameters or NULL

Return Value

Type: `Object`
The return value of the Method (as object).

Remarks

This method only supports primitive data types as inParameters. Any available outparameters will be ignored. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        using (AdsClient client = new AdsClient())
        {
            // Connect to the target device
client.Connect(address);

SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

// Get the Symbols (Dynamic Symbols)

IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

// Call a Method that has the following signature (within MAIN Program)
/*
   {attribute 'TcRpcEnable'}
   METHOD PUBLIC M_Add : INT
   VAR_INPUT
   i1 : INT := 0;
   i2 : INT := 0;
   END_VAR
   */
short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short)3, (short)4});

// Call a Method that has no parameter and returns VOID
main.InvokeRpcMethod("M_Method1", new object[]{});

// Browsing RpcMethods
foreach(IRpcMethod method in main.RpcMethods)
{
    string methodName = method.Name;
    foreach(IRpcMethodParameter parameter in method.Parameters)
    {
        string parameterName = parameter.Name;
        string parameterType = parameter.TypeName;
    }
}

Reference

IAdsRpcInvoke Interface [886]
InvokeRpcMethod Overload [889]
TwinCAT.Ads Namespace [151]

IAdsRpcInvoke.InvokeRpcMethod Method (String, String, .Object., .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object[] outParameters
)

Parameters

symbolPath Type: System.String
The symbol path.
methodName

Type: System.String

The method name.

inParameters

Type: System.Object

The input parameters or NULL

outParameters

Type: System.Object

The output parameters.

Return Value

Type: Object

The return value of the Method (as object).

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /* (attribute 'TcRpcEnable')
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
                i1 : INT := 0;
                i2 : INT := 0;
            END_VAR
            */

            short result = (short)main.InvokeRpcMethod("M_Add", new object[] { (short)3, (short)4 });

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] { });

            // Browsing RpcMethods
            foreach (IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;

                foreach (IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}
Reference

IAdsRpcInvoke Interface [» 886]
InvokeRpcMethod Overload [» 889]
TwinCAT.Ads Namespace [» 151]

IAdsRpcInvoke.InvokeRpcMethod Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [» 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters
)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbolPath</td>
<td>System.String</td>
<td>The symbol path.</td>
</tr>
<tr>
<td>methodName</td>
<td>System.String</td>
<td>The method name.</td>
</tr>
<tr>
<td>inParameters</td>
<td>System.Object</td>
<td>The parameters.</td>
</tr>
<tr>
<td>outSpecifiers</td>
<td>TwinCAT.TypeSystem.AnyTypeSpecifier[]</td>
<td>The out specifiers (specifying the out types) or NULL.</td>
</tr>
<tr>
<td>retSpecifier</td>
<td>TwinCAT.TypeSystem.AnyTypeSpecifier[]</td>
<td>The ret specifier (specifying the return value) or NULL.</td>
</tr>
<tr>
<td>outParameters</td>
<td>System.Object</td>
<td>The out parameters.</td>
</tr>
</tbody>
</table>

Return Value

Type: Object
The return value of the Method (as object).
Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(Symbol�LoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /* {attribute 'TcRpcEnable'}
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
                i1 : INT := 0;
                i2 : INT := 0;
            END_VAR
            */
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short)3, (short)4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] { });

            // Browsing RpcMethods
            foreach(IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;
                foreach(IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}

Reference

IAdsRpcInvoke Interface [886]
InvokeRpcMethod Overload [889]
6.2.39.1.2  IAdsRpcInvoke.InvokeRpcMethodAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvokeRpcMethodAsync(String, String, Object, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, String, Object, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, Object, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
</tbody>
</table>

Reference

IAdsRpcInvoke Interface [886]

TwinCAT.Ads Namespace [151]

IAdsRpcInvoke.InvokeRpcMethodAsync Method (String, String, .Object., CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace:  TwinCAT.Ads [151]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

Task<ResultRpcMethod> InvokeRpcMethodAsync(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    CancellationToken cancel
)
Parameters

- **symbolPath**
  - **Type:** System.String
  - The symbol/Instance path of the symbol.

- **methodName**
  - **Type:** System.String
  - The method name.

- **inParameters**
  - **Type:** System.Object
  - The parameters.

- **cancel**
  - **Type:** System.Threading.CancellationToken
  - The cancellation token

Return Value

- **Type:** Task<ResultRpcMethod>{1025}
  - A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethod{1025} results contain the return value together with the output parameters.

Remarks

Because this overload doesn't provide any AnyTypeSpecifier{1633} specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```csharp
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IIRpcStructInstance main = (IIRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /* (attribute 'TcRpcEnable')
               METHOD PUBLIC M_Add : INT
             VAR_INPUT
             i1 : INT := 0;
             i2 : INT := 0;
             END_VAR
            */

            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});

            // Browsing RpcMethods
            foreach (IRpcMethod method in main.RpcMethods)
            {
            }
        }
    }
}
```
```csharp
{
    string methodName = method.Name;
    foreach (IRpcMethodParameter parameter in method.Parameters)
    {
        string parameterName = parameter.Name;
        string parameterType = parameter.TypeName;
    }
}
```

**Reference**

IAdsRpcInvoke Interface  
InvokeRpcMethodAsync Overload

**TwinCAT.Ads Namespace**

IAdsRpcInvoke.InvokeRpcMethodAsync Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken)

Invokes the specified RPC Method asynchronously

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdc3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
Task<ResultRpcMethod> InvokeRpcMethodAsync(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    CancellationToken cancel
)
```

**Parameters**

- **symbolPath**
  - Type: System.String
  - The symbol/Instance path of the symbol.

- **methodName**
  - Type: System.String
  - The method name.

- **inParameters**
  - The parameters.

- **outSpecifiers**
  - Type: .TwinCAT.TypeSystem.AnyTypeSpecifier
  - The out specifiers (specifying the out types) or NULL.

- **retSpecifier**
  - Type: .TwinCAT.TypeSystem.AnyTypeSpecifier
  - The ret specifier (specifying the return value) or NULL.

- **cancel**
  - Type: .System.Threading.CancellationToken
  - The cancellation token

**Return Value**

Type: .Task<ResultRpcMethod>

A task that represents the asynchronous ‘InvokeRpcMethod’ operation. The ResultRpcMethod results contains the return value together with the output parameters.
Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.ReturnValue [1028] and the ErrorCode [992] of the ADS communication after execution.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        using (AdsClient client = new AdsClient())
        {
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"];// Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /* (attribute 'TcRpcEnable')
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
            END_VAR
            */

            short result = (short)main.InvokeRpcMethod("M_Add", new object[]{3, 4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[]{})

            // Browsing RpcMethods
            foreach (IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;

                foreach (IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}

Reference

IAdsRpcInvoke Interface [886]
InvokeRpcMethodAsync Overload [895]
TwinCAT.Ads Namespace [► 151]

IAdsRpcInvoke.InvokeRpcMethodAsync Method (IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.Ads [► 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Task<ResultRpcMethod> InvokeRpcMethodAsync(
    IRpcCallableInstance symbol,
    IRpcMethod method,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    CancellationToken cancel)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbol</td>
<td>TwinCAT.TypeSystem.IRpcCallableInstance</td>
<td>The RPC callable symbol.*</td>
</tr>
<tr>
<td>method</td>
<td>TwinCAT.TypeSystem.IRpcMethod</td>
<td>The method.</td>
</tr>
<tr>
<td>inParameters</td>
<td>System.Object</td>
<td>The parameters.</td>
</tr>
<tr>
<td>outSpecifiers</td>
<td>TwinCAT.TypeSystem.AnyTypeSpecifier</td>
<td>The out specifiers (specifying the out types) or NULL.</td>
</tr>
<tr>
<td>retSpecifier</td>
<td>TwinCAT.TypeSystem.AnyTypeSpecifier</td>
<td>The ret specifier (specifying the return value) or NULL.</td>
</tr>
<tr>
<td>cancel</td>
<td>System.Threading.CancellationToken</td>
<td>The cancellation token</td>
</tr>
</tbody>
</table>

Return Value

Type: Task<ResultRpcMethod> [► 1025].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethod [► 1025] results contains the return value together with the output parameters.

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set. The ReturnValue [► 1028] and the ErrorCode [► 992] of the ADS communication after execution.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.
Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program
            // Call a Method that has the following signature (within MAIN Program)
            /*
               * (attribute 'TcRpcEnable')
               METHOD PUBLIC M_Add : INT
               VAR_INPUT
               i1 : INT := 0;
               i2 : INT := 0;
               END_VAR
               */
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});
            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});
            //Browsing RpcMethods
            foreach (IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;
                foreach (IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}

Reference

IAdsRpcInvoke Interface [886]
InvokeRpcMethodAsync Overload [895]
TwinCAT.Ads Namespace [151]
# IAdsRpcInvoke.TryInvokeRpcMethod Method

## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon] TryInvokeRpcMethod(String, String, Object, Object) ![901]</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>![Icon] TryInvokeRpcMethod(String, String, Object, Object) ![903]</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>![Icon] TryInvokeRpcMethod(String, String, Any TypeSpecifier, Any TypeSpecifier, Object, Object) ![905]</td>
<td>Invokes the rpc method.</td>
</tr>
<tr>
<td>![Icon] TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, Object, Any TypeSpecifier, Any TypeSpecifier, Object, Object) ![906]</td>
<td>Invokes the rpc method.</td>
</tr>
</tbody>
</table>

## Reference

- `IAdsRpcInvoke` Interface ![886]
- `TwinCAT.Ads Namespace` ![151]

## IAdsRpcInvoke.TryInvokeRpcMethod Method (String, String, Object, Object.)

Invokes the specified RPC Method

**Namespace:** `TwinCAT.Ads` ![151]

**Assembly:** `TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    string parameter1,
    ...
)
```
Parameters

symbolPath  Type: System.String
The symbol path.

methodName  Type: System.String
The method name.

inParameters  Type: .System.Object.
The parameters.

retValue  Type: System.Object.
The return value of the RPC method as object.

Return Value

Type: AdsErrorCode [575]
The ADS Error Code.

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /* {attribute 'TcRpcEnable'}
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
            END_VAR
            */

            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});
        }
    }
}
```csharp
// Browsing RpcMethods
foreach (IRpcMethod method in main.RpcMethods)
{
    string methodName = method.Name;
    foreach (IRpcMethodParameter parameter in method.Parameters)
    {
        string parameterName = parameter.Name;
        string parameterType = parameter.TypeName;
    }
}
}
}
```

**Reference**

[IAdsRpcInvoke Interface](#)

[TryInvokeRpcMethod Overload](#)

[TwinCAT.Ads Namespace](#)

### IAdsRpcInvoke.TryInvokeRpcMethod Method (String, String, Object, Object, Object)

Invokes the specified RPC Method

**Namespace:** [TwinCAT.Ads](#)  
**Assembly:** [TwinCAT.Ads.Abstractions](#) (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```
AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object[] outParameters,
    out Object retValue
)
```

**Parameters**

- **symbolPath**  
  Type: `System.String`  
  The symbol path.

- **methodName**  
  Type: `System.String`  
  The method name.

- **inParameters**  
  Type: `System.Object`  
  The parameters.

- **outParameters**  
  Type: `System.Object`  
  The out parameters.

- **retValue**  
  Type: `System.Object`  
  The return value of the RPC method as object.

**Return Value**

Type: `AdsErrorCode`  
The ADS Error Code.
Remarks

Because this overload doesn’t provide any AnyTypeSpecifier [1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCall1VirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        using (AdsClient client = new AdsClient())
        {
            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /*
             * METHOD PUBLIC M_Add : INT
             * VAR_INPUT
             *     i1 : INT := 0;
             *     i2 : INT := 0;
             * END_VAR
             */

            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short)3, (short)4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});

            // Browsing RpcMethods
            foreach(IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;

                foreach(IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}

Reference

IAdsRpcInvoke Interface [886]
TryInvokeRpcMethod Overload [901]
TwinCAT.Ads Namespace [151]
IAdsRpcInvoke.TryInvokeRpcMethod Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.)

Invokes the rpc method.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

- **symbolPath**
  Type: System.String
  The symbol.

- **methodName**
  Type: System.String
  Name of the method.

- **inParameters**
  Type: .System.Object
  The parameters.

- **outSpecifiers**
  Type: .TwinCAT.TypeSystem.AnyTypeSpecifier
  The out specifiers (specifying the out types) or NULL.

- **retSpecifier**
  Type: TwinCAT.TypeSystem.AnyTypeSpecifier
  The ret specifier (specifying the return value) or NULL.

- **outParameters**
  Type: .System.Object
  The out parameters.

- **retValue**
  Type: System.Object
  The return value of the RPC method.

Return Value

Type: AdsErrorCode [575]
AdsErrorCode.

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

**Dynamic Tree Mode**

```csharp
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
```
/// </summary>
/// <param name="args">The arguments.</param>
static void Main(string[] args)
{
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)
        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"];
        // Gets the MAIN Instance of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /* {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */
        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        // Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
            string methodName = method.Name;

            foreach(IRpcMethodParameter parameter in method.Parameters)
            {
                string parameterName = parameter.Name;
                string parameterType = parameter.TypeName;
            }
        }
    }
}

Reference

IAdsRpcInvoke Interface [886]
TryInvokeRpcMethod Overload [901]
TwinCAT.Ads Namespace [151]


Invokes the rpc method.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
AdsErrorCode TryInvokeRpcMethod(
    IRpcCallableInstance symbol,
    IRpcMethod method,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

- **symbol**
  - Type: TwinCAT.TypeSystem.IRpcCallableInstance
  - The RPC callable symbol

- **method**
  - Type: TwinCAT.TypeSystem.IRpcMethod
  - The method.

- **inParameters**
  - Type: System.Object
  - The parameters.

- **outSpecifiers**
  - Type: TwinCAT.TypeSystem.AnyTypeSpecifier
  - The out specifiers (specifying the out types) or NULL.

- **retSpecifier**
  - Type: TwinCAT.TypeSystem.AnyTypeSpecifier
  - The ret specifier (specifying the return value) or NULL.

- **outParameters**
  - Type: System.Object
  - The out parameters.

- **retValue**
  - Type: System.Object
  - The return value of the RPC method.

Return Value

Type: AdsErrorCode

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```csharp
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
        }
    }
}
```
client.Connect(address);

SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolLoadMode.VirtualTree);
ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

// Get the Symbols (Dynamic Symbols)
IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

// Call a Method that has the following signature (within MAIN Program)
    /* (attribute 'TcRpcEnable')
    METHOD PUBLIC M_Add : INT
    VAR_INPUT
    i1 : INT := 0;
    i2 : INT := 0;
    END_VAR
    */
short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short) 3, (short) 4});

// Call a Method that has no parameter and returns VOID
main.InvokeRpcMethod("M_Method1", new object[] {});

// Browsing RpcMethods
foreach(IRpcMethod method in main.RpcMethods)
{
    string methodName = method.Name;
    foreach(IRpcMethodParameter parameter in method.Parameters)
    {
        string parameterName = parameter.Name;
        string parameterType = parameter.TypeName;
    }
}

---

**Reference**

*IAdsRpcInvoke Interface* [► 886]

*TryInvokeRpcMethod Overload* [► 901]

*TwinCAT.Ads Namespace* [► 151]

### 6.2.40 IAdsSession Interface

Interface IAdsSession

**Namespace:** TwinCAT.Ads [► 151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

*C#*

```csharp
public interface IAdsSession : ISession, IConnectionStateProvider, ISymbolServerProvider
```

The IAdsSession type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Gets the Address specifier of the Session / connection</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AddressSpecifier</td>
<td>Gets the communication endpoint address string representation. (Inherited from ISession[98])</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the Connection object. (Inherited from ISession[98])</td>
</tr>
<tr>
<td>ConnectionState</td>
<td>Gets the current Connection state of the IConnectionStateProvider[97] (Inherited from IConnectionStateProvider[97])</td>
</tr>
<tr>
<td>EstablishedAt</td>
<td>Gets the UTC time when the session was established. (Inherited from ISession[98])</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the Session Id (Inherited from ISession[98])</td>
</tr>
<tr>
<td>IsConnected</td>
<td>Gets a value indicating whether the session is connected. (Inherited from ISession[98])</td>
</tr>
<tr>
<td>NetId</td>
<td>Gets the NetId of the Session</td>
</tr>
<tr>
<td>Owner</td>
<td>Gets the Session owner.</td>
</tr>
<tr>
<td>Port</td>
<td>Gets the Ams Port of the Session</td>
</tr>
<tr>
<td>Provider</td>
<td>Gets the Session Provider (Inherited from ISession[98])</td>
</tr>
<tr>
<td>SymbolServer</td>
<td>Gets the symbol server. (Inherited from ISymbolServerProvider[99])</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Closes this ISession[98] (Inherited from ISession[98])</td>
</tr>
<tr>
<td>Connect</td>
<td>Connects the session and returns the established IConnection[74] object. (Inherited from ISession[98])</td>
</tr>
<tr>
<td>Disconnect</td>
<td>Disconnects the ISession[98] (Inherited from ISession[98])</td>
</tr>
</tbody>
</table>

**Events**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionStateChanged</td>
<td>Occurs when connection status of the IConnectionStateProvider[97] has been changed. (Inherited from IConnectionStateProvider[97])</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.Ads Namespace[151]

TwinCAT.ISession[88]

### 6.2.40.1 IAdsSession Properties

The IAdsSession[908] type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address ([910])</td>
<td>Gets the Address specifier of the Session / connection</td>
</tr>
<tr>
<td>AddressSpecifier ([89])</td>
<td>Gets the communication endpoint address string representation. (Inherited from [Session ([88]).)</td>
</tr>
<tr>
<td>Connection ([90])</td>
<td>Gets the Connection object. (Inherited from [Session ([88]).)</td>
</tr>
<tr>
<td>ConnectionState ([85])</td>
<td>Gets the current Connection state of the [ConnectionStateProvider ([84])](Inherited from [ConnectionStateProvider ([84]).)</td>
</tr>
<tr>
<td>EstablishedAt ([901])</td>
<td>Gets the UTC time when the session was established. (Inherited from [Session ([88]).)</td>
</tr>
<tr>
<td>Id ([91])</td>
<td>Gets the Session Id (Inherited from [Session ([88]).)</td>
</tr>
<tr>
<td>IsConnected ([91])</td>
<td>Gets a value indicating whether the session is connected. (Inherited from [Session ([88]).)</td>
</tr>
<tr>
<td>NetId ([911])</td>
<td>Gets the NetId of the Session</td>
</tr>
<tr>
<td>Owner ([911])</td>
<td>Gets the Session owner.</td>
</tr>
<tr>
<td>Port ([911])</td>
<td>Gets the Ams Port of the Session</td>
</tr>
<tr>
<td>Provider ([91])</td>
<td>Gets the Session Provider (Inherited from [Session ([88]).)</td>
</tr>
<tr>
<td>SymbolServer ([100])</td>
<td>Gets the symbol server. (Inherited from [SymbolServerProvider ([99]).)</td>
</tr>
</tbody>
</table>

### Reference

- [IAdsSession Interface ([908])](#)
- [TwinCAT.Ads Namespace ([151])](#)

### 6.2.40.1.1 IAdsSession.Address Property

Gets the Address specifier of the Session / connection

**Namespace:** TwinCAT.Ads ([151])

**Assembly:** TwinCAT.Ads.ABSTRACTIONS (in TwinCAT.Ads.ABSTRACTIONS.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
AmsAddress Address { get; }
```

#### Property Value

- **Type:** AmsAddress ([648])
- The address.

### Reference

- [IAdsSession Interface ([908])](#)
6.2.40.1.2  IAdsSession.NetId Property

Gets the NetId of the Session

Namespace:  TwinCAT.Ads
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
AmsNetId NetId { get; }

Property Value

Type:  AmsNetId
The net identifier.

Reference

IAdsSession Interface
TwinCAT.Ads Namespace

6.2.40.1.3  IAdsSession.Owner Property

Gets the Session owner.

Namespace:  TwinCAT.Ads
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
Object Owner { get; }

Property Value

Type:  Object
The owner or NULL

Reference

IAdsSession Interface
TwinCAT.Ads Namespace

6.2.40.1.4  IAdsSession.Port Property

Gets the Ams Port of the Session

Namespace:  TwinCAT.Ads
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
int Port { get; }
```

Property Value

Type: Int32
The port.

Reference

IAdsSession Interface [908]
TwinCAT.Ads Namespace [151]

6.2.40.2 IAdsSession Methods

The IAdsSession [908] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="92" alt="Close" /></td>
<td>Closes this ISession [88] (Inherited from ISession [88].)</td>
</tr>
<tr>
<td><img src="93" alt="Connect" /></td>
<td>Connects the session and returns the established IConnection [74] object.</td>
</tr>
<tr>
<td><img src="93" alt="Disconnect" /></td>
<td>Disconnects the ISession [88] (Inherited from ISession [88].)</td>
</tr>
</tbody>
</table>

Reference

IAdsSession Interface [908]
TwinCAT.Ads Namespace [151]

6.2.40.3 IAdsSession Events

The IAdsSession [908] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="86" alt="ConnectionStateChanged" /></td>
<td>Occurs when connection status of the IConnectionStateChangedProvider [84] has</td>
</tr>
<tr>
<td><img src="86" alt="ConnectionStateChanged" /></td>
<td>been changed. (Inherited from IConnectionStateChangedProvider [84].)</td>
</tr>
</tbody>
</table>

Reference

IAdsSession Interface [908]
TwinCAT.Ads Namespace [151]
6.2.41 IAdsSessionSettings Interface

Interface for ADS Session Settings

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public interface IAdsSessionSettings
```

The `IAdsSessionSettings` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResurrectionTime</td>
<td>Gets or sets the resurrection time.</td>
</tr>
<tr>
<td>SymbolLoader</td>
<td>Gets or sets the symbol loader settings</td>
</tr>
<tr>
<td>Timeout</td>
<td>Gets the ADS timeout in milliseconds.</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.Ads Namespace

IAdsSessionSettings

### 6.2.41.1 IAdsSessionSettings Properties

The `IAdsSessionSettings` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResurrectionTime</td>
<td>Gets or sets the resurrection time.</td>
</tr>
<tr>
<td>SymbolLoader</td>
<td>Gets or sets the symbol loader settings</td>
</tr>
<tr>
<td>Timeout</td>
<td>Gets the ADS timeout in milliseconds.</td>
</tr>
</tbody>
</table>

**Reference**

IAdsSessionSettings Interface

TwinCAT.Ads Namespace

### 6.2.41.1.1 IAdsSessionSettings.ResurrectionTime Property

Gets or sets the resurrection time.
Syntax

C#

```csharp
TimeSpan ResurrectionTime { get; set; }
```

Property Value

Type: `TimeSpan`

The resurrection time.

Reference

`IAdsSessionSettings Interface [913]`

`TwinCAT.Ads Namespace [151]`

### 6.2.41.1.2 `IAdsSessionSettings.SymbolLoader` Property

Gets or sets the symbol loader settings

**Namespace:** `TwinCAT.Ads [151]`

**Assembly:** `TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`

Syntax

C#

```csharp
SymbolLoaderSettings SymbolLoader { get; set; }
```

Property Value

Type: `SymbolLoaderSettings [140]`

The symbol loader.

Reference

`IAdsSessionSettings Interface [913]`

`TwinCAT.Ads Namespace [151]`

### 6.2.41.1.3 `IAdsSessionSettings.Timeout` Property

Gets the ADS timeout in milliseconds.

**Namespace:** `TwinCAT.Ads [151]`

**Assembly:** `TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`

Syntax

C#

```csharp
int Timeout { get; }
```
Property Value

Type: Int32
The timeout.

Reference

IAdsSessionSettings Interface [913]
TwinCAT.Ads Namespace [151]

6.2.42 IAdsStateControl Interface

Interface for reading and controlling the ADS state.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsStateControl
```

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ] TryWriteControl(StateInfo) [917]</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>![ ] TryWriteControl(StateInfo, ReadOnlyMemory) [917]</td>
<td></td>
</tr>
<tr>
<td>![ ] WriteControl(StateInfo) [918]</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>![ ] WriteControl(StateInfo, ReadOnlyMemory) [919]</td>
<td></td>
</tr>
<tr>
<td>![ ] WriteControlAsync(AdsState, UInt16, CancellationToken) [920]</td>
<td>Changes the ADS status and device status of the ADS server asynchronously.</td>
</tr>
<tr>
<td>![ ] WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [921]</td>
<td></td>
</tr>
</tbody>
</table>
## 6.2.42.1 IAdsStateControl Methods

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryWriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, ReadOnlyMemory)</td>
<td></td>
</tr>
<tr>
<td>WriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>WriteControl(StateInfo, ReadOnlyMemory)</td>
<td></td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, CancellationToken)</td>
<td>Changes the ADS status and device status of the ADS server asynchronously.</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void)</td>
<td></td>
</tr>
</tbody>
</table>

## Reference

IAdsStateControl Interface [915]

TwinCAT.Ads Namespace [151]

### 6.2.42.1.1 IAdsStateControl.TryWriteControl Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryWriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, ReadOnlyMemory)</td>
<td>New ADS status and device status.</td>
</tr>
</tbody>
</table>

**Reference**

IAdsStateControl Interface [915]

TwinCAT.Ads Namespace [151]

**IAdsStateControl.TryWriteControl Method (StateInfo)**

Changes the ADS status and the device status of an ADS server.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
AdsErrorCode TryWriteControl(
    StateInfo stateInfo
)
```

**Parameters**

stateInfo  
Type: TwinCAT.Ads.StateInfo [1041]  
New ADS status and device status.

**Return Value**

Type: AdsErrorCode [575]  
AdsErrorCode.

**Reference**

IAdsStateControl Interface [915]  
TryWriteControl Overload [916]  
TwinCAT.Ads Namespace [151]

**IAdsStateControl.TryWriteControl Method (StateInfo, ReadOnlyMemory`1)**

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
## Syntax

**C#**

```csharp
AdsErrorCode TryWriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer
)
```

### Parameters

- `stateInfo`: Type: `TwinCAT.Ads.StateInfo`[^1041]
- `writeBuffer`: Type: `ReadOnlyMemory`

### Return Value

Type: `AdsErrorCode`[^575]

### Reference

- **IAdsStateControl Interface**[^915]
- **TryWriteControl Overload**[^916]
- **TwinCAT.Ads Namespace**[^151]

## 6.2.42.1.2  IAdsStateControl.WriteControl Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteControl(StateInfo)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>WriteControl(StateInfo, ReadOnlyMemory)</td>
<td></td>
</tr>
</tbody>
</table>

### Reference

- **IAdsStateControl Interface**[^915]
- **TwinCAT.Ads Namespace**[^151]

## IAdsStateControl.WriteControl Method (StateInfo)

Changes the ADS status and the device status of an ADS server.

**Namespace**: TwinCAT.Ads[^151]

**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
void WriteControl(
    StateInfo stateInfo
)
```

Parameters

- `stateInfo` Type: `TwinCAT.Ads.StateInfo` [1041]
  New ADS status and device status.

Reference

- IAdsStateControl Interface [915]
- WriteControl Overload [918]
- TwinCAT.Ads Namespace [151]

IAdsStateControl.WriteControl Method (StateInfo, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
void WriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer
)
```

Parameters

- `stateInfo` Type: `TwinCAT.Ads.StateInfo` [1041]
- `writeBuffer` Type: `ReadOnlyMemory`

Reference

- IAdsStateControl Interface [915]
- WriteControl Overload [918]
- TwinCAT.Ads Namespace [151]
### 6.2.42.1.3 IAdsStateControl.WriteControlAsync Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteControlAsync(AdsState, UInt16, CancellationToken)</td>
<td>Changes the ADS status and device status of the ADS server asynchronously.</td>
</tr>
<tr>
<td>WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void)</td>
<td></td>
</tr>
</tbody>
</table>

#### Reference

IAdsStateControl Interface [915]

TwinCAT.Ads Namespace [151]

### IAdsStateControl.WriteControlAsync Method (AdsState, UInt16, CancellationToken)

Changes the ADS status and device status of the ADS server asynchronously.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
Task<ResultAds> WriteControlAsync(
    AdsState state,
    ushort deviceState,
    CancellationToken cancel
)
```

#### Parameters

- **state**
  - Type: TwinCAT.Ads.AdsState [626]
  - The ADS state.

- **deviceState**
  - Type: System.UInt16
  - The device state.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

#### Return Value

Type: Task<ResultAds> [989].

A task that represents the asynchronous 'WriteControl' operation. The ResultAds [989] parameter contains the state the ErrorCode [992] of the ADS communication after execution.

#### Reference

IAdsStateControl Interface [915]
WriteControlAsync Overload [920]

TwinCAT.Ads Namespace [151]

**IAdsStateControl.WriteControlAsync Method (AdsState, UInt16, ReadOnlyMemory`1, Void)**

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
Task<ResultAds> WriteControlAsync(
    AdsState state,
    ushort deviceState,
    ReadOnlyMemory data,
    void cancel
)
```

**Parameters**

- **state**
  - Type: TwinCAT.Ads.AdsState [626]
- **deviceState**
  - Type: System.UInt16
- **data**
  - Type: ReadOnlyMemory
- **cancel**
  - Type: System.Void

**Return Value**

Type: Task.ResultAds [989].

**Reference**

IAdsStateControl Interface [915]
WriteControlAsync Overload [920]
TwinCAT.Ads Namespace [151]

6.2.43  **IAdsStateControlTimeout Interface**

Interface IAdsStateControlTimeout

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public interface IAdsStateControlTimeout
```

---

TC1000  
Version: 1.1  
921
The IAdsStateControlTimeout type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadState</td>
<td>Reads the ADS status and the device status from an ADS server.</td>
</tr>
<tr>
<td>TryReadState</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, Int32)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, ReadOnlyMemory, Void)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>WriteControl(StateInfo, Int32)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>WriteControl(StateInfo, ReadOnlyMemory, Void)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
</tbody>
</table>

**Remarks**

For internal use only.

**Reference**

TwinCAT.Ads Namespace [151]

### 6.2.43.1 IAdsStateControlTimeout Methods

The IAdsStateControlTimeout [921] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadState</td>
<td>Reads the ADS status and the device status from an ADS server.</td>
</tr>
<tr>
<td>TryReadState</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.</td>
</tr>
<tr>
<td>TryWriteControl(StateInfo, Int32)</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
</tbody>
</table>
### TryWriteControl

Changes the ADS status and the device status of an ADS server.

### WriteControl

Changes the ADS status and the device status of an ADS server.

---

#### Reference

IAdsStateControlTimeout Interface [921]

TwinCAT.Ads Namespace [151]

---

### 6.2.43.1.1 IAdsStateControlTimeout.ReadState Method

Reads the ADS status and the device status from an ADS server.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
StateInfo ReadState(int timeout)
```

**Parameters**

- `timeout` : Type: System.Int32
  - The timeout.

**Return Value**

- Type: StateInfo [1041]
  - The ADS statue and device status.

---

#### Reference

IAdsStateControlTimeout Interface [921]

TwinCAT.Ads Namespace [151]
6.2.43.1.2 IAdsStateControlTimeout.TryReadState Method

Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294*Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
AdsErrorCode TryReadState(
    int timeout,
    out StateInfo stateInfo
)
```

**Parameters**

- **timeout**
  - Type: `System.Int32`
  - The timeout.

- **stateInfo**
  - Type: `TwinCAT.Ads.StateInfo` [1041].
  - The ADS statue and device status.

**Return Value**

- Type: `AdsErrorCode` [575]
  - AdsErrorCode of the ADS read state call. Check for AdsErrorCode.NoError to see if call was successful.

**Reference**

- `IAdsStateControlTimeout Interface` [921]
- TwinCAT.Ads Namespace [151]

6.2.43.1.3 IAdsStateControlTimeout.TryWriteControl Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![TryWriteControl(StateInfo, Int32)](TryWriteControl(StateInfo, Int32) [925])</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>![TryWriteControl(StateInfo, ReadOnlyMemory, Void)](TryWriteControl(StateInfo, ReadOnlyMemory, Void) [925])</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

- `IAdsStateControlTimeout Interface` [921]
- TwinCAT.Ads Namespace [151]
IAdsStateControlTimeout.TryWriteControl Method (StateInfo, Int32)

Changes the ADS status and the device status of an ADS server.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
AdsErrorCode TryWriteControl(
    StateInfo stateInfo,
    int timeout
)
```

**Parameters**

- **stateInfo**
  - Type: TwinCAT.Ads.StateInfo
  - New ADS status and device status.

- **timeout**
  - Type: System.Int32
  - The timeout.

**Return Value**

- Type: AdsErrorCode
- AdsErrorCode.

**Reference**

- IAdsStateControlTimeout Interface
- TryWriteControl Overload
- TwinCAT.Ads Namespace

---

IAdsStateControlTimeout.TryWriteControl Method (StateInfo, ReadOnlyMemory`1, Void)

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
AdsErrorCode TryWriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer,
    void timeout
)
```

**Parameters**

- **stateInfo**
  - Type: TwinCAT.Ads.StateInfo

- **writeBuffer**
  - Type: ReadOnlyMemory

---
timeout

Type: System.Void

Return Value

Type: AdsErrorCode [575]

Reference

IAdsStateControlTimeout Interface [921]

TryWriteControl Overload [924]

TwinCAT.Ads Namespace [151]

6.2.43.1.4 IAdsStateControlTimeout.WriteControl Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>➔ WriteControl(StateInfo, Int32) [926]</td>
<td>Changes the ADS status and the device status of an ADS server.</td>
</tr>
<tr>
<td>➔ WriteControl(StateInfo, ReadOnlyMemory, Void) [927]</td>
<td></td>
</tr>
</tbody>
</table>

Reference

IAdsStateControlTimeout Interface [921]

TwinCAT.Ads Namespace [151]

IAdsStateControlTimeout.WriteControl Method (StateInfo, Int32)

Changes the ADS status and the device status of an ADS server.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
void WriteControl(
    StateInfo stateInfo,
    int timeout)
```

Parameters

stateInfo

Type: TwinCAT.Ads.StateInfo [1041]

New ADS status and device status.
timeout

Type: System.Int32

The timeout.

Reference

IAdsStateControlTimeout Interface [► 921]
WriteControl Overload [► 926]
TwinCAT.Ads Namespace [► 151]

IAdsStateControlTimeout.WriteControl Method (StateInfo, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [► 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
void WriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer,
    void timeout
)
```

Parameters

stateInfo Type: TwinCAT.Ads.StateInfo [► 1041]
writeBuffer Type: ReadOnlyMemory
timeout Type: System.Void

Reference

IAdsStateControlTimeout Interface [► 921]
WriteControl Overload [► 926]
TwinCAT.Ads Namespace [► 151]

6.2.44 IAdsStateObserver Interface

Interface for an AdsState observer

Namespace: TwinCAT.Ads [► 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IAdsStateObserver
```
The IAdsStateObserver type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StateInfo</td>
<td>Gets the current state of the connected ADS Server.</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsStateChanged</td>
<td>Occurs when the ads state has been changed.</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads Namespace [151]

### 6.2.44.1 IAdsStateObserver Properties

The IAdsStateObserver [927] type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StateInfo</td>
<td>Gets the current state of the connected ADS Server.</td>
</tr>
</tbody>
</table>

### Reference

IAdsStateObserver Interface [927]

TwinCAT.Ads Namespace [151]

### 6.2.44.1.1 IAdsStateObserver.StateInfo Property

Gets the current state of the connected ADS Server.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
StateInfo StateInfo { get; }
```

**Property Value**

Type: StateInfo [1041]

ADS state

**Reference**

IAdsStateObserver Interface [927]

TwinCAT.Ads Namespace [151]
6.2.44.2 IAdsStateObserver Events

The I AdsStateObserver type exposes the following members.

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsStateChanged</td>
<td>Occurs when the ads state has been changed.</td>
</tr>
</tbody>
</table>

### Reference

IAdsStateObserver Interface

TwinCAT.Ads Namespace

6.2.44.2.1 IAdsStateObserver.AdsStateChanged Event

Occurs when the ads state has been changed.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

#### C#

```csharp
public interface IAdsStateProvider
```

### Value

**Type:** System.EventHandler<AdsStateChangedEventArgs2> AdsStateChanged

### Reference

IAdsStateObserver Interface

TwinCAT.Ads Namespace

6.2.45 IAdsStateProvider Interface

Interface IAdsStateProvider

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

#### C#

```csharp
public interface IAdsStateProvider
```

The IAdsStateProvider type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReadState</code> [931]</td>
<td>Reads the ADS status and the device status from an ADS server.</td>
</tr>
<tr>
<td><code>ReadStateAsync</code> [931]</td>
<td>Reads the ADS status and the device status from an ADS server.</td>
</tr>
<tr>
<td><code>RegisterAdsStateChangedAsync</code> [932]</td>
<td>Registers for <code>AdsStateChanged</code> [934] events as an asynchronous operation.</td>
</tr>
<tr>
<td><code>TryReadState</code> [932]</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the <code>ReadState</code> method this method does not call an exception on failure. Instead an <code>AdsErrorCode</code> is returned. If the return value is equal to <code>AdsErrorCode.NoError</code> the call was successful.</td>
</tr>
<tr>
<td><code>UnregisterAdsStateChangedAsync</code> [933]</td>
<td>Registers for <code>AdsStateChanged</code> [934] events as an asynchronous operation.</td>
</tr>
</tbody>
</table>

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>AdsStateChanged</code> [934]</td>
<td>Occurs when the AdsState of the target system has been changed.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]

6.2.45.1 IAdsStateProvider Methods

The `IAdsStateProvider` [929] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReadState</code> [931]</td>
<td>Reads the ADS status and the device status from an ADS server.</td>
</tr>
<tr>
<td><code>ReadStateAsync</code> [931]</td>
<td>Reads the ADS status and the device status from an ADS server.</td>
</tr>
<tr>
<td><code>RegisterAdsStateChangedAsync</code> [932]</td>
<td>Registers for <code>AdsStateChanged</code> [934] events as an asynchronous operation.</td>
</tr>
<tr>
<td><code>TryReadState</code> [932]</td>
<td>Reads the ADS status and the device status from an ADS server. Unlike the <code>ReadState</code> method this method does not call an exception on failure. Instead an <code>AdsErrorCode</code> is returned. If the return value is equal to <code>AdsErrorCode.NoError</code> the call was successful.</td>
</tr>
<tr>
<td><code>UnregisterAdsStateChangedAsync</code> [933]</td>
<td>Registers for <code>AdsStateChanged</code> [934] events as an asynchronous operation.</td>
</tr>
</tbody>
</table>

Reference

`IAdsStateProvider Interface` [929]
6.2.45.1.1 IAdsStateProvider.ReadState Method

 Reads the ADS status and the device status from an ADS server.

 **Namespace:** TwinCAT.Ads

 **Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

 **Syntax**

 **C#**

```csharp
StateInfo ReadState()
```

 **Return Value**

Type: [StateInfo](#)

The ADS statue and device status.

 **Reference**

IAdsStateProvider Interface

TwinCAT.Ads Namespace

6.2.45.1.2 IAdsStateProvider.ReadStateAsync Method

 Reads the ADS status and the device status from an ADS server.

 **Namespace:** TwinCAT.Ads

 **Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

 **Syntax**

 **C#**

```csharp
Task<ResultReadDeviceState> ReadStateAsync(
    CancellationToken cancel
)
```

 **Parameters**

`cancel`  
Type: System.Threading.CancellationToken  
The cancellation token

 **Return Value**

Type: [Task<ResultReadDeviceState>](#)

A task that represents the asynchronous ‘ReadState’ operation. The [ResultReadDeviceState](#) parameter contains the state ([State](#)) as long as the [ErrorCode](#) of the ADS communication after execution.

 **Reference**

IAdsStateProvider Interface
6.2.45.1.3 IAdsStateProvider/RegisterAdsStateChangedAsync Method

Registers for AdsStateChanged events as an asynchronous operation.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
Task<ResultAds> RegisterAdsStateChangedAsync(
    EventHandler<AdsStateChangedEventArgs> handler,
    CancellationToken cancel
)
```

**Parameters**

- **handler**
  - Type: `System.EventHandler<AdsStateChangedEventArgs>`
  - The handler function to be registered for AdsStateChanged calls.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.

**Return Value**

Type: `Task<ResultAds>`

A task that represents the asynchronous 'RegisterAdsStateChanged' operation. The `ResultAds` parameter contains the state the `ErrorCode` of the ADS communication after execution.

**Reference**

- IAdsStateProvider Interface
- TwinCAT.Ads Namespace

6.2.45.1.4 IAdsStateProvider.TryReadState Method

Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsError.NoError the call was successful.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
AdsErrorCode TryReadState(
    out StateInfo stateInfo
)
```
Parameters

stateInfo

Type: TwinCAT.Ads.StateInfo [1041].
The ADS statue and device status.

Return Value

Type: AdsErrorCode [575]
AdsErrorCode [575] of the ADS read state call. Check for NoError [575] to see if call was successful.

Reference

IAdsStateProvider Interface [929]
TwinCAT.Ads Namespace [151]

6.2.45.1.5 IAdsStateProvider.UnregisterAdsStateChangedAsync Method

Registers for AdsStateChanged [934] events as an asynchronous operation.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

Task<ResultAds> UnregisterAdsStateChangedAsync(
   EventHandler<AdsStateChangedEventArgs> handler,
   CancellationToken cancel
)

Parameters

handler

Type: System.EventHandler<AdsChangedEventArgs> [627].
The handler function to be unregistered.

cancel

Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task.ResultAds [989].
A task that represents the asynchronous 'UnregisterAdsStateChanged' operation. The ResultAds [989] parameter contains the state the ErrorCode [992] of the ADS communication after execution.

Reference

IAdsStateProvider Interface [929]
TwinCAT.Ads Namespace [151]

6.2.45.2 IAdsStateProvider Events

The IAdsStateProvider [929] type exposes the following members.
Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsStateChanged</td>
<td>Occurs when the AdsState of the target system has been changed.</td>
</tr>
</tbody>
</table>

Reference

IAdsStateProvider Interface [929]

TwinCAT.Ads Namespace [151]

6.2.45.2.1 IAdsStateProvider.AdsStateChanged Event

Occurs when the AdsState of the target system has been changed.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
    event EventHandler<AdsStateChangedEventArgs> AdsStateChanged
```

Value

Type: System.EventHandler<AdsStateChangedEventArgs>[627].

Remarks

This event occurs asynchronously if the synchronized flag is not set.

Reference

IAdsStateProvider Interface [929]

TwinCAT.Ads Namespace [151]

6.2.46 IAdsSymbolChangedProvider Interface

Interface IAdsConnectionLegacy

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
    public interface IAdsSymbolChangedProvider
```

The IAdsSymbolChangedProvider type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegisterSymbolVersionChangedAsync</td>
<td>Registers the symbol version changed asynchronously.</td>
</tr>
<tr>
<td>UnregisterSymbolVersionChangedAsync</td>
<td>Unregisters the symbol version changed asynchronously.</td>
</tr>
</tbody>
</table>

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsSymbolVersionChanged</td>
<td>Occurs when the ADS Symbol Version changed.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]

6.2.46.1 IAdsSymbolChangedProvider Methods

The IAdsSymbolChangedProvider [934] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegisterSymbolVersionChangedAsync</td>
<td>Registers the symbol version changed asynchronously.</td>
</tr>
<tr>
<td>UnregisterSymbolVersionChangedAsync</td>
<td>Unregisters the symbol version changed asynchronously.</td>
</tr>
</tbody>
</table>

Reference

IAdsSymbolChangedProvider Interface [934]

TwinCAT.Ads Namespace [151]

6.2.46.1.1 IAdsSymbolChangedProvider/RegisterSymbolVersionChangedAsync Method

Registers the symbol version changed asynchronously.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Task<ResultAds> RegisterSymbolVersionChangedAsync(
    EventHandler<AdsSymbolVersionChangedEventArgs> handler,
    CancellationToken cancel
)

Parameters

handler Type: System.EventHandler<AdsSymbolVersionChangedEventArgs>
The handler function to register.
cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultAds>
A task that represents the asynchronous 'RegisterSymbolVersionChanged' operation. The ResultAds parameter contains the value ErrorCode of the ADS communication after execution.

Reference

IAdsSymbolChangedProvider Interface
TwinCAT.Ads Namespace

6.2.46.1.2 IAdsSymbolChangedProvider.UnregisterSymbolVersionChangedAsync Method

Unregisters the symbol version changed asynchronous.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

Task<ResultAds> UnregisterSymbolVersionChangedAsync(
    EventHandler<AdsSymbolVersionChangedEventArgs> handler,
    CancellationToken cancel
)

Parameters

handler Type: System.EventHandler<AdsSymbolVersionChangedEventArgs>
The handler function to unregister.
cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultAds>
A task that represents the asynchronous 'UnregisterSymbolVersionChangedAsync' operation. The ResultAds parameter contains the value ErrorCode of the ADS communication after execution.
### IAdsSymbolChangedProvider Events

The `IAdsSymbolChangedProvider` type exposes the following members.

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="937" alt=" AdsSymbolVersionChanged " /></td>
<td>Occurs when the ADS Symbol Version changed.</td>
</tr>
</tbody>
</table>

#### IAdsSymbolChangedProvider.AdsSymbolVersionChanged Event

Occurs when the ADS Symbol Version changed.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```
C#

event EventHandler<AdsSymbolVersionChangedEventArgs> AdsSymbolVersionChanged
```

**Value**

Type: `System.EventHandler<AdsSymbolVersionChangedEventArgs>`

**Reference**

- `IAdsSymbolChangedProvider Interface` (934)
- TwinCAT.Ads Namespace (151)

### IAdsSymbolicAccess Interface

Interface for symbolic ads access.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public interface IAdsSymbolicAccess
```

The `IAdsSymbolicAccess` type exposes the following members.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CleanupSymbolTable</td>
<td>Clears the internal symbol / DataTypes cache.</td>
</tr>
<tr>
<td>ReadDataType</td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>ReadDataTypeAsync</td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>ReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>ReadSymbolAsync</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>ReadValue(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>ReadValue(String, Type)</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>ReadValue.T.(String)</td>
<td>Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>ReadValue.T.(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>ReadValueAsync(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously and returns it as an object.</td>
</tr>
<tr>
<td>ReadValueAsync(String, Type, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>ReadValueAsync.T.(String, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>ReadValueAsync.T.(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously and returns it as an object.</td>
</tr>
<tr>
<td>TryReadDataType</td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>TryReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryReadValue(ISymbol, Object) [954]</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>TryReadValue(String, Type, Object) [956]</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
<tr>
<td>TryReadValue.T.(String, T) [954]</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
<tr>
<td>TryReadValue.T.(ISymbol, T) [955]</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>TryWriteValue(String, Object) [957]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter</td>
</tr>
<tr>
<td></td>
<td>type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryWriteValue(ISymbol, Object) [959]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32,</td>
</tr>
<tr>
<td></td>
<td>Int32, Bool etc.) are supported. If a string is passed as parameter, the</td>
</tr>
<tr>
<td></td>
<td>method attempts to parse the string according to the ADS data type of the</td>
</tr>
<tr>
<td></td>
<td>symbol.</td>
</tr>
<tr>
<td>TryWriteValue.T.(String, T) [958]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter</td>
</tr>
<tr>
<td></td>
<td>type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryWriteValue.T.(ISymbol, T) [960]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32,</td>
</tr>
<tr>
<td></td>
<td>Int32, Bool etc.) are supported. If a string is passed as parameter, the</td>
</tr>
<tr>
<td></td>
<td>method attempts to parse the string according to the ADS data type of the</td>
</tr>
<tr>
<td></td>
<td>symbol.</td>
</tr>
<tr>
<td>WriteValue(String, Object) [961]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter</td>
</tr>
<tr>
<td></td>
<td>type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue(ISymbol, Object) [962]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32,</td>
</tr>
<tr>
<td></td>
<td>Int32, Bool etc.) are supported. If a string is passed as parameter, the</td>
</tr>
<tr>
<td></td>
<td>method attempts to parse the string according to the ADS data type of the</td>
</tr>
<tr>
<td></td>
<td>symbol.</td>
</tr>
<tr>
<td>WriteValue.T.(String, T) [961]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter</td>
</tr>
<tr>
<td></td>
<td>type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue.T.(ISymbol, T) [963]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32,</td>
</tr>
<tr>
<td></td>
<td>Int32, Bool etc.) are supported. If a string is passed as parameter, the</td>
</tr>
<tr>
<td></td>
<td>method attempts to parse the string according to the ADS data type of the</td>
</tr>
<tr>
<td></td>
<td>symbol.</td>
</tr>
<tr>
<td>WriteValueAsync(ISymbol, Object, CancellationToken) [965]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32,</td>
</tr>
<tr>
<td></td>
<td>Int32, Bool etc.) are supported. If a string is passed as parameter, the</td>
</tr>
<tr>
<td></td>
<td>method attempts to parse the string according to the ADS data type of the</td>
</tr>
<tr>
<td></td>
<td>symbol.</td>
</tr>
<tr>
<td>WriteValueAsync.T.(String, T, CancellationToken) [964]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter</td>
</tr>
<tr>
<td></td>
<td>type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValueAsync.T.(ISymbol, T, CancellationToken) [966]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32,</td>
</tr>
<tr>
<td></td>
<td>Int32, Bool etc.) are supported. If a string is passed as parameter, the</td>
</tr>
<tr>
<td></td>
<td>method attempts to parse the string according to the ADS data type of the</td>
</tr>
<tr>
<td></td>
<td>symbol.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]
## 6.2.47.1 IAdsSymbolicAccess Methods

The **IAdsSymbolicAccess** [� 937] type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CleanupSymbolTable</td>
<td>Clears the internal symbol / DataTypes cache.</td>
</tr>
<tr>
<td>ReadDataType</td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>ReadDataTypeAsync</td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>ReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>ReadSymbolAsync</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>ReadValue(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>ReadValue(String, Type)</td>
<td>Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>ReadValue.T.(String)</td>
<td>Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>ReadValue.T.(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>ReadValueAsync(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously and returns it as an object.</td>
</tr>
<tr>
<td>ReadValueAsync(String, Type, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>ReadValueAsync.T.(String, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td>ReadValueAsync.T.(ISymbol, CancellationToken)</td>
<td>Reads the value of a symbol asynchronously and returns it as an object.</td>
</tr>
<tr>
<td>TryReadDataType</td>
<td>Call this method to obtain information about the specified data type.</td>
</tr>
<tr>
<td>TryReadSymbol</td>
<td>Call this method to obtain information about the individual symbols (variables) in ADS devices.</td>
</tr>
<tr>
<td>TryReadValue(ISymbol, Object)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryReadValue(String, Type, Object)</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
<tr>
<td>TryReadValue.T.(String, T)</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
<tr>
<td>TryReadValue.T.(ISymbol, T)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>TryWriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryWriteValue(ISymbol, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>TryWriteValue.T.(String, T)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>TryWriteValue.T.(ISymbol, T)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue(ISymbol, Object)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValue.(ISymbol, T)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValueAsync(ISymbol, Object, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValueAsync.T.(String, T, CancellationToken)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValueAsync.T.(ISymbol, T, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
</tbody>
</table>

**Reference**

IAdsSymbolicAccess Interface [937]

TwinCAT.Ads Namespace [151]
### 6.2.47.1.1  IAdsSymbolicAccess.CleanupSymbolTable Method

Clears the internal symbol / DataTypes cache.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
void CleanupSymbolTable()
```

**Remarks**

Previously stored symbol information is cleared. As a consequence the symbol information must be obtained from the ADS server again if accessed, which which needs an extra ADS round trip.

**Reference**

IAdsSymbolicAccess Interface [937]

TwinCAT.Ads Namespace [151]

### 6.2.47.1.2  IAdsSymbolicAccess.ReadDataType Method

Call this method to obtain information about the specified data type.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
IDataType ReadDataType(
    string typeName
)
```

**Parameters**

**typeName**  
Type: System.String  
Name of the data type (without namespace)

**Return Value**

Type: IDataType [1986]  
An containing the requested type.

**Reference**

IAdsSymbolicAccess Interface [937]

TwinCAT.Ads Namespace [151]

IAdsSymbolicAccess.TryReadDataType(String, IDataType.) [952]

IAdsSymbolicAccess.ReadDataTypeAsync(String, CancellationToken) [943]
6.2.47.1.3 IAdsSymbolicAccess.ReadDataTypeAsync Method

Call this method to obtain information about the specified data type.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
Task<ResultValue<IDataType>> ReadDataTypeAsync(
    string typeName,
    CancellationToken cancel
)
```

**Parameters**

- **typeName**
  - Type: `System.String`
  - Name of the data type.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancel token.

**Return Value**

Type: `TaskForResultValue<IDataType>`

A task that represents the asynchronous 'ReadDataType' operation. The `ResultValue.TValue` parameter contains the read value (`Value`) and the `ErrorCode` after execution.

**Reference**

- `IAdsSymbolicAccess Interface` [937]
- TwinCAT.Ads Namespace [151]
- `IAdsSymbolicAccess.ReadDataType(String)` [942]
- `IAdsSymbolicAccess.TryReadDataType(String, IDataType)` [952]

6.2.47.1.4 IAdsSymbolicAccess.ReadSymbol Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
IAdsSymbol ReadSymbol(
    string symbolPath
)
```

**Parameters**

- **symbolPath**
  - Type: `System.String`
  - Name of the symbol.
Return Value

Type: IAdsSymbol

An IAdsSymbol containing the requested symbol information or null if symbol could not be found.

Reference

IAdsSymbolicAccess Interface

TwinCAT.Ads Namespace

IAdsSymbolicAccess.TryReadSymbol(String, IAdsSymbol)

IAdsSymbolicAccess.ReadSymbolAsync(String, CancellationToken)

6.2.47.1.5 IAdsSymbolicAccess.ReadSymbolAsync Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

Task<ResultValue<IAdsSymbol>> ReadSymbolAsync(
    string symbolPath,
    CancellationToken cancel
)

Parameters

symbolPath Type: System.String
Name of the symbol.

cancel Type: System.Threading.CancellationToken
The cancel token.

Return Value

Type: Task<ResultValue[IAdsSymbol]].

A task that represents the asynchronous ‘ReadSymbolInfo’ operation. The ResultValue.TValue parameter contains the read value (Value) and the ErrorCode after execution.

Reference

IAdsSymbolicAccess Interface

TwinCAT.Ads Namespace

IAdsSymbolicAccess.ReadSymbol(String)

IAdsSymbolicAccess.TryReadSymbol(String, IAdsSymbol)
6.2.47.1.6  IAdsSymbolicAccess.ReadValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadValue&lt;T&gt;(String)</td>
<td>Reads the value of a symbol and returns the value. The parameter type</td>
</tr>
<tr>
<td></td>
<td>must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>ReadValue(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>ReadValue&lt;T&gt;(ISymbol)</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>ReadValue(String, Type)</td>
<td>Reads the value of a symbol and returns the value as object. The parameter</td>
</tr>
<tr>
<td></td>
<td>type must have the same binary layout as the ADS symbol.</td>
</tr>
</tbody>
</table>

Reference

IAdsSymbolicAccess Interface [937]

TwinCAT.Ads Namespace [151]

IAdsSymbolicAccess.ReadValue.T. Method (String)

Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

T ReadValue<T>(
    string symbolPath
)

Parameters

symbolPath  Type: System.String
            Name of the ADS symbol.

Type Parameters

T  The value type

Return Value

Type: T
Value of the symbol

Reference

IAdsSymbolicAccess Interface [937]
ReadValue Overload [945]

TwinCAT.Ads Namespace [151]
IAdsSymbolicAccess.TryReadValue.T.(String, T.) [954]
IAdsSymbolicAccess.ReadValueAsync.T.(String, CancellationToken) [948]

IAdsSymbolicAccess.ReadValue Method (ISymbol)
Reads the value of a symbol and returns it as an object.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
Object ReadValue(
    ISymbol symbol
)
```

**Parameters**

- `symbol`  
  Type: TwinCAT.TypeSystem.ISymbol [2176]  
  The symbol that should be read.

**Return Value**

- Type: `Object`  
  The value of the symbol as an object.

**Remarks**

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

**Reference**

IAdsSymbolicAccess Interface [937]  
ReadValue Overload [945]  
TwinCAT.Ads Namespace [151]  
IAdsSymbolicAccess.TryReadValue(ISymbol, Object.) [954]  
IAdsSymbolicAccess.ReadValueAsync(ISymbol, CancellationToken) [949]

IAdsSymbolicAccess.ReadValue.T. Method (ISymbol)
Reads the value of a symbol and returns it as an object.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
T ReadValue<T>(
    ISymbol symbol
)
```

Parameters

- **symbol**: Type: TwinCAT.TypeSystem.ISymbol
  The symbol that should be read.

Type Parameters

- **T**: The value type.

Return Value

- Type: T
  The value of the symbol.

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

- IAdsSymbolicAccess Interface
- ReadValue Overload
- TwinCAT.Ads Namespace
- IAdsSymbolicAccess.TryReadValue<T>(ISymbol, T)
- IAdsSymbolicAccess.ReadValueAsync<T>(ISymbol, CancellationToken)

IAdsSymbolicAccess.ReadValue Method (String, Type)

Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Object ReadValue(
    string symbolPath,
    Type type
)
```

Parameters

- **symbolPath**: Type: System.String
  Symbol Path of the ADS symbol.
TwinCAT.Ads Namespaces

Return Value

Type: `Object`
Value of the symbol

Reference

- `IAdsSymbolicAccess Interface` [937]
- `ReadValue Overload` [945]
- `TwinCAT.Ads Namespace` [151]
- `IAdsSymbolicAccess.TryReadValue(String, Type, Object)` [956]
- `IAdsSymbolicAccess.ReadValueAsync(String, Type, CancellationToken)` [951]

### 6.2.47.1.7 `IAdsSymbolicAccess.ReadValueAsync` Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReadValueAsync&lt;T&gt;(String, CancellationToken)</code> [948]</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
<tr>
<td><code>ReadValueAsync(ISymbol, CancellationToken)</code> [949]</td>
<td>Reads the value of a symbol asynchronously and returns it as an object.</td>
</tr>
<tr>
<td><code>ReadValueAsync&lt;T&gt;(ISymbol, CancellationToken)</code> [950]</td>
<td>Reads the value of a symbol asynchronously and returns it as an object.</td>
</tr>
<tr>
<td><code>ReadValueAsync(String, Type, CancellationToken)</code> [951]</td>
<td>Reads the value of a symbol asynchronously.</td>
</tr>
</tbody>
</table>

Reference

- `IAdsSymbolicAccess Interface` [937]
- `TwinCAT.Ads Namespace` [151]

### IAdsSymbolicAccess.ReadValueAsync<T> Method (String, CancellationToken)

Reads the value of a symbol asynchronously.
### Syntax

C#

```csharp
Task<ResultValue<T>> ReadValueAsync<T>(
    string symbolPath,
    CancellationToken cancel
)
```

### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbolPath</td>
<td><code>System.String</code></td>
<td>Name of the ADS symbol.</td>
</tr>
<tr>
<td>cancel</td>
<td><code>System.Threading.CancellationToken</code></td>
<td>The cancel token.</td>
</tr>
</tbody>
</table>

### Type Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td><code>T</code></td>
<td>The value type.</td>
</tr>
</tbody>
</table>

### Return Value

Type: `Task<ResultValue<T>>`  
A task that represents the asynchronous read operation. The `ResultAnyValue` parameter contains the read value (`Value`) and the `ErrorCode` after execution.

### Remarks

The parameter type must have the same binary layout as the ADS symbol.

### Reference

- `IAdsSymbolicAccess Interface`
- `ReadValueAsync Overload`
- `TwinCAT.Ads Namespace`
- `IAdsSymbolicAccess.ReadValue.T.(String)`  
- `IAdsSymbolicAccess.TryReadValue.T.(String, T.)`

### `IAdsSymbolicAccess.ReadValueAsync Method` (ISymbol, CancellationToken)

Reads the value of a symbol asynchronously and returns it as an object.

**Namespace:** TwinCAT.Ads  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

Task<ResultAnyValue> ReadValueAsync(
    ISymbol symbol,
    CancellationToken cancel
)

Parameters

symbol  
Type: TwinCAT.TypeSystem.ISymbol
The symbol that should be read.

cancel  
Type: System.Threading.CancellationToken
The cancel token.

Return Value

Type: Task<ResultAnyValue>
A task that represents the asynchronous read operation. The ResultAnyValue parameter contains the read value (Value) and the ErrorCode after execution.

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

IAdsSymbolicAccess Interface
ReadValueAsync Overload
TwinCAT.Ads Namespace
IAdsSymbolicAccess.ReadValue(ISymbol)
IAdsSymbolicAccess.ReadValue(ISymbol)
IAdsSymbolicAccess.TryReadValue(ISymbol, Object.)

IAdsSymbolicAccess.ReadValueAsync.T. Method (ISymbol, CancellationToken)

Reads the value of a symbol asynchronously and returns it as an object.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

Task<ResultValue<T>> ReadValueAsync<T>(
    ISymbol symbol,
    CancellationToken cancel
)
Parameters

symbol

Type: TwinCAT.TypeSystem.ISymbol
The symbol that should be read.

cancel

Type: System.Threading.CancellationToken
The cancel token.

Type Parameters

T
The value type.

Return Value

Type: Task<ResultAnyValue>
A task that represents the asynchronous read operation. The ResultAnyValue parameter contains the read value (Value) and the ErrorCode after execution.

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

IAdsSymbolicAccess Interface

ReadValueAsync Overload

TwinCAT.Ads Namespace

IAdsSymbolicAccess.ReadValue.T.(ISymbol)
IAdsSymbolicAccess.TryReadValue.T.(ISymbol, T.)

IAdsSymbolicAccess.ReadValueAsync Method (String, Type, CancellationToken)

Reads the value of a symbol asynchronously.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

Task<ResultAnyValue> ReadValueAsync(
    string symbolPath,
    Type type,
    CancellationToken cancel
)

Parameters

symbolPath

Type: System.String
Name of the ADS symbol.

type

Type: System.Type
Managed type of the ADS symbol

cancel

Type: System.Threading.CancellationToken
The cancel token.
Return Value

Type: Task.ResultAnyValue [998].
A task that represents the asynchronous read operation. The ResultAnyValue [998] parameter contains the read value (Value [1000]) and the ErrorCode [992] after execution.

Remarks

The parameter type must have the same binary layout as the ADS symbol.

Reference

IAdsSymbolicAccess Interface [937]
ReadValueAsync Overload [948]
TwinCAT.Ads Namespace [151]
IAdsSymbolicAccess.ReadValue(String, Type) [947]
IAdsSymbolicAccess.TryReadValue(String, Type, Object) [956]

6.2.47.1.8 IAdsSymbolicAccess.TryReadDataType Method

Call this method to obtain information about the specified data type.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

AdsErrorCode TryReadDataType(
    string typeName,
    out IDataType dataType
)

Parameters

typeName Type: System.String
Name of the symbol.

dataType Type: TwinCAT.TypeSystem.IDataType [1986].
The symbol.

Return Value

Type: AdsErrorCode [575]
A IDataType [1986] containing the requested symbol information or null if symbol could not be found.

Reference

IAdsSymbolicAccess Interface [937]
TwinCAT.Ads Namespace [151]
IAdsSymbolicAccess.ReadDataType(String) [942]
6.2.47.1.9 IAdsSymbolicAccess.TryReadSymbol Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
AdsErrorCode TryReadSymbol(
    string symbolPath,
    out IAdsSymbol symbol
)
```

**Parameters**

- **symbolPath**
  - Type: System.String
  - Name of the symbol.

- **symbol**
  - Type: TwinCAT.Ads.TypeSystem.IAdsSymbol [1379]
  - The symbol.

**Return Value**

- Type: AdsErrorCode [575]
- An IAdsSymbol [1379] containing the requested symbol information or null if symbol could not be found.

**Reference**

- IAdsSymbolicAccess Interface [937]
- TwinCAT.Ads Namespace [151]
- IAdsSymbolicAccess.ReadSymbol(String) [943]

6.2.47.1.10 IAdsSymbolicAccess.TryReadValue Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryReadValue.T.&lt;br&gt;(String, T.) [954]</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
<tr>
<td>TryReadValue(ISymbol, Object.) [954]</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>TryReadValue.T.&lt;br&gt;(ISymbol, T.) [955]</td>
<td>Reads the value of a symbol and returns it as an object.</td>
</tr>
<tr>
<td>TryReadValue(String&lt;br&gt;, Type, Object.) [956]</td>
<td>Reads the value of a symbol and returns the value as object.</td>
</tr>
</tbody>
</table>

**Reference**

- IAdsSymbolicAccess Interface [937]
IAdsSymbolicAccess.TryReadValue<T> Method (String, T.)

Reads the value of a symbol and returns the value as object.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
AdsErrorCode TryReadValue<T>(
    string symbolPath,
    out T value
)
```

**Parameters**

- **symbolPath**
  - Type: `System.String`
  - Name of the ADS symbol.

- **value**
  - Type: `T`
  - The read value of the Symbol.

**Type Parameters**

- **T**
  - The value type.

**Return Value**

Type: `AdsErrorCode`

The `AdsErrorCode`.

**Remarks**

The parameter type must have the same binary layout as the ADS symbol.

**Reference**

- `IAdsSymbolicAccess Interface`  
- `TryReadValue Overload`  
- `TwinCAT.Ads Namespace`  
- `IAdsSymbolicAccess.ReadValue<T>(String)`  
- `IAdsSymbolicAccess.ReadValueAsync<T>(String, CancellationToken)`

**IAdsSymbolicAccess.TryReadValue Method (ISymbol, Object.)**

Reads the value of a symbol and returns it as an object.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#  

```csharp
AdsErrorCode TryReadValue<T>(
    ISymbol symbol,
    out T value
)
```

Parameters

symbol  
Type: TwinCAT.TypeSystem.ISymbol [2176]
The symbol that should be read.

value  
Type: T.
The value.

Return Value

Type: AdsErrorCode [575]
The ADS Error Code

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

IAdsSymbolicAccess Interface [937]
TryReadValue Overload [953]
TwinCAT.Ads Namespace [151]
IAdsSymbolicAccess.ReadValue(ISymbol) [946]
IAdsSymbolicAccess.ReadValueAsync(ISymbol, CancellationToken) [949]

IAdsSymbolicAccess.TryReadValue<T>. Method (ISymbol, T.)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  

```csharp
AdsErrorCode TryReadValue<T>(
    ISymbol symbol,
    out T value
)
```

Parameters

symbol  
Type: TwinCAT.TypeSystem.ISymbol [2176]
The symbol that should be read.

value  
Type: T.
The value.
Type Parameters

T  

The value type.

Return Value

Type: AdsErrorCode [575]

The ADS Error Code

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

IAdsSymbolicAccess Interface [937]

TryReadValue Overload [953]

TwinCAT.Ads Namespace [151]

IAdsSymbolicAccess.ReadValue.T.(ISymbol) [946]

IAdsSymbolicAccess.ReadValueAsync(ISymbol, CancellationToken) [949]

IAdsSymbolicAccess.TryReadValue Method (String, Type, Object.)

Reads the value of a symbol and returns the value as object.

Namespace:  TwinCAT.Ads [151]

Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#  

AdsErrorCode TryReadValue(
    string symbolPath,
    Type type,
    out Object value
)

Parameters

symbolPath  

Type: System.String

Name of the ADS symbol.

type  

Type: System.Type

Managed type of the ADS symbol.

value  

Type: System.Object

The read value of the Symbol.

Return Value

Type: AdsErrorCode [575]

The AdsErrorCode [575].
Remarks
The parameter type must have the same binary layout as the ADS symbol.

Reference

IAdsSymbolicAccess Interface [937]

TryReadValue Overload [953]

TwinCAT.Ads Namespace [151]

IAdsSymbolicAccess.ReadValue(String, Type) [947]

IAdsSymbolicAccess.ReadValueAsync<T>(String, CancellationToken) [948]

IAdsSymbolicAccess.ReadValueAsync(String, Type, CancellationToken) [951]

6.2.47.1.11 IAdsSymbolicAccess.TryWriteValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![TryWriteValue(String, Object)] [957]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.</td>
</tr>
<tr>
<td>![TryWriteValue(T, String, T)] [958]</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>![TryWriteValue(ISymbol, Object)] [959]</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>![TryWriteValue(T, ISymbol, T)] [960]</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
</tbody>
</table>

Reference

IAdsSymbolicAccess Interface [937]

TwinCAT.Ads Namespace [151]

IAdsSymbolicAccess.TryWriteValue Method (String, Object)

Writes the passed object value to the specified ADS symbol. The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
AdsErrorCode TryWriteValue(
    string symbolPath,
    Object value
)
```

**Parameters**

symbolPath  
**Type:** System.String  
Name of the ADS symbol.

value  
**Type:** System.Object  
Object holding the value to be written to the ADS symbol

**Return Value**

**Type:** AdsErrorCode

AdsErrorCode.

**Reference**

- IAdsSymbolicAccess Interface [937]
- TryWriteValue Overload [957]
- TwinCAT.Ads Namespace [151]
- IAdsSymbolicAccess.WriteValue(String, Object) [961]
- IAdsSymbolicAccess.WriteValueAsync.T. (String, T, CancellationToken) [964]

**IAdsSymbolicAccess.TryWriteValue.T. Method (String, T)**

Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
AdsErrorCode TryWriteValue<T>(
    string symbolPath,
    T value
)
```

**Parameters**

symbolPath  
**Type:** System.String  
Name of the ADS symbol.

value  
**Type:** T  
Object holding the value to be written to the ADS symbol

**Type Parameters**

T  
The value type.
Return Value

Type: AdsErrorCode [575]

AdsErrorCode.

Reference

IAdsSymbolicAccess Interface [937]

TryWriteValue Overload [957]

TwinCAT.Ads Namespace [151]

IAdsSymbolicAccess.WriteValue.T.(String, T) [961]

IAdsSymbolicAccess.TryWriteValue.T.(String, T)

IAdsSymbolicAccess.TryWriteValue Method (ISymbol, Object)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

AdsErrorCode TryWriteValue(
    ISymbol symbol,
    Object val
)

Parameters

symbol Type: TwinCAT.TypeSystem.ISymbol [2176]
The symbol the value is written to.

val Type: System.Object
The value to write.

Return Value

Type: AdsErrorCode [575]

AdsErrorCode.

Reference

IAdsSymbolicAccess Interface [937]

TryWriteValue Overload [957]

TwinCAT.Ads Namespace [151]

IAdsSymbolicAccess.WriteValue(ISymbol, Object) [962]

IAdsSymbolicAccess.WriteValueAsync(ISymbol, Object, CancellationToken) [965]
IAdsSymbolicAccess.TryWriteValue<T> Method (ISymbol, T)

Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdc3a3e72bc0ea15da1c14

Syntax

C#

AdsErrorCode TryWriteValue<T>(
   ISymbol symbol,
   T val
)

Parameters

symbol Type: TwinCAT.TypeSystem.ISymbol
The symbol the value is written to.

val Type: T
The value to write.

Type Parameters

T The value type.

Return Value

Type: AdsErrorCode
AdsErrorCode.

Reference

IAdsSymbolicAccess Interface
TryWriteValue Overload
TwinCAT.Ads Namespace
IAdsSymbolicAccess.WriteValue<T>.ISymbol, T
IAdsSymbolicAccess.WriteValueAsync<T>.ISymbol, T, CancellationToken

6.2.47.1.12 IAdsSymbolicAccess.WriteValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValue(String, Object)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValue&lt;T&gt;(String, T)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WriteValue(ISymbol, Object) [962]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValue.T(ISymbol, T) [963]</td>
<td>Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
</tbody>
</table>

Reference

IAdsSymbolicAccess Interface [937]

TwinCAT.Ads Namespace [151]

**IAdsSymbolicAccess.WriteValue Method (String, Object)**

Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
void WriteValue(  
    string symbolPath,  
    Object value  
)

**Parameters**

- **symbolPath**  
  Type: System.String  
  Name of the ADS symbol.

- **value**  
  Type: System.Object  
  Object holding the value to be written to the ADS symbol.

Reference

IAdsSymbolicAccess Interface [937]  
WriteValue Overload [960]  
TwinCAT.Ads Namespace [151]  
IAdsSymbolicAccess.TryWriteValue(String, Object) [957]  
IAdsSymbolicAccess.WriteValueAsync.T(String, T, CancellationToken) [964]

**IAdsSymbolicAccess.WriteValue.T Method (String, T)**

Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol.
**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
void WriteValue<T>(
    string symbolPath,
    T value
)
```

**Parameters**

- **symbolPath**
  - Type: System.String
  - Name of the ADS symbol.

- **value**
  - Type: T
  - Object holding the value to be written to the ADS symbol

**Type Parameters**

- **T**
  - the value type.

**Reference**

- IAdsSymbolicAccess Interface [937]
- WriteValue Overload [960]
- TwinCAT.Ads Namespace [151]
- IAdsSymbolicAccess.TryWriteValue.T.(String, T) [958]
- IAdsSymbolicAccess.WriteValueAsync.T.(String, T, CancellationToken) [964]

**IAdsSymbolicAccess.WriteValue Method (ISymbol, Object)**

Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
void WriteValue(
    ISymbol symbol,
    Object val
)
```

**Parameters**

- **symbol**
  - Type: TwinCAT.TypeSystem.ISymbol [2176]
  - The symbol the value is written to.

- **val**
  - Type: System.Object
  - The value to write.
IAdsSymbolicAccess.WriteValue.T. Method (ISymbol, T)

Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
void WriteValue<T>(
    ISymbol symbol,
    T val
)
```

Parameters

- **symbol**
  Type: TwinCAT.TypeSystem.ISymbol
  The symbol the value is written to.

- **val**
  Type: T
  The value to write.

Type Parameters

- **T**
  The value type.

Reference

- IAdsSymbolicAccess Interface [937]
- WriteValue Overload [960]
- TwinCAT.Ads Namespace [151]
- IAdsSymbolicAccess.TryWriteValue.ISymbol, Object [959]
- IAdsSymbolicAccess.WriteValueAsync.ISymbol, Object, CancellationToken [965]
6.2.47.1.13  IAdsSymbolicAccess.WriteValueAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValueAsync&lt;T&gt;.WriteValueAsync&lt;T&gt;(String, T, CancellationToken)</td>
<td>Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol.</td>
</tr>
<tr>
<td>WriteValueAsync(ISymbol, Object, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
<tr>
<td>WriteValueAsync&lt;T&gt;.WriteValueAsync&lt;T&gt;(ISymbol, T, CancellationToken)</td>
<td>Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.</td>
</tr>
</tbody>
</table>

Reference

IAdsSymbolicAccess Interface [937]

TwinCAT.Ads Namespace [151]

IAdsSymbolicAccess.WriteValueAsync<T>. Method (String, T, CancellationToken)

Writes the passed object value to the specified ADS symbol. The parameter type must have the same binary layout as the ADS symbol.

Namespace:  TwinCAT.Ads Namespace [151]

Syntax

C#

```csharp
Task<ResultWrite> WriteValueAsync<T>(
    string symbolPath,
    T value,
    CancellationToken cancel
)
```

Parameters

- **symbolPath**
  - Type: `System.String`
  - Name of the ADS symbol.
- **value**
  - Type: `T`
  - Object holding the value to be written to the ADS symbol
- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancel token.

Type Parameters

- **T**
  - The value type.
Return Value

Type: Task.ResultWrite [1032].
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [1032] parameter contains the ErrorCode [992] after execution.

Reference

IAdsSymbolicAccess Interface [937]
WriteValueAsync Overload [964]
TwinCAT.Ads Namespace [151]
IAdsSymbolicAccess.WriteValue.T.(String, T) [961]
IAdsSymbolicAccess.TryWriteValue.T.(ISymbol, T) [960]

IAdsSymbolicAccess.WriteValueAsync Method (ISymbol, Object, CancellationToken)

Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Task<ResultWrite> WriteValueAsync(
    ISymbol symbol,
    Object val,
    CancellationToken cancel
)
```

Parameters

symbol Type: TwinCAT.TypeSystem.ISymbol [2176]
The symbol the value is written to.

val Type: System.Object
The value to write.

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task.ResultWrite [1032].
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [1032] parameter contains the ErrorCode [992] after execution.

Reference

IAdsSymbolicAccess Interface [937]
WriteValueAsync Overload [964]
IAdsSymbolicAccess.WriteValueAsync<T>. Method (ISymbol, T, CancellationToken)

Writes a value to the symbol. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
Task<ResultWrite> WriteValueAsync<T>(
    ISymbol symbol,
    T val,
    CancellationToken cancel
)
```

**Parameters**

- `symbol`  
  Type: TwinCAT.TypeSystem.ISymbol [2176]
  The symbol the value is written to.

- `val`  
  Type: T
  The value to write.

- `cancel`  
  Type: System.Threading.CancellationToken
  The cancellation token.

**Type Parameters**

- `T`  
  The value type.

**Return Value**

Type: Task<ResultWrite> [1032].

A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [1032] parameter contains the ErrorCode [992] after execution.

**Reference**

- IAdsSymbolicAccess Interface [937]
- WriteValueAsync Overload [964]
- TwinCAT.Ads Namespace [151]
- IAdsSymbolicAccess.WriteByte<T, ISymbol, T> [963]
- IAdsSymbolicAccess.TryWriteValue<T, ISymbol, T> [960]
6.2.48 IAdsSymbolTableProvider Interface

Interface IAdsSymbolTableProvider

Namespace:  TwinCAT.Ads [151]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public interface IAdsSymbolTableProvider

The IAdsSymbolTableProvider type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSymbolTableAsync</td>
<td>Gets the symbol table asynchronously</td>
</tr>
<tr>
<td>SetSymbolEncoding</td>
<td>Sets the default encoding.</td>
</tr>
<tr>
<td>TryGetSymbolTable</td>
<td>Get the symbol table.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]

6.2.48.1 IAdsSymbolTableProvider Methods

The IAdsSymbolTableProvider [967] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSymbolTableAsync</td>
<td>Gets the symbol table asynchronously</td>
</tr>
<tr>
<td>SetSymbolEncoding</td>
<td>Sets the default encoding.</td>
</tr>
<tr>
<td>TryGetSymbolTable</td>
<td>Get the symbol table.</td>
</tr>
</tbody>
</table>

Reference

IAdsSymbolTableProvider Interface [967]
TwinCAT.Ads Namespace [151]

6.2.48.1.1 IAdsSymbolTableProvider.GetSymbolTableAsync Method

Gets the symbol table asynchronously
Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Task<ResultValue<ISymbolInfoTable>> GetSymbolTableAsync(
    CancellationToken cancel
)
```

Parameters

cancel Type: System.Threading.CancellationToken

The cancellation token.

Return Value

Type: Task<ResultValue<ISymbolInfoTable>>

A task that represents the asynchronous 'RegisterSymbolVersionChanged' operation. The ResultValue.TValue parameter contains the value Value and the return code ErrorCode of the ADS communication after execution.

Reference

IAdsSymbolTableProvider Interface

TwinCAT.Ads Namespace

6.2.48.1.2 IAdsSymbolTableProvider.SetSymbolEncoding Method

Sets the default encoding.

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
void SetSymbolEncoding(
    Encoding encoding
)
```

Parameters

coding Type: System.Text.Encoding

The encoding.

Reference

IAdsSymbolTableProvider Interface

TwinCAT.Ads Namespace

6.2.48.1.3 IAdsSymbolTableProvider.TryGetSymbolTable Method

Get the symbol table.
Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
AdsErrorCode TryGetSymbolTable(out ISymbolInfoTable table)
```

Parameters

table Type: ISymbolInfoTable.
The symbol table.

Return Value

Type: AdsErrorCode
The ADS Error Code.

Reference

IAdsSymbolTableProvider Interface
TwinCAT.Ads Namespace

6.2.49 INotification Interface

Common INotification interface

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface INotification
```

The INotification type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data [970]</td>
<td>The notification Data.</td>
</tr>
<tr>
<td>Handle [970]</td>
<td>The notification handle</td>
</tr>
<tr>
<td>TimeStamp [971]</td>
<td>Gets the time stamp of the INotification</td>
</tr>
<tr>
<td>UserData [971]</td>
<td>Attached UserData/Tag at the INotification</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace
6.2.49.1  INotification Properties

The INotification [969] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>The notification Data.</td>
</tr>
<tr>
<td>Handle</td>
<td>The notification handle</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Gets the time stamp of the INotification [969]</td>
</tr>
<tr>
<td>UserData</td>
<td>Attached UserData/Tag at the INotification [969]</td>
</tr>
</tbody>
</table>

Reference

INotification Interface [969]
TwinCAT.Ads Namespace [151]

6.2.49.1.1  INotification.Data Property

The notification Data.

Namespace:  TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdca3e72bc0ea15da1c14

Syntax

C#
ReadOnlyMemory Data { get; }

Property Value

Type:  ReadOnlyMemory
The data.

Reference

INotification Interface [969]
TwinCAT.Ads Namespace [151]

6.2.49.1.2  INotification.Handle Property

The notification handle

Namespace:  TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdca3e72bc0ea15da1c14
Syntax

C#

```csharp
uint Handle { get; }
```

**Property Value**

Type: UInt32
The handle.

**Remarks**

This can be the Client handle or the Server handle!

**Reference**

INotification Interface [969]
TwinCAT.Ads Namespace [151]

6.2.49.1.3 INotification.TimeStamp Property

Gets the time stamp of the INotification [969]

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstracts (in TwinCAT.Ads.Abstracts.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
DateTimeOffset TimeStamp { get; }
```

**Property Value**

Type: DateTimeOffset
The time stamp.

**Reference**

INotification Interface [969]
TwinCAT.Ads Namespace [151]

6.2.49.1.4 INotification.UserData Property

Attached UserData/Tag at the INotification [969]

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstracts (in TwinCAT.Ads.Abstracts.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Object UserData { get; }
```
**Property Value**

Type: Object  
The user data.

**Reference**

INotification Interface [969]  
TwinCAT.Ads Namespace [151]

### 6.2.50 INotificationSettings Interface

Interface for Notification Settings implements the `IComparable<T>`.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public interface INotificationSettings : IComparable<INotificationSettings>
```

The `INotificationSettings` type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CompareTo</code></td>
<td>Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object. (Inherited from <code>IComparable&lt;T&gt;</code>)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.Ads Namespace [151]  
System.IComparable<T>

### 6.2.50.1 INotificationSettings Methods

The `INotificationSettings` [972] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CompareTo</code></td>
<td>Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object. (Inherited from <code>IComparable&lt;T&gt;</code>)</td>
</tr>
</tbody>
</table>

**Reference**

INotificationSettings Interface [972]  
TwinCAT.Ads Namespace [151]
6.2.51    IRouterNotificationProvider Interface

Interface for AMS Router Notifications.

Namespace:  TwinCAT.Ads [151]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IRouterNotificationProvider
```

The IRouterNotificationProvider type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RouterStateChanged</td>
<td>Router state changed event.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]

6.2.51.1    IRouterNotificationProvider Events

The IRouterNotificationProvider [973] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RouterStateChanged</td>
<td>Router state changed event.</td>
</tr>
</tbody>
</table>

Reference

IRouterNotificationProvider Interface [973]

TwinCAT.Ads Namespace [151]

6.2.51.1.1    IRouterNotificationProvider.RouterStateChanged Event

Router state changed event.

Namespace:  TwinCAT.Ads [151]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
event EventHandler<AmsRouterNotificationEventArgs> RouterStateChanged
```
6.2.52 Notification Class

Class Notification. Implements the INotification.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.Notification

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

public class Notification : INotification

The Notification type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification(INotification)</td>
<td>Initializes a new instance of the Notification class.</td>
</tr>
<tr>
<td>Notification(UInt32, DateTimeOffset, Object, ReadOnlyMemory)</td>
<td></td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>The notification Data.</td>
</tr>
<tr>
<td>Handle</td>
<td>The notification handle</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Gets the time stamp of the INotification</td>
</tr>
<tr>
<td>UserData</td>
<td>Attached UserData/Tag at the INotification</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

### Reference

- TwinCAT.Ads Namespace [151]
- TwinCAT.Ads.INotification [969]

### 6.2.52.1 Notification Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification(INotification)</td>
<td>Initializes a new instance of the <code>Notification</code> [974] class.</td>
</tr>
<tr>
<td>Notification(UInt32, DateTimeOffset, Object, ReadOnlyMemory)</td>
<td>[976]</td>
</tr>
</tbody>
</table>

### Reference

- Notification Class [974]
- TwinCAT.Ads Namespace [151]

### 6.2.52.1.1 Notification Constructor (INotification)

Initializes a new instance of the `Notification` [974] class.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public Notification(
    INotification notification
)
```
Parameters

notification | Type: TwinCAT.Ads.INotification

The notification.

Reference

Notification Class [974]

Notification Overload [975]

TwinCAT.Ads Namespace [151]

6.2.52.1.2 Notification Constructor (UInt32, DateTimeOffset, Object, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Notification(
    uint handle,
    DateTimeOffset dateTime,
    Object userData,
    ReadOnlyMemory data
)
```

Parameters

handle | Type: System.UInt32

dateTime | Type: System.DateTimeOffset

userData | Type: System.Object

data | Type: ReadOnlyMemory

Reference

Notification Class [974]

Notification Overload [975]

TwinCAT.Ads Namespace [151]

6.2.52.2 Notification Properties

The Notification [974] type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data [151]</td>
<td>The notification Data.</td>
</tr>
<tr>
<td>Handle [151]</td>
<td>The notification handle</td>
</tr>
<tr>
<td>TimeStamp [151]</td>
<td>Gets the time stamp of the INotification [151]</td>
</tr>
<tr>
<td>UserData [151]</td>
<td>Attached UserData/Tag at the INotification [151]</td>
</tr>
</tbody>
</table>

### Reference

**Notification Class [151]**

**TwinCAT.Ads Namespace [151]**

#### 6.2.52.2.1 Notification.Data Property

The notification Data.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)  
**Version:** 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```
C#
public ReadOnlyMemory Data { get; }
```

**Property Value**

**Type:** ReadOnlyMemory

The data.

**Implements**

INotification.Data [151]

**Reference**

**Notification Class [151]**

**TwinCAT.Ads Namespace [151]**

#### 6.2.52.2.2 Notification.Handle Property

The notification handle

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)  
**Version:** 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
6.2.52.2.3 NotificationTimeStamp Property

Gets the time stamp of the INotification

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public DateTimeOffset TimeStamp { get; }

Property Value

Type: DateTimeOffset
The time stamp.

Implements

INotification.TimeStamp

Reference

Notification Class
TwinCAT.Ads Namespace

6.2.52.2.4 NotificationUserData Property

Attached UserData/Tag at the INotification

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public uint Handle { get; }

Property Value

Type: UInt32
The handle.

Implements

INotification.Handle

Reference

Notification Class
TwinCAT.Ads Namespace

TwinCAT.Ads Namespaces
Syntax

C#

```csharp
public Object UserData { get; }
```

Property Value

Type: Object
The user data.

Implements

INotification.UserData [971]

Reference

Notification Class [974]

TwinCAT.Ads Namespace [151]

6.2.52.3 Notification Methods

The Notification [974] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

Notification Class [974]

TwinCAT.Ads Namespace [151]

6.2.53 NotificationSettings Class

Notification communication settings
Inheritance Hierarchy

System.Object
   TwinCAT.Ads.NotificationSettings

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd3ca3e72bc0ea15da1c14

Syntax

C#

public class NotificationSettings : INotificationSettings, IComparable<INotificationSettings>

The NotificationSettings type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotificationSettings(AdsTransMode, Int32, Int32)</td>
<td>Initializes a new instance of the NotificationSettings class.</td>
</tr>
<tr>
<td>NotificationSettings(AdsTransMode, TimeSpan, TimeSpan)</td>
<td>Initializes a new instance of the NotificationSettings class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CycleTime</td>
<td>Gets or sets the cycle time (in milliseconds) for AdsNotifications.</td>
</tr>
<tr>
<td>Default</td>
<td>Gets the default Settings (AdsTransMode.OnChange, CycleTime 200 ms, MaxDelay: off)</td>
</tr>
<tr>
<td>ImmediatelyOnChange</td>
<td>Gets the settings for a 'Immediate on change' notification.</td>
</tr>
<tr>
<td>MaxDelay</td>
<td>Gets or sets the Maximum Delay Time (in milliseconds) for AdsNotifications.</td>
</tr>
<tr>
<td>NotificationMode</td>
<td>Gets or sets the ADS Transmission mode.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompareTo</td>
<td>Compares this NotificationSettings in term of priorities to the other NotificationSettings.</td>
</tr>
<tr>
<td>Equals</td>
<td>Equals (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Overrides Object.GetHashCode.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operators Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equality</td>
<td>Operator==</td>
</tr>
<tr>
<td>Inequality</td>
<td>Implements the != operator.</td>
</tr>
</tbody>
</table>

### Remarks

<table>
<thead>
<tr>
<th>AdsTransMode</th>
<th>Parameter semantic</th>
</tr>
</thead>
<tbody>
<tr>
<td>CyclicInContext</td>
<td>Value of parameter is interpreted as task context number IAdsSymbol.ContextMask/&gt;</td>
</tr>
<tr>
<td>OnChangeInContext</td>
<td>Value of parameter is interpreted as task context number IAdsSymbol.ContextMask/&gt;</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads Namespace [151]

### 6.2.53.1 NotificationSettings Constructor

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotificationSettings (AdsTransMode, Int32, Int32)</td>
<td>Initializes a new instance of the NotificationSettings class.</td>
</tr>
<tr>
<td>NotificationSettings (AdsTransMode, TimeSpan, TimeSpan)</td>
<td>Initializes a new instance of the NotificationSettings class.</td>
</tr>
</tbody>
</table>

### Reference

NotificationSettings Class [979]

TwinCAT.Ads Namespace [151]
6.2.53.1.1 NotificationSettings Constructor (AdsTransMode, Int32, Int32)

Initializes a new instance of the NotificationSettings [979] class.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public NotificationSettings(
    AdsTransMode mode,
    int cycleTime,
    int maxDelay
)
```

**Parameters**

- **mode**
  - Type: TwinCAT.Ads.AdsTransMode [639]
  - The ADS Transmission mode.

- **cycleTime**
  - Type: System.Int32
  - The cycle time in ms.

- **maxDelay**
  - Type: System.Int32
  - The maximum delay in ms

**Reference**

- NotificationSettings Class [979]
- NotificationSettings Overload [981]
- TwinCAT.Ads Namespace [151]

---

6.2.53.1.2 NotificationSettings Constructor (AdsTransMode, TimeSpan, TimeSpan)

Initializes a new instance of the NotificationSettings [979] class.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public NotificationSettings(
    AdsTransMode mode,
    TimeSpan cycleTime,
    TimeSpan maxDelay
)
```

**Parameters**

- **mode**
  - Type: TwinCAT.Ads.AdsTransMode [639]
  - The ADS Transmission mode.

- **cycleTime**
  - Type: System.TimeSpan
  - The cycle time in ms.
maxDelay

Type: System.TimeSpan
The maximum delay in ms

Reference

NotificationSettings Class [979]

NotificationSettings Overload [981]

TwinCAT.Ads Namespace [151]

6.2.53.2 NotificationSettings Properties

The NotificationSettings [979] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CycleTime [983]</td>
<td>Gets or sets the cycle time (in milliseconds) for AdsNotifications.</td>
</tr>
<tr>
<td>Default [984]</td>
<td>Gets the default Settings (AdsTransMode.OnChange, CycleTime 200 ms, MaxDelay: off)</td>
</tr>
<tr>
<td>ImmediatelyOnChange</td>
<td>Gets the settings for a 'Immediate on change' notification.</td>
</tr>
<tr>
<td>MaxDelay [985]</td>
<td>Gets or sets the Maximum Delay Time (in milliseconds) for AdsNotifications.</td>
</tr>
<tr>
<td>NotificationMode [985]</td>
<td>Gets or sets the ADS Transmission mode.</td>
</tr>
</tbody>
</table>

Reference

NotificationSettings Class [979]

TwinCAT.Ads Namespace [151]

6.2.53.2.1 NotificationSettings.CycleTime Property

Gets or sets the cycle time (in milliseconds) for AdsNotifications.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

class

cpublic int CycleTime { get; }

Property Value

Type: Int32
The cycle time.
Remarks
The ADS server checks if the value changes in this time slice. The unit is 1ms

Reference
NotificationSettings Class [979]
TwinCAT.Ads Namespace [151]

6.2.53.2.2 NotificationSettings.Default Property

Gets the default Settings (AdsTransMode.OnChange, CycleTime 200 ms, MaxDelay: off)

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static NotificationSettings Default { get; }

Property Value

Type: NotificationSettings [979]
The default.

Reference
NotificationSettings Class [979]
TwinCAT.Ads Namespace [151]

6.2.53.2.3 NotificationSettings.ImmediatelyOnChange Property

Gets the settings for a 'Immediate on change' notification.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static NotificationSettings ImmediatelyOnChange { get; }

Property Value

Type: NotificationSettings [979]
The immediately on change.

Remarks
AdsTransMode.OnChange, CycleTime: 0 ms, MaxDelay: off)
6.2.53.2.4 NotificationSettings.MaxDelay Property

Gets or sets the Maximum Delay Time (in milliseconds) for AdsNotifications.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public int MaxDelay { get; }
```

**Property Value**

Type: `Int32`
The maximum Delay time for ADS Notifications.

---

6.2.53.2.5 NotificationSettings.NotificationMode Property

Gets or sets the ADS Transmission mode.

**Namespace:** TwinCAT.Ads [151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public AdsTransMode NotificationMode { get; }
```

**Property Value**

Type: `AdsTransMode` [639]
The Transmission mode.

---

6.2.53.3 NotificationSettings Methods

The NotificationSettings [979] type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompareTo [986]</td>
<td>Compares this NotificationSettings [979] in term of priorities to the other NotificationSettings [979].</td>
</tr>
<tr>
<td>Equals [987]</td>
<td>Equals (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

NotificationSettings Class [979]

TwinCAT.Ads Namespace [151]

6.2.53.3.1 NotificationSettings.CompareTo Method

Compares this NotificationSettings [979] in term of priorities to the other NotificationSettings [979].

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int CompareTo(
    INotificationSettings other
)
```

Parameters

other                  Type: TwinCAT.Ads.INotificationSettings [972]
The other.

Return Value

Type: Int32
1: this has higher priority (shorter times), 0: Equal, 1: Lower priority

Implements

IComparable<T.CompareTo(T)

Reference

NotificationSettings Class [979]
6.2.53.3.2  NotificationSettings.Equals Method

Equals

Namespace:  TwinCAT.Ads [151]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public override bool Equals(
    Object obj
)

Parameters

obj  Type:  System.Object
The object to compare with the current object.

Return Value

Type:  Boolean
ture if the specified Object is equal to this instance; otherwise, false.

Reference

NotificationSettings Class [979]
TwinCAT.Ads Namespace [151]

6.2.53.3.3  NotificationSettings.GetHashCode Method

Gets the HashCode of the Address

Namespace:  TwinCAT.Ads [151]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public override int GetHashCode()

Return Value

Type:  Int32
A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

Reference

NotificationSettings Class [979]
TwinCAT.Ads Namespace [151]
6.2.53.4 NotificationSettings Operators

The NotificationSettings [979] type exposes the following members.

Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equality</td>
<td>Operator==</td>
</tr>
<tr>
<td>Inequality</td>
<td>Implements the != operator.</td>
</tr>
</tbody>
</table>

Reference

NotificationSettings Class [979]
TwinCAT.Ads Namespace [151]

6.2.53.4.1 NotificationSettings.Equality Operator

Operator==

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C# public static bool operator ==(
    NotificationSettings o1,
    NotificationSettings o2
)

Parameters


Return Value

Type: Boolean The result of the operator.

Reference

NotificationSettings Class [979]
TwinCAT.Ads Namespace [151]
6.2.53.4.2 NotificationSettings.Inequality Operator

Implements the != operator.

**Namespace:** TwinCAT.Ads [➤ 151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public static bool operator !=(  
  NotificationSettings o1,
  NotificationSettings o2
)
```

**Parameters**

- **o1**  
  Type: TwinCAT.Ads.NotificationSettings [➤ 979]
  The o1.

- **o2**  
  Type: TwinCAT.Ads.NotificationSettings [➤ 979]
  The o2.

**Return Value**

Type: Boolean  
The result of the operator.

**Reference**

NotificationSettings Class [➤ 979]
TwinCAT.Ads Namespace [➤ 151]

6.2.54 ResultAds Class

Base class for an (asynchronous) ADS Task Result

**Inheritance Hierarchy**

System.Object  
TwinCAT.Ads.ResultAds

**Namespace:** TwinCAT.Ads [➤ 151]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public class ResultAds
```

The ResultAds type exposes the following members.
Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultAds</td>
<td>Initializes a new instance of the ResultAds class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets an empty ResultAds initialized to None.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result object.</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds state is failed.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds state is succeeded.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateError(AdsErrorCode)</td>
<td>Creates an Error Result.</td>
</tr>
<tr>
<td>CreateError.TValue.(AdsErrorCode, TValue)</td>
<td>Creates an Error Result.</td>
</tr>
<tr>
<td>CreateSuccess</td>
<td>Creates a success result.</td>
</tr>
<tr>
<td>CreateSuccess.TValue.(TValue)</td>
<td>Creates a success result.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from System.Object)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from System.Object)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from System.Object)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from System.Object)</td>
</tr>
<tr>
<td>SetError [997]</td>
<td>Sets the error state of this ResultAds</td>
</tr>
<tr>
<td>ToString [997]</td>
<td>Returns a string that represents the current object. (Inherited from System.Object)</td>
</tr>
</tbody>
</table>

Remarks

The base class is used whenever an asynchronous task communicates via ADS and should return its AdsErrorCode [575] within its tasks result.
Reference

TwinCAT.Ads Namespace [151]

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds
    TwinCAT.Ads.ResultAnyValue [998]
    TwinCAT.Ads.ResultDeviceInfo [1001]
    TwinCAT.Ads.ResultHandle [1005]
    TwinCAT.Ads.ResultRead [1008]
    TwinCAT.Ads.ResultReadAdsState [1010]
    TwinCAT.Ads.ResultReadDeviceState [1016]
    TwinCAT.Ads.ResultRpcMethod [1025]
    TwinCAT.Ads.ResultValue.TValue. [1029]
    TwinCAT.Ads.ResultWrite [1032]
    TwinCAT.Ads.SumCommand.ResultSumCommand [1210]

6.2.54.1 ResultAds Constructor

Initializes a new instance of the ResultAds [989] class.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b609593fddca3e72bc0ea15da1c14

Syntax

C#

protected ResultAds(
    AdsErrorCode errorCode
)

Parameters

eretCode Type: TwinCAT.Ads.AdsErrorCode [575]

The Ads ErrorCode

Reference

ResultAds Class [989]

TwinCAT.Ads Namespace [151]

6.2.54.2 ResultAds Properties

The ResultAds [989] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets an empty ResultAds [989] initialized to None [575].</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object.</td>
</tr>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed.</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded.</td>
</tr>
</tbody>
</table>

#### Reference

ResultAds Class [989]

TwinCAT.Ads Namespace [151]

#### 6.2.54.2.1 ResultAds.Empty Property

Gets an empty ResultAds [989] initialized to None [575].

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public static ResultAds Empty { get; }
```

**Property Value**

Type: ResultAds [989]

The empty.

**Reference**

ResultAds Class [989]

TwinCAT.Ads Namespace [151]

#### 6.2.54.2.2 ResultAds.ErrorCode Property

Gets the ADS Error code bound to this Result [989] object.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public AdsErrorCode ErrorCode { get; }
```

**Property Value**

Type: AdsErrorCode [575]

The error code.
6.2.54.2.3 ResultAds.Failed Property

Gets a value indicating whether the ResultAds state is failed.

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool Failed { get; }
```

Property Value

Type: Boolean
true if failed; otherwise, false.

Reference

ResultAds Class

TwinCAT.Ads Namespace

---

6.2.54.2.4 ResultAds.Succeeded Property

Gets a value indicating whether the ResultAds state is succeeded.

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool Succeeded { get; }
```

Property Value

Type: Boolean
true if succeeded; otherwise, false.

Reference

ResultAds Class

TwinCAT.Ads Namespace
6.2.54.3 ResultAds Methods

The ResultAds [989] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateError(AdsErrorCode) [995]</td>
<td>Creates an Error Result.</td>
</tr>
<tr>
<td>CreateError.TValue.(AdsErrorCode, TValue) [995]</td>
<td>Creates an Error Result.</td>
</tr>
<tr>
<td>CreateSuccess [996]</td>
<td>Creates a success result.</td>
</tr>
<tr>
<td>CreateSuccess.TValue.(TValue) [997]</td>
<td>Creates a success result.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError [997]</td>
<td>Sets the error state of this ResultAds [989]</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

ResultAds Class [989]
TwinCAT.Ads Namespace [151]

6.2.54.3.1 ResultAds.CreateError Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateError(AdsErrorCode) [995]</td>
<td>Creates an Error Result.</td>
</tr>
<tr>
<td>CreateError.TValue.(AdsErrorCode, TValue) [995]</td>
<td>Creates an Error Result.</td>
</tr>
</tbody>
</table>

Reference

ResultAds Class [989]
ResultAds.CreateError Method (AdsErrorCode)

Creates an Error Result.

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static ResultAds CreateError(
    AdsErrorCode errorCode
)
```

Parameters

- `errorCode` Type: TwinCAT.Ads.AdsErrorCode
  The error code.

Return Value

Type: ResultAds

ResultValue<T>.

Reference

ResultAds Class

CreateError Overload

TwinCAT.Ads Namespace

ResultAds.CreateError.TValue. Method (AdsErrorCode, TValue)

Creates an Error Result.

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static ResultValue<TValue> CreateError<TValue>(
    AdsErrorCode errorCode,
    TValue defaultValue
)
```

Parameters

- `errorCode` Type: TwinCAT.Ads.AdsErrorCode
  The error code.

- `defaultValue` Type: TValue
  The default value.
### Type Parameters

**TValue**
The type of the t value.

### Return Value

Type: `ResultValue<T>.TValue`

### Reference

- ResultAds Class [989]
- CreateError Overload [994]
- TwinCAT.Ads Namespace [151]

#### 6.2.54.3.2 ResultAds.CreateSuccess Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateSuccess</td>
<td>Creates a success result.</td>
</tr>
<tr>
<td>CreateSuccess.TValue</td>
<td></td>
</tr>
</tbody>
</table>

### Reference

- ResultAds Class [989]
- TwinCAT.Ads Namespace [151]

#### ResultAds.CreateSuccess Method

Creates a success result.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static ResultAds CreateSuccess()
```

### Return Value

Type: `ResultAds [989]`

### Reference

- ResultAds Class [989]
ResultAds.CreateSuccess<TValue>. Method (TValue)

Creates a success result.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public static ResultValue<TValue> CreateSuccess<TValue>(
    TValue value
)
```

**Parameters**

value  
Type: TValue  
The value.

**Type Parameters**

TValue

**Return Value**

Type: ResultValue<TValue>.TValue. ResultValue<T>.

**Reference**

ResultAds Class

CreateSuccess Overload

TwinCAT.Ads Namespace

---

6.2.54.3.3 ResultAds.SetError Method

Sets the error state of this ResultAds.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public void SetError(
    AdsErrorCode error
)
```
Parameters

error Type: TwinCAT.Ads.AdsErrorCode
The error.

Reference

ResultAds Class [989]
TwinCAT.Ads Namespace [151]

6.2.55 ResultAnyValue Class

Result object for asynchronous reading an 'AnyValue'/Primitive Value via tasks.

Inheritance Hierarchy

System.Object
TwinCAT.Ads.ResultAds [989]
TwinCAT.Ads.ResultAnyValue

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class ResultAnyValue : ResultAds

The ResultAnyValue type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultAnyValue</td>
<td>Initializes a new instance of the ResultAnyValue struct.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets the empty ResultAnyValue object.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result object. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds state is failed. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds state is succeeded. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>Value</td>
<td>Gets the read value.</td>
</tr>
</tbody>
</table>
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds (Inherited from ResultAds)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]

6.2.55.1 ResultAnyValue Constructor

Initializes a new instance of the ResultAnyValue struct.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public ResultAnyValue(    AdsErrorCode errorCode,    Object value)
```

Parameters

errorCode Type: TwinCAT.Ads.AdsErrorCode [575]
The error code.

value Type: System.Object
The value.

Reference

ResultAnyValue Class [998]
TwinCAT.Ads Namespace [151]

6.2.55.2 ResultAnyValue Properties

The ResultAnyValue type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Value [1000]</td>
<td>Gets the read value.</td>
</tr>
</tbody>
</table>

Reference

ResultAnyValue Class [998]

TwinCAT.Ads Namespace [151]

6.2.55.2.1 ResultAnyValue.Empty Property

Gets the empty ResultAnyValue [998] object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static ResultAnyValue Empty { get; }
```

Property Value

Type: ResultAnyValue [998]
The empty / unprocessed result.

Reference

ResultAnyValue Class [998]

TwinCAT.Ads Namespace [151]

6.2.55.2.2 ResultAnyValue.Value Property

Gets the read value.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#
public Object Value { get; }

Property Value
Type: Object

Reference
ResultAnyValue Class [998]
TwinCAT.Ads Namespace [151]

6.2.55.3 ResultAnyValue Methods

The ResultAnyValue [998] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference
ResultAnyValue Class [998]
TwinCAT.Ads Namespace [151]

6.2.56 ResultDeviceInfo Class

Ads Task Result forDeviceInfo [1003] requests (async operations).

Inheritance Hierarchy

System.Object
TwinCAT.Ads.ResultAds [989]
TwinCAT.Ads.ResultDeviceInfo

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public class ResultDeviceInfo : ResultAds
```

The `ResultDeviceInfo` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceInfo</td>
<td>Gets the device information (Task result)</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this <code>Result</code> object. (Inherited from <code>ResultAds</code>)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the <code>ResultAds</code> state is failed. (Inherited from <code>ResultAds</code>)</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the Device</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the <code>ResultAds</code> state is succeeded. (Inherited from <code>ResultAds</code>)</td>
</tr>
<tr>
<td>Version</td>
<td>The ADS Version of the Device.</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this <code>ResultAds</code> (Inherited from <code>ResultAds</code>)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.Ads Namespace [151]

6.2.56.1 ResultDeviceInfo Properties

The `ResultDeviceInfo` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceInfo</td>
<td>Gets the device information (Task result)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Name [1003]</td>
<td>The name of the Device</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>

### Reference

- ResultDeviceInfo Class [1001]
- TwinCAT.Ads Namespace [151]

#### 6.2.56.1.1 ResultDeviceInfo.DeviceInfo Property

Gets the device information (Task result)

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public DeviceInfo DeviceInfo { get; }
```

**Property Value**

Type: DeviceInfo [698]

The device information.

---

**Reference**

- ResultDeviceInfo Class [1001]
- TwinCAT.Ads Namespace [151]

#### 6.2.56.1.2 ResultDeviceInfo.Name Property

The name of the Device

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public string Name { get; }
```
6.2.56.1.3 ResultDeviceInfo.Version Property

The ADS Version of the the Device.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public AdsVersion Version { get; }

6.2.56.2 ResultDeviceInfo Methods

The ResultDeviceInfo type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds (Inherited from ResultAds)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
6.2.57 ResultHandle Class

Result object for asynchronous registering an ADS Handle via tasks.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public class ResultHandle : ResultAds

The ResultHandle type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets the empty ResultHandle object.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result object. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds state is failed. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>Handle</td>
<td>Gets the registered handle</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds state is succeeded. (Inherited from ResultAds.)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.Ads Namespace

---

### 6.2.57.1 ResultHandle Properties

The ResultHandle type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets the empty ResultHandle object.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result object. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds state is failed. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>Handle</td>
<td>Gets the registered handle</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds state is succeeded. (Inherited from ResultAds.)</td>
</tr>
</tbody>
</table>

**Reference**

ResultHandle Class

---

### 6.2.57.1.1 ResultHandle.Empty Property

Gets the empty ResultHandle object.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static ResultHandle Empty { get; }
```

**Property Value**

Type: ResultHandle

The empty / unprocessed result.
6.2.57.1.2 ResultHandle.Handle Property

Gets the registered handle

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public uint Handle { get; }
```

**Property Value**

Type: UInt32

Reference

ResultHandle Class

TwinCAT.Ads Namespace

6.2.57.2 ResultHandle Methods

The ResultHandle type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

ResultHandle Class

TwinCAT.Ads Namespace
6.2.58 ResultRead Class

Asynchronous ADS Read result.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [¶ 989]
  TwinCAT.Ads.ResultRead
    TwinCAT.Ads.ResultReadBytes [¶ 1013]
    TwinCAT.Ads.ResultReadWrite [¶ 1019]

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: S.0.294+Branch.releases-S.0.Sha.90bb9a1b43b6095934fددca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class ResultRead : ResultAds
```

The ResultRead type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets the empty ResultRead object.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result [¶ 989] object. (Inherited from ResultAds [¶ 989].)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds [¶ 989] state is failed. (Inherited from ResultAds [¶ 989].)</td>
</tr>
<tr>
<td>ReadBytes</td>
<td>Gets the number of Read bytes.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds [¶ 989] state is succeeded. (Inherited from ResultAds [¶ 989].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds [¶ 989] (Inherited from ResultAds [¶ 989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
6.2.58.1  ResultRead Properties

The ResultRead [1008] type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets the empty ResultRead [1008] object.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ReadBytes</td>
<td>Gets the number of Read bytes.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>

### ResultRead.Empty Property

Gets the empty ResultRead [1008] object.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

C#

```csharp
public static ResultRead Empty { get; }
```

**Property Value**

Type: ResultRead [1008]

The empty / unprocessed result.

**Reference**

ResultRead Class [1008]

TwinCAT.Ads Namespace [151]
### 6.2.58.1.2 ResultRead.ReadByte Property

 Gets the number of Read bytes.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public int ReadBytes { get; }
```

**Property Value**

Type: Int32

**Reference**

ResultRead Class [1008]

TwinCAT.Ads Namespace [151]

### 6.2.58.2 ResultRead Methods

The ResultRead [1008] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>Equals</td>
</tr>
<tr>
<td></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>==</td>
<td>Finalize</td>
</tr>
<tr>
<td></td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>==</td>
<td>GetHashCode</td>
</tr>
<tr>
<td></td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>==</td>
<td>GetType</td>
</tr>
<tr>
<td></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>==</td>
<td>MemberwiseClone</td>
</tr>
<tr>
<td></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>==</td>
<td>SetError [997]</td>
</tr>
<tr>
<td></td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds</td>
</tr>
<tr>
<td></td>
<td>[989].)</td>
</tr>
<tr>
<td>==</td>
<td>ToString</td>
</tr>
<tr>
<td></td>
<td>Returns a string that represents the current object. (Inherited from Object.</td>
</tr>
</tbody>
</table>

**Reference**

ResultRead Class [1008]

TwinCAT.Ads Namespace [151]

### 6.2.59 ResultReadAdsState Class

Result object for asynchronous reading AdsStates via tasks.
Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [989]
    TwinCAT.Ads.ResultReadAdsState

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public class ResultReadAdsState : ResultAds

The ResultReadAdsState type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets the empty ResultReadAdsState object.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>State</td>
<td>Gets the ADS state.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError [997]</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]
### 6.2.59.1 ResultReadAdsState Properties

The **ResultReadAdsState** type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets the empty <strong>ResultReadAdsState</strong> object.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this <strong>Result</strong> object. (Inherited from <strong>ResultAds</strong>.)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the <strong>ResultAds</strong> state is failed. (Inherited from <strong>ResultAds</strong>.)</td>
</tr>
<tr>
<td>State</td>
<td>Gets the ADS state.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the <strong>ResultAds</strong> state is succeeded. (Inherited from <strong>ResultAds</strong>.)</td>
</tr>
</tbody>
</table>

Reference

**ResultReadAdsState Class**

TwinCAT.Ads Namespace

### 6.2.59.1.1 ResultReadAdsState.Empty Property

Gets the empty **ResultReadAdsState** object.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public static ResultReadAdsState Empty { get; }
```

**Property Value**

Type: **ResultReadAdsState**  
The empty / unprocessed result.

Reference

**ResultReadAdsState Class**

TwinCAT.Ads Namespace
Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsState State { get; }
```

Property Value

Type: AdsState [626]

Reference

ResultReadAdsState Class [1010]
TwinCAT.Ads Namespace [151]

6.2.59.2 ResultReadAdsState Methods

The ResultReadAdsState [1010] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

ResultReadAdsState Class [1010]
TwinCAT.Ads Namespace [151]

6.2.60 ResultReadBytes Class

ADS Task Result returning Read data for async Read operations.
Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [989]
    TwinCAT.Ads.ResultRead [1008]
      TwinCAT.Ads.ResultReadWriteBytes
        TwinCAT.Ads.ResultReadBytes [1022]

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class ResultReadBytes : ResultRead
```

The ResultReadBytes type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data [1015]</td>
<td>The read bytes as continuous region of memory.</td>
</tr>
<tr>
<td>Empty [1016]</td>
<td>Gets the empty ResultReadBytes object.</td>
</tr>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ReadBytes [1010]</td>
<td>Gets the number of Read bytes. (Inherited from ResultRead [1008].)</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError [997]</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
6.2.60.1 ResultReadBytes Properties

The ResultReadBytes [1013] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data [1015]</td>
<td>The read bytes as continuous region of memory.</td>
</tr>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ReadBytes [1010]</td>
<td>Gets the number of Read bytes. (Inherited from ResultRead [1008].)</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>

Reference

ResultReadBytes Class [1013]
TwinCAT.Ads Namespace [151]

6.2.60.1.1 ResultReadBytes.Data Property

The read bytes as continuous region of memory.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public ReadOnlyMemory Data { get; }
```

Property Value

Type: ReadOnlyMemory

Reference

ResultReadBytes Class [1013]
TwinCAT.Ads Namespace [151]
6.2.60.1.2 ResultReadBytes.Empty Property

Gets the empty `ResultReadBytes` object.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd2ca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static ResultReadBytes Empty { get; }
```

**Property Value**

Type: `ResultReadBytes`

The empty / unprocessed result.

**Reference**

- ResultReadBytes Class
- TwinCAT.Ads Namespace

### 6.2.60.2 ResultReadBytes Methods

The `ResultReadBytes` type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>SetError</code></td>
<td>Sets the error state of this <code>ResultAds</code> (Inherited from <code>ResultAds</code>).</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

**Reference**

- ResultReadBytes Class
- TwinCAT.Ads Namespace

### 6.2.61 ResultReadDeviceState Class

Result object for asynchronous ADS ReadDeviceState tasks.
Inheritance Hierarchy


Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class ResultReadDeviceState : ResultAds

The ResultReadDeviceState type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets the empty ResultReadDeviceState object.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this ResultAds object. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds state is failed. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>State</td>
<td>The Device state.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds state is succeeded. (Inherited from ResultAds.)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds object. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace
6.2.61.1 **ResultReadDeviceState Properties**

The `ResultReadDeviceState` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets the empty <code>ResultReadDeviceState</code> object.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this <code>Result</code> object. (Inherited from <code>ResultAds</code>.)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the <code>ResultAds</code> state is failed. (Inherited from <code>ResultAds</code>.)</td>
</tr>
<tr>
<td>State</td>
<td>The Device state.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the <code>ResultAds</code> state is succeeded. (Inherited from <code>ResultAds</code>.)</td>
</tr>
</tbody>
</table>

### Reference

- `ResultReadDeviceState Class`  
- `TwinCAT.Ads Namespace`

---

6.2.61.1.1 **ResultReadDeviceState.Empty Property**

Gets the empty `ResultReadDeviceState` object.

**Namespace:** TwinCAT.Ads  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)  
**Version:** 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static ResultReadDeviceState Empty { get; }
```

### Property Value

Type: `ResultReadDeviceState`  
The empty / unprocessed result.

### Reference

- `ResultReadDeviceState Class`  
- `TwinCAT.Ads Namespace`

---

6.2.61.1.2 **ResultReadDeviceState.State Property**

The Device state.
**Namespace**: TwinCAT.Ads [151]

**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public StateInfo State { get; }
```

**Property Value**

Type: StateInfo [1041]

**Reference**

ResultReadDeviceState Class [1016]

TwinCAT.Ads Namespace [151]

**6.2.61.2 ResultReadDeviceState Methods**

The ResultReadDeviceState [1016] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds</td>
</tr>
<tr>
<td></td>
<td>[989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Reference**

ResultReadDeviceState Class [1016]

TwinCAT.Ads Namespace [151]

**6.2.62 ResultReadWrite Class**

Result object for asynchronous ADS ReadWrite tasks.
Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds
    TwinCAT.Ads.ResultRead
      TwinCAT.Ads.ResultReadWrite

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class ResultReadWrite : ResultRead

The ResultReadWrite type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty [1021]</td>
<td>Gets the empty ResultReadWrite object.</td>
</tr>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ReadBytes [1010]</td>
<td>Gets the number of Read bytes. (Inherited from ResultRead [1008].)</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError [997]</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [151]
6.2.62.1 ResultReadWrite Properties

The ResultReadWrite [1019] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ReadBytes [1010]</td>
<td>Gets the number of Read bytes. (Inherited from ResultRead [1008].)</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>

Reference

ResultReadWrite Class [1019]

TwinCAT.Ads Namespace [151]

6.2.62.1.1 ResultReadWrite.Empty Property

Gets the empty ResultReadWrite [1019] object.

Namespace: TwinCAT.Ads [151]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static ResultReadWrite Empty { get; }
```

Property Value

Type: ResultReadWrite [1019]
The empty / unprocessed result.

Reference

ResultReadWrite Class [1019]

TwinCAT.Ads Namespace [151]

6.2.62.2 ResultReadWrite Methods

The ResultReadWrite [1019] type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

ResultReadWrite Class [1019]
TwinCAT.Ads Namespace [151]

6.2.63 ResultReadWriteBytes Class

Result object for asynchronous ADS ReadWrite tasks.

Inheritance Hierarchy

System.Object
TwinCAT.Ads.ResultAds [989]
TwinCAT.Ads.ResultRead [1008]
TwinCAT.Ads.ResultReadBytes [1013]
TwinCAT.Ads.ResultReadWriteBytes

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class ResultReadWriteBytes : ResultReadBytes
```

The ResultReadWriteBytes type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data [1015]</td>
<td>The read bytes as continuous region of memory. (Inherited from ResultReadWriteBytes [1013].)</td>
</tr>
<tr>
<td>Empty [1024]</td>
<td>Gets the empty ResultReadWriteBytes object.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ReadBytes [1010]</td>
<td>Gets the number of Read bytes. (Inherited from ResultRead [1008].)</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError [997]</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

#### Reference

TwinCAT.Ads Namespace [151]

### 6.2.63.1 ResultReadWriteBytes Properties

The ResultReadWriteBytes [1022] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data [1015]</td>
<td>The read bytes as continuous region of memory. (Inherited from ResultReadWriteBytes [1022].)</td>
</tr>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ReadBytes [1010]</td>
<td>Gets the number of Read bytes. (Inherited from ResultRead [1008].)</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>
6.2.63.1.1 ResultReadWriteBytes.Empty Property

Gets the empty ResultReadWriteBytes [1022] object.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public static ResultReadWriteBytes Empty { get; }
```

**Property Value**

Type: ResultReadWriteBytes [1022]
The empty / unprocessed result.

**Methods**

The ResultReadWriteBytes [1022] type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td><code>Equals</code></td>
</tr>
<tr>
<td><img src="emptyicon" alt="" /></td>
<td><code>Finalize</code></td>
</tr>
<tr>
<td><img src="hashicon" alt="" /></td>
<td><code>GetHashCode</code></td>
</tr>
<tr>
<td><img src="typeicon" alt="" /></td>
<td><code>GetType</code></td>
</tr>
<tr>
<td>![memberwiseclone]</td>
<td><code>MemberwiseClone</code></td>
</tr>
<tr>
<td>![seterror]</td>
<td><code>SetErro</code> [997]</td>
</tr>
<tr>
<td>![tostring]</td>
<td><code>ToString</code></td>
</tr>
</tbody>
</table>

**Reference**

ResultReadWriteBytes Class [1022]

TwinCAT.Ads Namespace [151]
TwinCAT.Ads Namespaces

6.2.64 ResultRpcMethod Class

Class representing a result of an asynchronous RpcMethod call. Implements the ResultAds

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds
    TwinCAT.Ads.ResultRpcMethod

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public class ResultRpcMethod : ResultAds

The ResultRpcMethod type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultRpcMethod</td>
<td>Initializes a new instance of the ResultRpcMethod struct.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets the Empty Result (initialized to None and default ReturnValue (NULL).</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result object. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds state is failed. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>OutValues</td>
<td>Gets the output parameter values.</td>
</tr>
<tr>
<td>ReturnValue</td>
<td>The (optional) return value of the RPC Method.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds state is succeeded. (Inherited from ResultAds.)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>SetError [997]</td>
<td>Sets the error state of this <code>ResultAds [989]</code> (Inherited from <code>ResultAds [989]</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

#### Remarks
Optionally this Result holds an `ReturnValue [1028]` if the RpcCall is not void.

#### Reference
- TwinCAT.Ads Namespace [151]
- TwinCAT.Ads.ResultAds [989]

### 6.2.64.1 ResultRpcMethod Constructor


**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public ResultRpcMethod(
    AdsErrorCode errorCode,
    Object returnValue,
    Object[] outParameters
)
```

#### Parameters

- **errorCode**
  - Type: TwinCAT.Ads.AdsErrorCode [575]  
    The error code.

- **returnValue**
  - Type: System.Object  
    The value.

- **outParameters**
  - Type: System.Object  
    The out parameters.

#### Reference

- ResultRpcMethod Class [1025]
- TwinCAT.Ads Namespace [151]

### 6.2.64.2 ResultRpcMethod Properties

The `ResultRpcMethod [1025]` type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty [1027]</td>
<td>Gets the Empty Result (initialized to None [575] and default ReturnValue [1028] (NULL)).</td>
</tr>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>OutValues [1027]</td>
<td>Gets the output parameter values.</td>
</tr>
<tr>
<td>ReturnValue [1028]</td>
<td>The (optional) return value of the RPC Method.</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>

Reference

ResultRpcMethod Class [1025]
TwinCAT.Ads Namespace [151]

6.2.64.2.1 ResultRpcMethod.Empty Property

Gets the Empty Result (initialized to None [575] and default ReturnValue [1028] (NULL)).

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Extensions (in TwinCAT.Ads.Extensions.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static ResultRpcMethod Empty { get; }
```

Property Value

Type: ResultRpcMethod [1025]
The empty.

Reference

ResultRpcMethod Class [1025]
TwinCAT.Ads Namespace [151]

6.2.64.2.2 ResultRpcMethod.OutValues Property

Gets the output parameter values.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Extensions (in TwinCAT.Ads.Extensions.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

public Object[] OutValues { get; }

Property Value

Type: Object
The output parameters.

Reference

ResultRpcMethod Class [1025]
TwinCAT.Ads Namespace [151]

6.2.64.2.3 ResultRpcMethod.ReturnValue Property

The (optional) return value of the RPC Method.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: S.0.294+Branch.releases-S.0.Sha.9b0b9a1b43b8095934fddca3e72bc0ea15da1c14

Syntax

C#

public Object ReturnValue { get; }

Property Value

Type: Object

Reference

ResultRpcMethod Class [1025]
TwinCAT.Ads Namespace [151]

6.2.64.3 ResultRpcMethod Methods

The ResultRpcMethod [1025] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="SetError" /></td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td><img src="image" alt="ToString" /></td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

#### Reference
- ResultRpcMethod Class [1025]
- TwinCAT.Ads Namespace [151]

### 6.2.65 ResultValue.TValue. Class

ADS Result object returning a generic value result (TValue) (asynchronous read). Implements the ResultAds [989].

#### Inheritance Hierarchy

- System.Object
  - TwinCAT.Ads.ResultAds [989]
    - TwinCAT.Ads.ResultValue.TValue.
      - TwinCAT.TypeSystem.ResultDataTypes [2347]
      - TwinCAT.TypeSystem.ResultSymbols.T. [2356]

#### Namespace: TwinCAT.Ads [151]

- Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public class ResultValue<TValue> : ResultAds
```

#### Type Parameters

- TValue: The type of the result value.

The ResultValue.TValue. type exposes the following members.

#### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="ResultValue.TValue." /></td>
<td>Initializes a new instance of the ResultValue class.</td>
</tr>
</tbody>
</table>

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Empty" /></td>
<td>Gets the Empty result initialized to NoError [575] and the default of TValue.</td>
</tr>
<tr>
<td><img src="image" alt="ErrorCode" /></td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td><img src="image" alt="Failed" /></td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### TwinCAT.Ads ResultAds [989]

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Value [1032]</td>
<td>The value object.</td>
</tr>
</tbody>
</table>

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError [997]</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

#### Remarks

The ResultValue.TValue. type is used in generic ADS read operations, where a value is returned.

#### Reference

- TwinCAT.Ads Namespace [151]
- TwinCAT.Ads.ResultAds [989]

### 6.2.65.1 ResultValue.TValue. Constructor

Initializes a new instance of the ResultValue class.

- **Namespace:** TwinCAT.Ads [151]
- **Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public ResultValue(
    AdsErrorCode errorCode,
    TValue value
)
```

#### Parameters

- **errorCode**
  - Type: TwinCAT.Ads.AdsErrorCode [575]
  - The error code.

- **value**
  - Type: TValue [1029]
  - The value.
6.2.65.2 ResultValue.TValue. Properties

The ResultValue.TValue. generic type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty [1031]</td>
<td>Gets the Empty result initialized to NoError [575] and the default of TValue.</td>
</tr>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Value [1032]</td>
<td>The value object.</td>
</tr>
</tbody>
</table>

Reference

ResultValue.TValue.Class [1029]
TwinCAT.Ads Namespace [151]

6.2.65.2.1 ResultValue.TValue..Empty Property

Gets the Empty result initialized to NoError [575] and the default of TValue.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static ResultValue<TValue> Empty { get; }
```

Property Value

Type: ResultValue [1029], TValue [1029].
The empty.

Reference

ResultValue.TValue.Class [1029]
TwinCAT.Ads Namespace [151]
6.2.65.2.2 ResultValue.TValue..Value Property

The value object.

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public TValue Value { get; }
```

**Property Value**

Type: TValue

**Reference**

ResultValue.TValue. Class

TwinCAT.Ads Namespace

6.2.65.3 ResultValue.TValue. Methods

The ResultValue.TValue. generic type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError [997]</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Reference**

ResultValue.TValue. Class

TwinCAT.Ads Namespace

6.2.66 ResultWrite Class

Result for asynchronous ADS write tasks.
Inheritance Hierarchy

System.Object
	TwinCAT.Ads.ResultAds [➠ 989]
	TwinCAT.Ads.ResultWrite

Namespace: TwinCAT.Ads [➠ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class ResultWrite : ResultAds
```

The ResultWrite type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets the empty ResultWrite object.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result [➠ 989] object. (Inherited from ResultAds [➠ 989].)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds [➠ 989] state is failed. (Inherited from ResultAds [➠ 989].)</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds [➠ 989] state is succeeded. (Inherited from ResultAds [➠ 989].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds [➠ 989] (Inherited from ResultAds [➠ 989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace [➠ 151]

6.2.66.1 ResultWrite Properties

The ResultWrite [➠ 1032] type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>

Reference

ResultWrite Class [1032]
TwinCAT.Ads Namespace [151]

6.2.66.1.1 ResultWrite.Empty Property

Gets the empty ResultWrite [1032] object.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static ResultWrite Empty { get; }
```

Property Value

Type: ResultWrite [1032]
The empty / unprocessed result.

Reference

ResultWrite Class [1032]
TwinCAT.Ads Namespace [151]

6.2.66.2 ResultWrite Methods

The ResultWrite [1032] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object. )</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object. )</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetLastError</td>
<td>Sets the error state of this ResultAds (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Reference**

[ResultWrite Class](#) [1032]

[TwinCAT.Ads Namespace](#) [151]

### 6.2.67 SessionSettings Class

Session settings class

**Inheritance Hierarchy**

System.Object

TwinCAT.Ads.SessionSettings

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch/releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public class SessionSettings : IAdsSessionSettings
```

The SessionSettings type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default [1037]</td>
<td>Gets the default Settings (Synchronized).</td>
</tr>
<tr>
<td>DefaultCommunicationTimeout [1037]</td>
<td>The default communication timeout (5 Seconds)</td>
</tr>
<tr>
<td>DefaultResurrectionTime [1038]</td>
<td>The default resurrection time (21 Seconds)</td>
</tr>
<tr>
<td>FastWriteThrough [1038]</td>
<td>Gets a Settings object that configures the AdsSession for FastWriteThrough</td>
</tr>
<tr>
<td>ResurrectionTime [1039]</td>
<td>Gets or sets the resurrection time (Default: DefaultResurrectionTime [1038])</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolLoader</td>
<td>Gets or sets the symbol loader settings</td>
</tr>
<tr>
<td>Timeout</td>
<td>Gets the ADS timeout in milliseconds.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads Namespace [151]

### 6.2.67.1 SessionSettings Properties

The SessionSettings [1035] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Gets the default Settings (Synchronized).</td>
</tr>
<tr>
<td>DefaultCommunicationTimeout</td>
<td>The default communication timeout (5 Seconds)</td>
</tr>
<tr>
<td>DefaultResurrectionTime</td>
<td>The default resurrection time (21 Seconds)</td>
</tr>
<tr>
<td>FastWriteThrough</td>
<td>Gets a Settings object that configures the AdsSession for FastWriteThrough</td>
</tr>
<tr>
<td>ResurrectionTime</td>
<td>Gets or sets the resurrection time (Default: DefaultResurrectionTime)</td>
</tr>
<tr>
<td>SymbolLoader</td>
<td>Gets or sets the symbol loader settings</td>
</tr>
<tr>
<td>Timeout</td>
<td>Gets the ADS timeout in milliseconds.</td>
</tr>
</tbody>
</table>
6.2.67.1.1 SessionSettings.Default Property

Gets the default Settings (Synchronized).

**Namespace:** TwinCAT.Ads

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public static SessionSettings Default { get; }
```

**Property Value**

Type: `SessionSettings` [1035]

The default settings.

**Remarks**

The following defaults are set here:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Timeout (Timeout)</td>
<td>Default communication timeout ([DefaultCommunicationTimeout][1037], default 5s)</td>
</tr>
<tr>
<td>Resurrection Timeout (ResurrectionTime)</td>
<td>Default communication timeout ([DefaultResurrectionTime][1038], default 21s)</td>
</tr>
<tr>
<td>Dynamic SymbolLoader settings SymboLoader</td>
<td>Synchronized mode activated ([DefaultDynamic][146])</td>
</tr>
</tbody>
</table>

**Reference**

SessionSettings Class [1035]

TwinCAT.Ads Namespace [151]

6.2.67.1.2 SessionSettings.DefaultCommunicationTimeout Property

The default communication timeout (5 Seconds)

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public static TimeSpan DefaultCommunicationTimeout { get; }
```
6.2.67.1.3 SessionSettings.DefaultResurrectionTime Property

The default resurrection time (21 Seconds)

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
```csharp
public static TimeSpan DefaultResurrectionTime { get; }
```

Property Value

Type: TimeSpan

Reference

SessionSettings Class [1035]
TwinCAT.Ads Namespace [151]

6.2.67.1.4 SessionSettings.FastWriteThrough Property

Gets a Settings object that configures the AdsSession for FastWriteThrough

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
```csharp
public static SessionSettings FastWriteThrough { get; }
```

Property Value

Type: SessionSettings

Session settings for a fast write through (with 200 ms Timeout).

Remarks

The settings typically can be used for polling clients, where the "FailFast" feature will be bypassed. That means, that communication fails doesn't trigger the FailFast interceptor and every Request will go out via ADS. This has the Drawback that communication Timeouts are longer and subsequent timeouts block the ADS mailbox (with the danger of overflows). So use this setting with care for specific purposes and should not be used for standard communication.
• No Resurrection time and therefore:
• No FailFastHandler active.
• Default communication timeout 200ms.
• Not synchronized Notifications.

Reference

SessionSettings Class [1035]
TwinCAT.Ads Namespace [151]

6.2.67.1.5 SessionSettings.ResurrectionTime Property

Gets or sets the resurrection time (Default: DefaultResurrectionTime [1038])

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public TimeSpan ResurrectionTime { get; set; }

Property Value

Type: TimeSpan
The resurrection time.

Implements

IAdsSessionSettings.ResurrectionTime [913]

Remarks

The resurrection time is the time after a lost connection Lost [67] can be 'resurrected'. This time is set to 21 Seconds by default (a value greater than the standard Ethernet connection timeout of 20s). The reason for this timeout is not to flood the ADS mailbox with requests that cannot be handled by the ethernet infrastructure. As long this Timespan is not expired after a recognized Lost [67], no further data communication is done, and requests are immediately ('FailFast') answered by communication exceptions. Change this value only for edge cases.

Reference

SessionSettings Class [1035]
TwinCAT.Ads Namespace [151]

6.2.67.1.6 SessionSettings.SymbolLoader Property

Gets or sets the symbol loader settings

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#
public SymbolLoaderSettings SymbolLoader { get; set; }

Property Value

Type: SymbolLoaderSettings [140]
The symbol loader.

Implements

IAdsSessionSettings.SymbolLoader [914]

Reference

SessionSettings Class [1035]
TwinCAT.Ads Namespace [151]

6.2.67.1.7 SessionSettings.Timeout Property

Gets the ADS timeout in milliseconds.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public int Timeout { get; }

Property Value

Type: Int32
The timeout.

Implements

IAdsSessionSettings.Timeout [914]

Reference

SessionSettings Class [1035]
TwinCAT.Ads Namespace [151]

6.2.67.2 SessionSettings Methods

The SessionSettings [1035] type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

## Reference

- SessionSettings Class [1035]
- TwinCAT.Ads Namespace [151]

### 6.2.68 StatInfo Structure

The structure contains the ADS state and device state.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**C#**

```csharp
public struct StateInfo : IEquatable<StateInfo>
```

The StatInfo type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StateInfo(ReadOnlySpan)</td>
<td></td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsState</td>
<td>Gets or sets the ADS state of this StatInfo object.</td>
</tr>
<tr>
<td>DeviceState</td>
<td>Gets or sets the device state of this StatInfo object.</td>
</tr>
<tr>
<td>Empty</td>
<td>Empty / Invalid / Uninitialized state.</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(Object)</code></td>
<td>Determines whether the specified Object is equal to this instance. (Overrides <code>ValueType.Equals(Object)</code>.)</td>
</tr>
<tr>
<td><code>Equals(StateInfo)</code></td>
<td>Determines whether the specified StateInfo is equal to this instance.</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Returns a hash code for this instance. (Overrrides <code>ValueType.GetHashCode</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns the fully qualified type name of this instance. (Inherited from <code>ValueType</code>.)</td>
</tr>
</tbody>
</table>

### Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equality</code></td>
<td>Implements the <code>==</code>.</td>
</tr>
<tr>
<td><code>Inequality</code></td>
<td>Implements the <code>!=</code>.</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads Namespace [151]

#### 6.2.68.1 StateInfo Constructor

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>StateInfo(ReadOnlySpan)</code></td>
<td>Initializes a new Instance of the StateInfo struct.</td>
</tr>
<tr>
<td><code>StateInfo(AdsState, Int16)</code></td>
<td>Initializes a new Instance of the StateInfo struct.</td>
</tr>
</tbody>
</table>

### Reference

StateInfo Structure [1041]

TwinCAT.Ads Namespace [151]

#### 6.2.68.1.1 StateInfo Constructor (ReadOnlySpan`1`)

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

**C#**

```csharp
public StateInfo(
    ReadOnlySpan data
)
```

**Parameters**

- **data**  
  Type: `ReadOnlySpan`

**Reference**

- StateInfo Structure [1041]
- StateInfo Overload [1042]
- TwinCAT.Ads Namespace [151]

### 6.2.68.1.2 StateInfo Constructor (AdsState, Int16)

Initializes a new Instance of the StateInfo struct.

**Namespace:** TwinCAT.Ads [151]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)  
Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public StateInfo(
    AdsState adsState,
    short deviceState
)
```

**Parameters**

- **adsState**  
  Type: `TwinCAT.Ads.AdsState` [626]  
  Ads state.
- **deviceState**  
  Type: `System.Int16`  
  Device state.

**Reference**

- StateInfo Structure [1041]
- StateInfo Overload [1042]
- TwinCAT.Ads Namespace [151]

### 6.2.68.2 StateInfo Properties

The `StateInfo` [1041] type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsState</td>
<td>Gets or sets the ADS state of this StateInfo object.</td>
</tr>
<tr>
<td>DeviceState</td>
<td>Gets or sets the device state of this StateInfo object.</td>
</tr>
<tr>
<td>Empty</td>
<td>Empty / Invalid / Uninitialized state.</td>
</tr>
</tbody>
</table>

### Reference

StateInfo Structure [1041]

TwinCAT.Ads Namespace [151]

#### 6.2.68.2.1 StateInfo.AdsState Property

Gets or sets the ADS state of this StateInfo object.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public AdsState AdsState { get; set; }
```

**Property Value**

Type: AdsState [626]

**Reference**

StateInfo Structure [1041]

TwinCAT.Ads Namespace [151]

#### 6.2.68.2.2 StateInfo.DeviceState Property

Gets or sets the device state of this StateInfo object.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public short DeviceState { get; set; }
```

**Property Value**

Type: Int16
6.2.68.2.3 StateInfo.Empty Property

Empty / Invalid / Uninitialized state.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static StateInfo Empty { get; }
```

Property Value

Type: StateInfo
The empty.

Methods

The StateInfo type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals(Object)</td>
<td>Determines whether the specified Object is equal to this instance. (Overrides ValueType.Equals(Object).)</td>
</tr>
<tr>
<td>Equals(StateInfo)</td>
<td>Determines whether the specified StateInfo is equal to this instance.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Returns a hash code for this instance. (Overrides ValueType.GetHashCode.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns the fully qualified type name of this instance. (Inherited from ValueType.)</td>
</tr>
</tbody>
</table>

Reference

StateInfo Structure
TwinCAT.Ads Namespace
### 6.2.68.3.1 StateInfo.Equals Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals(Object)</td>
<td>Determines whether the specified Object is equal to this instance. (Overrides ValueType.Equals(Object).)</td>
</tr>
<tr>
<td>Equals(StateInfo)</td>
<td>Determines whether the specified StateInfo is equal to this instance.</td>
</tr>
</tbody>
</table>

#### Reference

StateInfo Structure [1041]

TwinCAT.Ads Namespace [151]

### StatInfo.Equals Method (Object)

Determines whether the specified Object is equal to this instance.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public override bool Equals(
    Object ob
)
```

**Parameters**

- **ob**
  
  Type: System.Object
  
  The Object to compare with this instance.

**Return Value**

Type: Boolean

true if the specified Object is equal to this instance; otherwise, false.

#### Reference

StateInfo Structure [1041]

Equals Overload [1046]

TwinCAT.Ads Namespace [151]

### StatInfo.Equals Method (StateInfo)

Determines whether the specified StateInfo is equal to this instance.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public bool Equals(
    StateInfo info
)
```

Parameters

- **info**: Type: `TwinCAT.Ads.StateInfo` [1041]
  - The `StateInfo` [1041] to compare with this instance.

Return Value

Type: `Boolean`
- true if the specified `StateInfo` [1041] is equal to this instance; otherwise, false.

Implements

`IEquatable<T>.Equals(T)`

Reference

- `StateInfo Structure` [1041]
- `Equals Overload` [1046]
- `TwinCAT.Ads Namespace` [151]

### 6.2.68.3.2 StateInfo.GetHashCode Method

Returns a hash code for this instance.

**Namespace**: `TwinCAT.Ads` [151]
**Assembly**: `TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public override int GetHashCode()
```

Return Value

Type: `Int32`
- A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

Reference

- `StateInfo Structure` [1041]
- `TwinCAT.Ads Namespace` [151]

### 6.2.68.4 StateInfo Operators

The `StateInfo` [1041] type exposes the following members.
Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equality [1048]</td>
<td>Implements the ==.</td>
</tr>
<tr>
<td>Inequality [1048]</td>
<td>Implements the !=.</td>
</tr>
</tbody>
</table>

Reference

StateInfo Structure [1041]
TwinCAT.Ads Namespace [151]

6.2.68.4.1 StateInfo.Equality Operator

Implements the ==.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static bool operator ==(StateInfo a, StateInfo b)
```

Parameters

- a of Type: TwinCAT.Ads.StateInfo [1041]
- b of Type: TwinCAT.Ads.StateInfo [1041]

Return Value

Type: Boolean
The result of the operator.

Reference

StateInfo Structure [1041]
TwinCAT.Ads Namespace [151]

6.2.68.4.2 StateInfo.Inequality Operator

Implements the !=.

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public static bool operator !=(
    StateInfo a,
    StateInfo b
)
```

**Parameters**

- `a`: Type: `TwinCAT.Ads.StateInfo [1041]`
- `b`: Type: `TwinCAT.Ads.StateInfo [1041]`

**Return Value**

Type: `Boolean`
The result of the operator.

**Reference**

- `StateInfo Structure [1041]`
- `TwinCAT.Ads Namespace [151]`

### 6.2.69 TaskExtensions Class

TaskExtensions for Task Cancellation and Timeout

**Inheritance Hierarchy**

- `System.Object`
  - `TwinCAT.Ads.TaskExtensions`

**Namespace:** `TwinCAT.Ads` [151]

**Assembly:** `TwinCAT.Ads.Abstractions` (in `TwinCAT.Ads.Abstractions.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static class TaskExtensions
```

The `TaskExtensions` type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ] WithCancellationAndTimeout [1051]</td>
<td>Extends a worker task with timeout and Cancellation</td>
</tr>
</tbody>
</table>
6.2.69.1 TaskExtensions Methods

The TaskExtensions type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WithCancellation.T.</td>
<td>Extends a worker Task with cancellation.</td>
</tr>
<tr>
<td>WithCancellationAndTimeout</td>
<td>Extends a worker task with timeout and Cancellation</td>
</tr>
<tr>
<td>WithTimeout</td>
<td>Extends a worker task with a timeout.</td>
</tr>
</tbody>
</table>

### Reference

TaskExtensions Class [1049]

TwinCAT.Ads Namespace [151]

#### 6.2.69.1.1 TaskExtensions.WithCancellation.T. Method

Extends a worker Task with cancellation.

**Namespace:** TwinCAT.Ads [151]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public static Task<T> WithCancellation<T>(
    this Task<T> worker,
    CancellationToken cancellationToken
)
```

**Parameters**

- **worker**
  - Type: System.Threading.Tasks.Task<T>
  - The worker task.

- **cancellationToken**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

**Type Parameters**

- **T**
  - The Task Result
Return Value

Type: Task<T>. Task<AdsErrorCode>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type Task<T>. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>OperationCanceledException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

TaskExtensions Class [1049]
TwinCAT.Ads Namespace [151]

6.2.69.1.2 TaskExtensions.WithCancellationAndTimeout Method

Extends a worker task with timeout and Cancellation

Namespace: TwinCAT.Ads [151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static Task<AdsErrorCode> WithCancellationAndTimeout(
    this Task<AdsErrorCode> worker,
    TimeSpan timeout,
    CancellationToken cancellationToken)
```

Parameters


Return Value

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type `Task<AdsErrorCode>`. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>OperationCanceledException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

TaskExtensions Class [P. 1049]

TwinCAT.Ads Namespace [P. 151]

6.2.69.1.3 TaskExtensions.WithTimeout Method

Extends a worker task with a timeout.

Namespace: TwinCAT.Ads [P. 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static Task<AdsErrorCode> WithTimeout(
    this Task<AdsErrorCode> worker,
    TimeSpan timeout)
```

Parameters

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>timeout</td>
<td>Type: System.TimeSpan. The timeout.</td>
</tr>
</tbody>
</table>

Return Value


Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type `Task<AdsErrorCode>`. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

TaskExtensions Class [P. 1049]
6.2.70 TransportProtocols Enumeration

Enum ADS TransportProtocol

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
[FlagsAttribute]
public enum TransportProtocols
```

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>None / Uninitialized</td>
</tr>
<tr>
<td>Router</td>
<td>1</td>
<td>ADS via Router</td>
</tr>
<tr>
<td>Tcplp</td>
<td>2</td>
<td>ADS via TCP/IP (without router)</td>
</tr>
<tr>
<td>All</td>
<td>3</td>
<td>Indicates that Router and Tcplp are appropriate (for establishing connections)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads Namespace

6.2.71 ValueNotificationEventArgs<T>. Class

Arguments for AdsNotificationEx events.

Inheritance Hierarchy

System.Object
    System.EventArgs
        TwinCAT.Ads.AdsNotificationEventArgs
            TwinCAT.Ads.ValueNotificationEventArgs<T>

Namespace: TwinCAT.Ads

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class ValueNotificationEventArgs<T> : AdsNotificationEventArgs
```

Type Parameters

T

The ValueNotificationEventArgs<T> type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data [598]</td>
<td>Memory object holding the Notification Data/Value. (Inherited from AdsNotificationEventArgs [597].)</td>
</tr>
<tr>
<td>Handle [599]</td>
<td>Gets the Notification handle. (Inherited from AdsNotificationEventArgs [597].)</td>
</tr>
<tr>
<td>TimeStamp [600]</td>
<td>Gets the time stamp of this Notification as DateTimeOffset. (Inherited from AdsNotificationEventArgs [597].)</td>
</tr>
<tr>
<td>UserData [600]</td>
<td>Gets the user object. This object is passed by to AddDeviceNotification and can be used to store data. (Inherited from AdsNotificationEventArgs [597].)</td>
</tr>
<tr>
<td>Value [1055]</td>
<td>Value of the ADS Notification.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads Namespace [151]

#### 6.2.71.1 ValueNotificationEventArgs.T. Properties

The ValueNotificationEventArgs.T. [1053] generic type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data [598]</td>
<td>Memory object holding the Notification Data/Value. (Inherited from AdsNotificationEventArgs [597].)</td>
</tr>
<tr>
<td>Handle [599]</td>
<td>Gets the Notification handle. (Inherited from AdsNotificationEventArgs [597].)</td>
</tr>
<tr>
<td>TimeStamp [600]</td>
<td>Gets the time stamp of this Notification as DateTimeOffset. (Inherited from AdsNotificationEventArgs [597].)</td>
</tr>
<tr>
<td>UserData [600]</td>
<td>Gets the user object. This object is passed by to AddDeviceNotification and can be used to store data. (Inherited from AdsNotificationEventArgs [597].)</td>
</tr>
<tr>
<td>Value [1055]</td>
<td>Value of the ADS Notification.</td>
</tr>
</tbody>
</table>
6.2.71.1.1 ValueNotificationEventArgs.T.Value Property

Value of the ADS Notification.

Namespace: TwinCAT.Ads
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public T Value { get; }
```

Property Value

Type: T

Reference

ValueNotificationEventArgs.T.Class
TwinCAT.Ads Namespace

6.2.71.2 ValueNotificationEventArgs.T.Methods

The ValueNotificationEventArgs.T.[1053] generic type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

ValueNotificationEventArgs.T.Class
TwinCAT.Ads Namespace

TC1000 Version: 1.1 1055
6.3 TwinCAT.Ads.Reactive Namespace

Reactive Extensions for the ADS Client. All types within are contained in the ADS companion package "Beckhoff.TwinCAT.Ads.Reactive" which must be referenced separately. https://www.nuget.org/packages/Beckhoff.TwinCAT.Ads.Reactive/

Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsClientExtensions</td>
<td>Extension class for AdsClient respective IAdsConnection to provide reactive ADS extensions.</td>
</tr>
<tr>
<td>AnyTypeExtensions</td>
<td>Extension class for IAdsConnection to provide reactive ADS extensions (accessing symbol value sequences with the ANY_TYPE concept)</td>
</tr>
<tr>
<td>SymbolValueNotification</td>
<td>Symbol Notification class</td>
</tr>
<tr>
<td>ValueSymbolExtensions</td>
<td>Extension class for IAdsConnection to provide reactive ADS extensions for accessing symbols that are loaded by the IAdsSymbolLoaderFactory</td>
</tr>
</tbody>
</table>

6.3.1 AdsClientExtensions Class

Extension class for AdsClient respective IAdsConnection to provide reactive ADS extensions.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.Reactive.AdsClientExtensions

Namespace: TwinCAT.Ads.Reactive [1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static class AdsClientExtensions
```

The AdsClientExtensions type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollAdsState(IAdsConnection, IObservable&lt;Unit&gt;)</td>
<td>Gets an observable sequence of AdsState via Polling.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollAdsState(IAdsConnection, TimeSpan)</td>
<td>Gets an observable sequence of AdsState}s via Polling.</td>
</tr>
<tr>
<td>PollAdsStateAsync(IAdsConnection, IObservable&lt;Unit&gt;, CancellationToken)</td>
<td>Gets an observable sequence of AdsState}s via Polling.</td>
</tr>
<tr>
<td>PollAdsStateAsync(IAdsConnection,(TimeSpan, CancellationToken)</td>
<td>Gets an observable sequence of AdsState}s via Polling.</td>
</tr>
<tr>
<td>WhenAdsStateChanged</td>
<td>Gets an observable sequence of AdsState}s.</td>
</tr>
<tr>
<td>WhenNotification(IAdsConnection, ISymbol)</td>
<td>Gets an observable sequence of Notification}s.</td>
</tr>
<tr>
<td>WhenNotification(IAdsConnection, ISymbolCollection)</td>
<td>Gets an observable sequence of Notification} objects.</td>
</tr>
<tr>
<td>WhenNotification(IAdsConnection, ISymbol, NotificationSettings)</td>
<td>Gets an observable sequence of SymbolValueChangedNotification} objects.</td>
</tr>
<tr>
<td>WhenNotification(IAdsConnection, ISymbolCollection, NotificationSettings)</td>
<td>Gets an observable sequence of Notification} objects.</td>
</tr>
<tr>
<td>WhenSymbolVersionChanges(IAdsConnection)</td>
<td>Gets an observable sequence of SymbolVersion changed counts.</td>
</tr>
<tr>
<td>WhenSymbolVersionChanges(IAdsConnection, IScheduler)</td>
<td>Gets an observable sequence of SymbolVersion changed counts.</td>
</tr>
</tbody>
</table>
TwinCAT.Ads Namespaces
Remarks
Reactive Extensions (Rx) are a library for composing asynchronous and event-based programs using
observable sequences and LINQ-style query operators. Using Rx, developers represent asynchronous data
streams with Observables, query asynchronous data streams using LINQ operators, and parameterize the
concurrency in the asynchronous data streams using Schedulers. Simply put, Rx = Observables + LINQ +
Schedulers. The ADS reactive extensions are build on top of this library to enable ADS Symbol and State
Observables, seamlessly bound to the reactive extensions. To use the ADS reactive extensions the
TwinCAT.Ads.Reactive Nuget package (or the included TwinCAT.Ads.Reactive.dll) must be referenced.
(Beckhoff.TwinCAT.Ads.Reactive package on Nuget).
Examples
The following sample shows how observe Value changed Notifications with the reactive AdsClientExtensions
Observe changing ADS Symbols with reactive extensions.
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
// Connect to target
client.Connect(new AmsAddress(AmsNetId.Local, 851));
// Create Symbol information
var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
int eventCount = 1;
// Reactive Notification Handler
var valueObserver = Observer.Create<SymbolValueNotification>(not =>
{
Console.WriteLine(string.Format("{0} {1:u} {2} = '{3}' ({4})", eventCount+
+, not.TimeStamp, not.Symbol.InstancePath, not.Value, not.Symbol.DataType));
}
);
// Collect the symbols that are registered as Notification sources for their changed values.
SymbolCollection notificationSymbols = new SymbolCollection();
IArrayInstance taskInfo = (IArrayInstance)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskI
nfo"];
foreach(ISymbol element in taskInfo.Elements)
{
ISymbol cycleCount = element.SubSymbols["CycleCount"];
ISymbol lastExecTime = element.SubSymbols["LastExecTime"];
notificationSymbols.Add(cycleCount);
notificationSymbols.Add(lastExecTime);
}
// Create a subscription for the first 200 Notifications on Symbol Value changes.
IDisposable subscription = client.WhenNotification(notificationSymbols,NotificationSettings.Defa
ult).Take(200).Subscribe(valueObserver);
Console.ReadKey(); // Wait for Key press
subscription.Dispose(); // Dispose the Subscription
}

Examples
The following sample shows how observe AdsState [} 626] changed Notifications with the reactive
AdsClientExtensions
Observe changing ADS states with reactive extensions.
// To Test the observer, Start/Stop the local PLC
using (AdsClient client = new AdsClient())
{
// Connect to target
client.Connect(new AmsAddress(AmsNetId.Local, 851));

1058

Version: 1.1

TC1000


// Create Symbol information
var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);

// Reactive Notification Handler
var valueObserver = Observer.Create<IList<AdsState>>(not =>
{
    AdsState oldValue = not[0];
    AdsState newValue = not[1];

    Console.WriteLine(string.Format("Changed ADSState from '{0}' --
> '{1}'!", oldValue, newValue));
});

// Create a subscription for the AdsState change and buffering 2 Values (for oldValue --
> newValue output).
IDisposable subscription = client.WhenAdsStateChanges().Buffer(2,1).Subscribe(valueObserver);

Console.ReadKey(); // Wait for Key press
subscription.Dispose(); // Dispose the Subscription

Reference

TwinCAT.Ads.Reactive Namespace [1056]
TwinCAT.Ads.Reactive.AnyTypeExtensions [1075]
TwinCAT.Ads.Reactive.ValueSymbolExtensions [1106]

6.3.1.1 AdsClientExtensions Methods

The AdsClientExtensions [1056] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WhenAdsStateChanges</td>
<td>Gets an observable sequence of AdsState[626]s.</td>
</tr>
<tr>
<td>WhenNotification(IAdsConnection, ISymbol)</td>
<td>Gets an observable sequence of Notification[974]s.</td>
</tr>
<tr>
<td>WhenNotification(IAdsConnection, ISymbolCollection)</td>
<td>Gets an observable sequence of Notification[974] objects.</td>
</tr>
<tr>
<td>WhenNotification(IAdsConnection, ISymbol, NotificationSettings)</td>
<td>Gets an observable sequence of SymbolValueNotification[1104]s.</td>
</tr>
<tr>
<td>WhenSymbolVersionChanges(IAdsConnection)</td>
<td>Gets an observable sequence of SymbolVersion changed counts.</td>
</tr>
<tr>
<td>WhenSymbolVersionChanges(IAdsConnection, IScheduler)</td>
<td>Gets an observable sequence of SymbolVersion changed counts.</td>
</tr>
</tbody>
</table>

**Reference**

AdsClientExtensions Class [1056]

TwinCAT.Ads.Reactive Namespace [1056]

### 6.3.1.1.1 AdsClientExtensions.PollAdsState Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollAdsState(IAdsConnection, TimeSpan)</td>
<td>Gets an observable sequence of AdsState}s via Polling.</td>
</tr>
</tbody>
</table>

**Reference**

AdsClientExtensions Class [1056]

TwinCAT.Ads.Reactive Namespace [1056]

**AdsClientExtensions.PollAdsState Method (IAdsConnection, IObservable.Unit.)**

Gets an observable sequence of AdsState}s via Polling.

**Namespace:** TwinCAT.Ads.Reactive [1056]
**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static IObservable<AdsState> PollAdsState(
    this IAdsConnection client,
    IObservable<Unit> trigger
)
```

**Parameters**

- **client**
  - Type: TwinCAT.Ads.IAdsConnection [765]
  - The client.

- **trigger**
  - Type: System.IObservable.Unit
  - The polling trigger

**Return Value**

Type: IObservable<AdsState> [626].

**Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

**Examples**

The following sample shows how observe AdsState}s via polling with the reactive AdsClientExtensions [1056].

```
Observe changing ADS States with reactive extensions.

// To Test the observer, Start/Stop the local PLC
using (AdsClient client = new AdsClient())
{
    // Connect to target
```
```csharp
client.Connect(new AmsAddress(AmsNetId.Local, 851));

// Create Symbol information
var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);

// Reactive Change Handler
var valueObserver = Observer.Create<IList<AdsState>>(not =>
{
    AdsState oldValue = not[0];
    AdsState newValue = not[1];

    Console.WriteLine(string.Format("Changed ADSState from '{0}' --> '{1}!'", oldValue, newValue));
});

// Create a subscription for the AdsState change and buffering 2 Values (for oldValue --> newValue output).
IDisposable subscription = client.PollAdsState(TimeSpan.FromMilliseconds(200)).Buffer(2, 1).Subscribe(valueObserver);

Console.ReadKey(); // Wait for Key press
subscription.Dispose(); // Dispose the Subscription
```

**Reference**

AdsClientExtensions Class [› 1056]

PollAdsState Overload [› 1060]

TwinCAT.Ads.Reactive Namespace [› 1056]

AdsClientExtensions.WhenAdsStateChanges(IAdsConnection) [› 1066]

---

### AdsClientExtensions.PollAdsState Method (IAdsConnection, TimeSpan)

 Gets an observable sequence of AdsState via Polling.

**Namespace:** TwinCAT.Ads.Reactive [› 1056]

**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public static IObservable<AdsState> PollAdsState(
    this IAdsConnection client,
    TimeSpan period
)
```

**Parameters**

- **client**
  
  Type: TwinCAT.Ads.IAdsConnection [› 765]

  The client.

- **period**

  Type: System.TimeSpan

  The period.

**Return Value**

Type: IObservable<AdsState> [› 626].

IObservable<AdsState>. 
Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

The following sample shows how observe AdsState via polling with the reactive AdsClientExtensions.

Observe changing ADS States with reactive extensions.

```csharp
// To Test the observer, Start/Stop the local PLC
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);

    // Reactive Change Handler
    var valueObserver = Observer.Create<IList<AdsState>>(not =>
    {
        AdsState oldValue = not[0];
        AdsState newValue = not[1];

        Console.WriteLine(string.Format("Changed ADSState from '{0}' --> '{1}'!", oldValue, newValue));
    });

    // Create a subscription for the AdsState change and buffering 2 Values (for oldValue --> newValue output).
    IDisposable subscription = client.PollAdsState(TimeSpan.FromMilliseconds(200)).Buffer(2, 1).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press

    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

AdsClientExtensions Class

PollAdsState Overload

TwinCAT.Ads.Reactive Namespace

AdsClientExtensions.WhenAdsStateChanges(IAdsConnection)

6.3.1.1.2 AdsClientExtensions.PollAdsStateAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollAdsStateAsync(IAdsConnection, IObservable&lt;Unit&gt;, CancellationToken)</td>
<td>Gets an observable sequence of AdsState via Polling.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="PollAdsStateAsync(IAdsConnection, TimeSpan, CancellationToken)" /></td>
<td>Gets an observable sequence of AdsState via Polling.</td>
</tr>
</tbody>
</table>

**Reference**

- AdsClientExtensions Class [1056]
- TwinCAT.Ads Reactive Namespace [1056]

**AdsClientExtensions.PollAdsStateAsync Method (IAdsConnection, IObservable.Unit., CancellationToken)**

Gets an observable sequence of AdsState via Polling.

- **Namespace:** TwinCAT.Ads.Reactive [1056]
- **Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public static IObservable<AdsState> PollAdsStateAsync(
    this IAdsConnection client,
    IObservable<Unit> trigger,
    CancellationToken cancel
)
```

**Parameters**

- **client**
  - Type: TwinCAT.Ads.IAdsConnection [765]
  - The client.

- **trigger**
  - Type: System.IObservable.Unit.
  - The polling trigger

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

**Return Value**

Type: IObservable<AdsState> [626].

- IObservable<AdsState>.

**Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

**Examples**

The following sample shows how observe AdsState via polling with the reactive AdsClientExtensions.
Observe changing ADS States with reactive extensions.

```csharp
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);

    // Reactive Change Handler
    var valueObserver = Observer.Create<IList<AdsState>>(not =>
    {
        AdsState oldValue = not[0];
        AdsState newValue = not[1];

        Console.WriteLine(string.Format("Changed ADSState from '{0}' --> '{1}'!", oldValue, newValue));
    });

    // Create a subscription for the AdsState change and buffering 2 Values (for oldValue --> newValue output).
    IDisposable subscription = client.PollAdsState(TimeSpan.FromMilliseconds(200)).Buffer(2, 1).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

**Reference**

- **AdsClientExtensions Class** [1056]
- **PollAdsStateAsync Overload** [1063]
- **TwinCAT.Ads.Reactive Namespace** [1056]
- **AdsClientExtensions.WhenAdsStateChanges(IAdsConnection)** [1066]

**AdsClientExtensions.PollAdsStateAsync Method (IAdsConnection, TimeSpan, CancellationToken)**

Gets an observable sequence of `AdsState` s via Polling.

**Namespace:** TwinCAT.Ads.Reactive [1056]

**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static IObservable<AdsState> PollAdsStateAsync(
    this IAdsConnection client,
    TimeSpan interval,
    CancellationToken cancellationToken
)
```

**Parameters**

- **client**
  - Type: `TwinCAT.Ads.IAdsConnection` [765]
  - The client.

- **interval**
  - Type: `System.TimeSpan`
  - The interval.
cancel

Return Value

Type: System.Threading.CancellationToken
The cancellation token.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type AdsConnection. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

The following sample shows how observe AdsState via polling with the reactive AdsClientExtensions.

```csharp
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);

    // Reactive Change Handler
    var valueObserver = Observer.Create<IList<AdsState>>(not =>
    {
        AdsState oldValue = not[0];
        AdsState newValue = not[1];

        Console.WriteLine(string.Format("Changed ADSState from '{0}' --> '{1}'!", oldValue, newValue));
    });

    // Create a subscription for the AdsState change and buffering 2 Values (for oldValue -- > newValue output).
    IDisposable subscription = client.PollAdsState(TimeSpan.FromMilliseconds(200)).Buffer(2, 1).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

AdsClientExtensions Class [1056]
PollAdsStateAsync Overload [1063]
TwinCAT.Ads.Reactive Namespace [1056]
AdsClientExtensions.WhenAdsStateChanges(IAdsConnection) [1066]

6.3.1.1.3 AdsClientExtensions.WhenAdsStateChanges Method

Gets an observable sequence of AdsState s.
Namespace: TwinCAT.Ads.Reactive [1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static IObservable<AdsState> WhenAdsStateChanges(
    this IAdsConnection client
)
```

Parameters

client Type: TwinCAT.Ads.IAdsConnection [765]
The client.

Return Value

Type: IObservable<AdsState [626]>
IObservable<AdsState>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

The following sample shows how observe AdsState [626] changed Notifications with the reactive AdsClientExtensions [1056]

```
// To Test the observer, Start/Stop the local PLC
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);

    // Reactive Notification Handler
    var valueObserver = Observer.Create<IList<AdsState>>(not =>
    {
        AdsState oldValue = not[0];
        AdsState newValue = not[1];

        Console.WriteLine(string.Format("Changed ADSState from '{0}' -- > '{1}!", oldValue, newValue));
    });

    // Create a subscription for the AdsState change and buffering 2 Values (for oldValue --> newValue output).
    IDisposable subscription = client.WhenAdsStateChanges().Buffer(2, 1).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

AdsClientExtensions Class [1056]
## AdsClientExtensions.WhenNotification Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Notify" /></td>
<td>Gets an observable sequence of <code>Notification</code> objects.</td>
</tr>
<tr>
<td><img src="image.png" alt="Notify" /></td>
<td>Gets an observable sequence of <code>Notification</code> objects.</td>
</tr>
<tr>
<td><img src="image.png" alt="Notify" /></td>
<td>Gets an observable sequence of <code>SymbolValueNotification</code> objects.</td>
</tr>
<tr>
<td><img src="image.png" alt="Notify" /></td>
<td>Gets an observable sequence of <code>Notification</code> objects.</td>
</tr>
</tbody>
</table>

### Reference

- AdsClientExtensions Class [1056]
- TwinCAT.Ads.Reactive Namespace [1056]

### AdsClientExtensions.WhenNotification Method (IAdsConnection, ISymbol)

Gets an observable sequence of `Notification` objects.

**Namespace:** TwinCAT.Ads.Reactive [1056]

**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

```csharp
public static IObservable<Notification> WhenNotification(IAdsConnection client, ISymbol symbol)
```
Parameters

client
Type: TwinCAT.Ads.IAdsConnection
The client.

symbol
Type: TwinCAT.TypeSystem.ISymbol
The symbol.

Return Value

Type: IObservable.Notification
IObservable<NotificationValue>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AdsClientExtensions Class
WhenNotification Overload
TwinCAT.Ads.Reactive Namespace
AdsClientExtensions.WhenNotification(IAdsConnection, ISymbolCollection, NotificationSettings)
AdsClientExtensions.WhenNotification(IAdsConnection, ISymbol, NotificationSettings)

AdsClientExtensions.WhenNotification Method (IAdsConnection, ISymbolCollection)

Gets an observable sequence of Notification objects.

Namespace: TwinCAT.Ads.Reactive
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static IObservable<SymbolValueNotification> WhenNotification(
    this IAdsConnection connection,
    ISymbolCollection symbols
)

Parameters

connection
Type: TwinCAT.Ads.IAdsConnection
The client.

symbols
Type: TwinCAT.TypeSystem.ISymbolCollection
The symbols.

Return Value

Type: IObservable.SymbolValueNotification
IObservable<NotificationValue>.
TwinCAT.Ads Namespaces
Usage Note
In Visual Basic and C#, you can call this method as an instance method on any object of type
IAdsConnection [} 765]. When you use instance method syntax to call this method, omit the first parameter.
For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).
Examples
The following sample shows how observe Value changed Notifications with the reactive AdsClientExtensions
[} 1056]
Observe changing ADS Symbols with reactive extensions.
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
// Connect to target
client.Connect(new AmsAddress(AmsNetId.Local, 851));
// Create Symbol information
var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
int eventCount = 1;
// Reactive Notification Handler
var valueObserver = Observer.Create<SymbolValueNotification>(not =>
{
Console.WriteLine(string.Format("{0} {1:u} {2} = '{3}' ({4})", eventCount+
+, not.TimeStamp, not.Symbol.InstancePath, not.Value, not.Symbol.DataType));
}
);
// Collect the symbols that are registered as Notification sources for their changed values.
SymbolCollection notificationSymbols = new SymbolCollection();
IArrayInstance taskInfo = (IArrayInstance)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskI
nfo"];
foreach(ISymbol element in taskInfo.Elements)
{
ISymbol cycleCount = element.SubSymbols["CycleCount"];
ISymbol lastExecTime = element.SubSymbols["LastExecTime"];
notificationSymbols.Add(cycleCount);
notificationSymbols.Add(lastExecTime);
}
// Create a subscription for the first 200 Notifications on Symbol Value changes.
IDisposable subscription = client.WhenNotification(notificationSymbols,NotificationSettings.Defa
ult).Take(200).Subscribe(valueObserver);
Console.ReadKey(); // Wait for Key press
subscription.Dispose(); // Dispose the Subscription
}

Reference
AdsClientExtensions Class [} 1056]
WhenNotification Overload [} 1068]
TwinCAT.Ads.Reactive Namespace [} 1056]
AdsClientExtensions.WhenNotification(IAdsConnection, ISymbol, NotificationSettings) [} 1070]

AdsClientExtensions.WhenNotification Method (IAdsConnection, ISymbol,
NotificationSettings)
Gets an observable sequence of SymbolValueNotification [} 1104]s.
1070

Version: 1.1

TC1000


**Namespace:** TwinCAT.Ads.Reactive
**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll)  Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```
public static IObserveable<SymbolValueNotification> WhenNotification(
    this IAdsConnection client,
    ISymbol symbol,
    NotificationSettings settings
)
```

**Parameters**

- **client**
  
  Type: TwinCAT.Ads.IAdsConnection
  The client.

- **symbol**
  
  Type: TwinCAT.TypeSystem.ISymbol
  The symbol to observe.

- **settings**
  
  Type: TwinCAT.Ads.NotificationSettings
  Notification settings.

**Return Value**

Type: IObserveable<SymbolValueNotification>

**Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>symbol</td>
</tr>
<tr>
<td>ArgumentOutOfRangeException</td>
<td>Symbol is not an IValueSymbol - symbol</td>
</tr>
</tbody>
</table>

**Reference**

- AdsClientExtensions Class

- WhenNotification Overload

- TwinCAT.Ads.Reactive Namespace

- AdsClientExtensions.WhenNotification(IAdsConnection, ISymbolCollection, NotificationSettings)

**AdsClientExtensions.WhenNotification Method (IAdsConnection, ISymbolCollection, NotificationSettings)**

Gets an observable sequence of Notification objects.
TwinCAT.Ads Namespaces
Namespace: TwinCAT.Ads.Reactive [} 1056]
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax
C#
public static IObservable<SymbolValueNotification> WhenNotification(
this IAdsConnection client,
ISymbolCollection symbols,
NotificationSettings settings
)

Parameters
client

Type: TwinCAT.Ads.IAdsConnection [} 765]
The client.

symbols

Type: TwinCAT.TypeSystem.ISymbolCollection [} 2182]
The symbols to observe.

settings

Type: TwinCAT.Ads.NotificationSettings [} 979]
The Notification settings.

Return Value
Type: IObservable.SymbolValueNotification [} 1104].
IObservable<NotificationValue>.
Usage Note
In Visual Basic and C#, you can call this method as an instance method on any object of type
IAdsConnection [} 765]. When you use instance method syntax to call this method, omit the first parameter.
For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).
Examples
The following sample shows how observe Value changed Notifications with the reactive AdsClientExtensions
[} 1056]
Observe changing ADS Symbols with reactive extensions.
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
// Connect to target
client.Connect(new AmsAddress(AmsNetId.Local, 851));
// Create Symbol information
var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
int eventCount = 1;
// Reactive Notification Handler
var valueObserver = Observer.Create<SymbolValueNotification>(not =>
{
Console.WriteLine(string.Format("{0} {1:u} {2} = '{3}' ({4})", eventCount+
+, not.TimeStamp, not.Symbol.InstancePath, not.Value, not.Symbol.DataType));
}
);
// Collect the symbols that are registered as Notification sources for their changed values.
SymbolCollection notificationSymbols = new SymbolCollection();
IArrayInstance taskInfo = (IArrayInstance)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskI
nfo"];

1072

Version: 1.1

TC1000


foreach(ISymbol element in taskInfo.Elements)
{
    ISymbol cycleCount = element.SubSymbols["CycleCount"];
    ISymbol lastExecTime = element.SubSymbols["LastExecTime"];

    notificationSymbols.Add(cycleCount);
    notificationSymbols.Add(lastExecTime);
}

// Create a subscription for the first 200 Notifications on Symbol Value changes.
IDisposable subscription = client.WhenNotification(notificationSymbols,NotificationSettings.Default).Take(200).Subscribe(valueObserver);

Console.ReadKey(); // Wait for Key press
subscription.Dispose(); // Dispose the Subscription

Reference

AdsClientExtensions Class [1056]
WhenNotification Overload [1068]
TwinCAT.Ads.Reactive Namespace [1056]

6.3.1.1.5 AdsClientExtensions.WhenSymbolVersionChanges Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets an observable sequence of SymbolVersion changed counts.</td>
</tr>
</tbody>
</table>

Reference

AdsClientExtensions Class [1056]
TwinCAT.Ads.Reactive Namespace [1056]

AdsClientExtensions.WhenSymbolVersionChanges Method (IAdsConnection)

Gets an observable sequence of SymbolVersion changed counts.

Namespace: TwinCAT.Ads.Reactive [1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static IObservable<int> WhenSymbolVersionChanges(
    this IAdsConnection connection
)
```
AdsClientExtensions.WhenSymbolVersionChanges Method (IAdsConnection, IScheduler)

Gets an observable sequence of SymbolVersion changed counts.

**Namespace:** TwinCAT.Ads.Reactive

**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public static IObservable<int> WhenSymbolVersionChanges(IAdsConnection connection, IScheduler scheduler)
```

**Parameters**

- **connection**  
  Type: **TwinCAT.Ads.IAdsConnection**  
  The client.

- **scheduler**  
  Type: **IScheduler**  
  The scheduler.

**Return Value**

Type: **IObservable.Int32**  
Counter, unique only within the WhenSymbolVersionChanges(IAdsConnection, IScheduler) observable.

**Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type **IAdsConnection**. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).
6.3.2 AnyTypeExtensions Class

Extension class for IAdsConnection to provide reactive ADS extensions (accessing symbol value sequences with the ANY_TYPE concept)

Inheritance Hierarchy

System.Object
    TwinCAT.Ads.Reactive.AnyTypeExtensions

Namespace: TwinCAT.Ads.Reactive
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static class AnyTypeExtensions

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollValues(IAdsConnection, String, Type, IObserver.Unit.)</td>
<td>Polls the symbol values on timepoints where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>PollValues(IAdsConnection, String, Type, TimeSpan)</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues(IAdsConnection, String, Type, Int32, TimeSpan)</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues(IAdsConnection, String, Type, IObserver.Unit, Func.Exception, Object.)</td>
<td>Polls the symbol values on timepoints where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollValues(IAdsConnection, String, Type, TimeSpan, Func.Exception, Object)</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues(IAdsConnection, String, Type, Int32., IObserverable.Unit., Func.Exception, Object)</td>
<td>Polls the symbol values on time points where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>PollValues(IAdsConnection, String, Type, Int32., TimeSpan, Func.Exception, Object)</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues.T.(IAdsConnection, String, IObserverable.Unit.)</td>
<td>Polls the symbol values on timepoints where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>PollValues.T.(IAdsConnection, String, TimeSpan)</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues.T.(IAdsConnection, String, Int32., IObserverable.Unit.)</td>
<td>Polls the symbol values on time points where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>PollValues.T.(IAdsConnection, String, Int32., TimeSpan)</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues.T.(IAdsConnection, String, IObserverable.Unit., Func.Exception, T.)</td>
<td>Polls the symbol values on timepoints where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>PollValues.T.(IAdsConnection, String, TimeSpan, Func.Exception, T.)</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollValues.T.</td>
<td>Polls the symbol values on timepoints where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>PollValues.T.</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>WhenNotification</td>
<td>Creates an observable sequence of values that are created by ADS Notifications.</td>
</tr>
<tr>
<td>WhenNotification.T.</td>
<td>Creates an observable sequence of values that are created by ADS Notifications.</td>
</tr>
<tr>
<td>WriteValues.T.</td>
<td>Writes the sequence of values to the symbol specified by the instance path.</td>
</tr>
<tr>
<td>WriteValues.T.</td>
<td>Writes the sequence of values to the symbol specified by the instance path.</td>
</tr>
</tbody>
</table>

**Remarks**

Reactive Extensions (Rx) are a library for composing asynchronous and event-based programs using observable sequences and LINQ-style query operators. Using Rx, developers represent asynchronous data streams with Observables, query asynchronous data streams using LINQ operators, and parameterize the concurrency in the asynchronous data streams using Schedulers. Simply put, Rx = Observables + LINQ + Schedulers. The ADS reactive extensions are build on top of this library to enable ADS Symbol and State Observables, seamlessly bound to the reactive extensions. To use the ADS reactive extensions the TwinCAT.Ads.Reactive Nuget package (or the included TwinCAT.Ads.Reactive.dll) must be referenced. (Beckhoff.TwinCAT.Ads.Reactive package on Nuget).

**Examples**

Example1: Observe Value changed Notifications with the reactive AnyTypeExtensions

**Observe a single changing ADS Symbols (Extended AdsNotifications, ANY_TYPE)**

```csharp
// To Test the Observer run a project on the local PLC System (Port 851) using (AdsClient client = new AdsClient())
{...
```
// Connect to target
client.Connect(new AmsAddress(AmsNetId.Local, 851));

// Reactive Notification Handler
var valueObserver = Observer.Create<ushort>(val => {
    Console.WriteLine(string.Format("Value: {0}", val.ToString()));
});

// Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
// and subscribe to them.
IDisposable subscription = client.WhenNotification<ushort>("TwinCAT_SystemInfoVarList._TaskInfo.
CycleCount", NotificationSettings.Default).Take(20).Subscribe(valueObserver);

Console.ReadKey(); // Wait for Key press
subscription.Dispose(); // Dispose the Subscription

Examples

Example2: Polling ANY_TYPE values.

Observe changing ADS Symbols by polling (Read Polling) (ANY_TYPE)

// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInfo.
CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val => {
        Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToString()));
    });

    // Take 20 Values in an Interval of 500ms
    IDisposable subscription = cycleCount.PollValues(TimeSpan.FromMilliseconds(500)).Take(20).Subscr
    ipe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}

Examples

Write values sequentially.

Write sequences of values to the target (ANY_TYPE)

using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information (Symbol 'i : INT' in PLC Global Variables list.
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol gvlIntSymbol = (IValueSymbol)symbolLoader.Symbols["GVL.i"];

    // Produces object (short) Values 0,1,2,3 ... in seconds period
    IObservable<object> timerObservable = Observable.Interval(TimeSpan.FromSeconds(1.0)).Select(i => (object)(short)i);

    // Take 10 Values (0..9) and write them to GVL.i
    IDisposable dispose = gvlIntSymbol.WriteValues(timerObservable.Take(10));
```csharp
Console.ReadKey(); // Wait for Key press
dispose.Dispose(); // Dispose the Subscription
```
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollValues(IAdsConnection, String, Type, Int32, TimeSpan, Func.Exception, Object.) [1097]</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues.T(IAdsConnection, String, IObservable.Unit.) [1083]</td>
<td>Polls the symbol values on timepoints where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>PollValues.T(IAdsConnection, String, TimeSpan) [1084]</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues.T(IAdsConnection, String, Int32, IObservable.Unit.) [1085]</td>
<td>Polls the symbol values on time points where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>PollValues.T(IAdsConnection, String, Int32, TimeSpan) [1086]</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues.T(IAdsConnection, String, IObservable.Unit, Func.Exception, T.) [1087]</td>
<td>Polls the symbol values on timepoints where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>PollValues.T(IAdsConnection, String, TimeSpan, Func.Exception, T.) [1088]</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues.T(IAdsConnection, String, Int32, IObservable.Unit, Func.Exception, T.) [1091]</td>
<td>Polls the symbol values on timepoints where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>PollValues.T(IAdsConnection, String, Int32, TimeSpan, Func.Exception, T.) [1092]</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### TwinCAT.Ads.Reactive Namespace

6.3.2.1.1 **AnyTypeExtensions.PollValues Method**

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PollValues(T.(IAdsConnection, String, IObservable.Unit))</code></td>
<td>Polls the symbol values on timepoints where the polling observable streams data / triggers</td>
</tr>
<tr>
<td><code>PollValues(T.(IAdsConnection, String, TimeSpan))</code></td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td><code>PollValues(T.(IAdsConnection, String, Int32, IObservable.Unit))</code></td>
<td>Polls the symbol values on time points where the polling observable streams data / triggers</td>
</tr>
</tbody>
</table>

---

**Reference**

AnyTypeExtensions Class [1075]

TwinCAT.Ads.Reactive Namespace [1056]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollValues.T. (IAdsConnection, String, .Int32., TimeSpan)</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues.T. (IAdsConnection, String, IObservable.Unit., Func.Exception, T.)</td>
<td>Polls the symbol values on timepoints where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>PollValues.T. (IAdsConnection, String, TimeSpan, Func.Exception, T.)</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues(IAdsConnection, String, Type, IObservable.Unit.)</td>
<td>Polls the symbol values on timepoints where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>PollValues((IAdsConnection, String, Type, IObservable.Unit.)</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues.T. (IAdsConnection, String, Int32., IObservable.Unit., Func.Exception, T.)</td>
<td>Polls the symbol values on timepoints where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>PollValues.T. (IAdsConnection, String, .Int32., TimeSpan)</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues.T. (IAdsConnection, String, Type, Int32., IObservable.Unit., Func.Exception, Object.)</td>
<td>Polls the symbol values on timepoints where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollValues(IAdsConnection, String, Type, TimeSpan, Func.Exception, Object.)</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
<tr>
<td>PollValues(IAdsConnection, String, Type, Int32, IObservable.Unit, Func.Exception, Object.)</td>
<td>Polls the symbol values on time points where the polling observable streams data / triggers</td>
</tr>
<tr>
<td>PollValues(IAdsConnection, String, Type, Int32, TimeSpan, Func.Exception, Object.)</td>
<td>Polls the symbol as value sequence of object values with a specified period time.</td>
</tr>
</tbody>
</table>

**Reference**

*AnyTypeExtensions Class* [1075]

*TwinCAT.Ads.Reactive Namespace* [1056]

**AnyTypeExtensions.PollValues.T Method (IAdsConnection, String, IObservable.Unit)**

Polls the symbol values on timepoints where the polling observable streams data / triggers

**Namespace:** TwinCAT.Ads.Reactive [1056]

**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    IObservable<Unit> trigger
)
```

**Parameters**

- **connection**
  Type: TwinCAT.Ads.IAdsConnection [765]
  The connection.

- **instancePath**
  Type: System.String
  The instance path.

- **trigger**
  Type: System.IObservable.Unit
  The Polling trigger
Type Parameters

T

The ANY_TYPE compatible .NET Type.

Return Value

Type: IObservable<T>. IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class
PollValues Overload
TwinCAT.Ads.Reactive Namespace

AnyTypeExtensions.PollValues.T. Method (IAdsConnection, String, TimeSpan)

Polls the symbol as value sequence of object values with a specified period time.

Namespace: TwinCAT.Ads.Reactive
Assembly: TwinCAT.Ads.Reactive.dll

Syntax

C#

```csharp
public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    TimeSpan period
)
```

Parameters

connection

Type: TwinCAT.Ads.IAdsConnection

The connection.

instancePath

Type: System.String

The instance path.

period

Type: System.TimeSpan

The period.

Type Parameters

T

The ANY_TYPE compatible .NET Type.

Return Value

Type: IObservable<T>. IObservable<System.Object>. 
Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class [765]

PollValues Overload [1081]

TwinCAT.Ads.Reactive Namespace [1056]

AnyTypeExtensions.PollValues.T. Method (IAdsConnection, String, .Int32., IObservable.Unit.)

Polls the symbol values on time points where the polling observable streams data / triggers

Namespace: TwinCAT.Ads.Reactive [1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    int[] args,
    IObservable<Unit> trigger
)
```

Parameters

- **connection**
  Type: TwinCAT.Ads.IAdsConnection [765]
  The connection.

- **instancePath**
  Type: System.String
  The instance path.

- **args**
  Type: System.Int32.
  ANY_TYPE arguments

- **trigger**
  Type: System.IObservable<Unit.
  The Polling trigger

Type Parameters

- **T**
  The ANY_TYPE compatible .NET Type.

Return Value

Type: IObservable<T.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).
AnyTypeExtensions.PollValues.T Method (IAdsConnection, String, .Int32., TimeSpan)

Polls the symbol as value sequence of object values with a specified period time.

**Namespace:** TwinCAT.Ads.Reactive

**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    int[] args,
    TimeSpan period
)
```

**Parameters**

- **connection**
  
  Type: TwinCAT.Ads.IAdsConnection
  
  The connection.

- **instancePath**
  
  Type: System.String
  
  The instance path.

- **args**
  
  Type: .System.Int32.
  
  ANY_TYPE arguments.

- **period**
  
  Type: System.TimeSpan
  
  The period.

**Type Parameters**

- **T**
  
  The ANY_TYPE compatible .NET Type.

**Return Value**

Type: IObservable<T>.

IObservable<System.Object>.

**Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

**Examples**

Polling ANY_TYPE values.
Observe changing ADS Symbols by polling (Read Polling) (ANY_TYPE)

// To Test the Observer run a project on the local PLC System (Port 851) using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInfo.CycleCount"];  

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
      Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToString()));
    });

    // Take 20 Values in an Interval of 500ms
    IDisposable subscription = cycleCount.PollValues(TimeSpan.FromMilliseconds(500)).Take(20).Subscribe(valueObserver);  

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}

Reference

AnyTypeExtensions Class [1075]

PollValues Overload [1081]

TwinCAT.Ads.Reactive Namespace [1056]


Polls the symbol values on timepoints where the polling observable streams data / triggers

Namespace: TwinCAT.Ads.Reactive [1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    IObservable<Unit> trigger,
    Func<Exception, T> errorHandler
)

Parameters

connection Type: TwinCAT.Ads.IAdsConnection [765]
The connection.

instancePath Type: System.String
The instance path.

trigger Type: System.IObservable.Unit.
The Polling trigger

errorHandler Type: System.Func.Exception, T.
The error handler.
### Type Parameters

**T**

The ANY_TYPE compatible .NET Type.

### Return Value

Type: `IObservable<T>`

`IObservable<System.Object>`.

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type `IAdsConnection`. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

### Reference

- **AnyTypeExtensions Class**
- **PollValues Overload**
- **TwinCAT.Ads.Reactive Namespace**

### AnyTypeExtensions.PollValues.T Method (IAdsConnection, String, TimeSpan, Func.Exception, T.)

Polls the symbol as value sequence of object values with a specified period time.

**Namespace:** TwinCAT.Ads.Reactive**

**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version:

5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    TimeSpan period,
    Func<Exception, T> errorHandler
)
```

#### Parameters

- **connection**
  
  Type: `TwinCAT.Ads.IAdsConnection`
  
  The connection.

- **instancePath**
  
  Type: `System.String`
  
  The instance path.

- **period**
  
  Type: `System.TimeSpan`
  
  The period.

- **errorHandler**
  
  Type: `System.Func<Exception, T>`
  
  The error handler.

### Type Parameters

**T**

The ANY_TYPE compatible .NET Type.
Return Value
Type: `IObservable<T>`. `IObservable<System.Object>`.

Usage Note
In Visual Basic and C#, you can call this method as an instance method on any object of type `IAdsConnection`. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference
AnyTypeExtensions class
PollValues Overload
TwinCAT.Ads.Reactive Namespace

AnyTypeExtensions.PollValues Method (IAdsConnection, String, Type, I Observable.Unit.)
Polls the symbol values on timepoints where the polling observable streams data / triggers

Syntax
C#
```csharp
public static IObservable<Object> PollValues(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    IObservable<Unit> trigger)
```

Parameters
- `connection`: Type: `TwinCAT.Ads.IAdsConnection`. The connection.
- `instancePath`: Type: `System.String`. The instance path.
- `type`: Type: `System.Type`. The data type of the symbol (ANY_TYPE)
- `trigger`: Type: `System.IObservable<Unit>`. The Polling trigger

Return Value
Type: `IObservable<Object>`. `IObservable<System.Object>`.

Usage Note
In Visual Basic and C#, you can call this method as an instance method on any object of type `IAdsConnection`. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).
AnyTypeExtensions.PollValues Method (IAdsConnection, String, Type, TimeSpan)

Polls the symbol as value sequence of object values with a specified period time.

**Namespace:** TwinCAT.Ads.Reactive

**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public static IObservable<Object> PollValues(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    TimeSpan period
)
```

**Parameters**

- **connection**
  - Type: TwinCAT.Ads.IAdsConnection
  - The connection.

- **instancePath**
  - Type: System.String
  - The instance path.

- **type**
  - Type: System.Type
  - The data type of the symbol (ANY_TYPE)

- **period**
  - Type: System.TimeSpan
  - The period.

**Return Value**

Type: IObservable<Object>, IObservable<System.Object>.

**Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

**Reference**

- AnyTypeExtensions Class [1075]
- PollValues Overload [1081]
- TwinCAT.Ads.Reactive Namespace [1056]

Polls the symbol values on timepoints where the polling observable streams data / triggers

Namespace: TwinCAT.Ads.Reactive {1056}
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static IObserverable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    int[] args,
    IObserverable<Unit> trigger,
    Func<Exception, T> errorHandler
)
```

Parameters

- connection
  - Type: TwinCAT.Ads.IAdsConnection {765}
  - The connection.
- instancePath
  - Type: System.String
  - The instance path.
- args
  - Type: System.Int32
  - ANY_TYPE arguments
- trigger
  - Type: System.IObservable.Unit
  - The Polling trigger
- errorHandler
  - Type: System.Func<Exception, T>
  - The error handler.

Type Parameters

- T
  - The ANY_TYPE compatible .NET Type.

Return Value

Type: IObserverable<T>, IObserverable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection {765}. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

- AnyTypeExtensions Class {1075}
- PollValues Overload {1081}
- TwinCAT.Ads.Reactive Namespace {1056}

Polls the symbol as value sequence of object values with a specified period time.

**Namespace:** TwinCAT.Ads.Reactive [1056]
**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    int[] args,
    TimeSpan period,
    Func<Exception, T> errorHandler
)
```

### Parameters

- **connection**
  - Type: TwinCAT.Ads.IAdsConnection [765]
  - The connection.
- **instancePath**
  - Type: System.String
  - The instance path.
- **args**
  - Type: System.Int32
  - ANY_TYPE arguments.
- **period**
  - Type: System.TimeSpan
  - The period.
- **errorHandler**
  - Type: System.Func<Exception, T>
  - The error handler.

### Type Parameters

- **T**
  - The ANY_TYPE compatible .NET Type.

### Return Value

- Type: IObservable<T>
  - IObservable<System.Object>.

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

### Examples

Polling ANY_TYPE values.

**Observe changing ADS Symbols by polling (Read Polling) (ANY_TYPE)**

```csharp
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));
    // Create Symbol information
```


var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
var valueObserver = Observer.Create<object>(val =>
{
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}"), cycleCount.InstancePath, val.ToString()));
});

// Take 20 Values in an Interval of 500ms
IDisposable subscription = cycleCount.PollValues(TimeSpan.FromMilliseconds(500)).Take(20).Subscribe(valueObserver);
Console.ReadKey(); // Wait for Key press
subscription.Dispose(); // Dispose the Subscription

Reference

AnyTypeExtensions Class [» 1075]
PollValues Overload [» 1081]
TwinCAT.Ads.Reactive Namespace [» 1056]

AnyTypeExtensions.PollValues Method (IAdsConnection, String, Type, .Int32., TimeSpan)

Polls the symbol as value sequence of object values with a specified period time.

Namespace: TwinCAT.Ads.Reactive [» 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static IObservable<Object> PollValues(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    int[] args,
    TimeSpan period
)

Parameters

connection Type: TwinCAT.Ads.IAdsConnection [» 765]
The connection.

instancePath Type: System.String
The instance path.

type Type: System.Type
The data type of the symbol (ANY_TYPE)

args Type: System.Int32
The ANY_TYPE arguments.

period Type: System.TimeSpan
The period.
Return Value
Type: IObservable<Object>, IObservable<System.Object>.

Usage Note
In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference
AnyTypeExtensions Class [1075]
PollValues Overload [1081]
TwinCAT.Ads.Reactive Namespace [1056]

AnyTypeExtensions.PollValues Method (IAdsConnection, String, Type, IObservable<Unit>, Func<Exception, Object>)
Polls the symbol values on timepoints where the polling observable streams data / triggers

Namespace: TwinCAT.Ads.Reactive [1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
```csharp
public static IObservable<Object> PollValues(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    IObservable<Unit> trigger,
    Func<Exception, Object> errorHandler
)
```

Parameters
- connection Type: TwinCAT.Ads.IAdsConnection [765]
The connection.
- instancePath Type: System.String
  The instance path.
- type Type: System.Type
  The data type of the symbol (ANY_TYPE)
- trigger Type: System.IObservable.Unit
  The Polling trigger
- errorHandler Type: System.Func<Exception, Object>
  The error handler.

Return Value
Type: IObservable<Object>, IObservable<System.Object>.
Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class
PollValues Overload
TwinCAT.Ads.Reactive Namespace

AnyTypeExtensions.PollValues Method (IAdsConnection, String, Type, TimeSpan, Func<Exception, Object, Object>, Object)

Polls the symbol as value sequence of object values with a specified period time.

Namespace: TwinCAT.Ads.Reactive
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static IObserver<Object> PollValues(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    TimeSpan period,
    Func<Exception, Object> errorHandler
)
```

Parameters

- connection
  Type: TwinCAT.Ads.IAdsConnection
  The connection.
- instancePath
  Type: System.String
  The instance path.
- type
  Type: System.Type
  The data type of the symbol (ANY_TYPE)
- period
  Type: System.TimeSpan
  The period.
- errorHandler
  Type: System.Func<Exception, Object>
  The error handler.

Return Value

Type: IObserver<Object>, IObserver<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).
AnyTypeExtensions.PollValues Method (IAdsConnection, String, Type, .Int32., IObservable.Unit., Func.Exception, Object.)

Polls the symbol values on time points where the polling observable streams data / triggers

**Namespace:** TwinCAT.Ads.Reactive

**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public static IObservable<Object> PollValues(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    int[] args,
    IObservable<Unit> trigger,
    Func<Exception, Object> errorHandler
)
```

**Parameters**

- **connection**
  - Type: `TwinCAT.Ads.IAdsConnection`
  - The connection.

- **instancePath**
  - Type: `System.String`
  - The instance path.

- **type**
  - Type: `System.Type`
  - The data type of the symbol (ANY_TYPE)

- **args**
  - Type: `System.Int32`
  - The ANY_TYPE arguments.

- **trigger**
  - Type: `System.IObservable<Unit>`
  - The Polling trigger

- **errorHandler**
  - Type: `System.Func<Exception, Object>`
  - The error handler.

**Return Value**

Type: `IObservable<Object>`, `IObservable<System.Object>`.

**Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type `IAdsConnection`. When you use instance method syntax to call this method, omit the first parameter.

For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

**Reference**

- AnyTypeExtensions Class [1075]
PollValues Overload [1081]

TwinCAT.Ads.Reactive Namespace [1056]

**AnyTypeExtensions.PollValues Method (IAdsConnection, String, Type, .Int32., TimeSpan, Func.Exception, Object.)**

Polls the symbol as value sequence of object values with a specified period time.

**Namespace:** TwinCAT.Ads.Reactive [1056]

**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static IObservable<Object> PollValues(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    int[] args,
    TimeSpan period,
    Func<Exception, Object> errorHandler
)
```

**Parameters**

- **connection**
  - Type: TwinCAT.Ads.IAdsConnection [765]
  - The connection.

- **instancePath**
  - Type: System.String
  - The instance path.

- **type**
  - Type: System.Type
  - The data type of the symbol (ANY_TYPE)

- **args**
  - Type: System.Int32.
  - The ANY_TYPE arguments.

- **period**
  - Type: System.TimeSpan
  - The period.

- **errorHandler**
  - Type: System.Func.Exception, Object
  - The error handler.

**Return Value**

Type: IObservable<Object>, IObservable<System.Object>.

**Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

**Reference**

- AnyTypeExtensions Class [1075]
- PollValues Overload [1081]
- TwinCAT.Ads.Reactive Namespace [1056]
6.3.2.1.2 AnyTypeExtensions.WhenNotification Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhenNotification.T.</td>
<td>Creates an observable sequence of values that are created by ADS Notifications.</td>
</tr>
<tr>
<td>(IAdsConnection, String, NotificationSettings)</td>
<td></td>
</tr>
<tr>
<td>WhenNotification(IAdsConnection, String, Type, NotificationSettings)</td>
<td>Creates an observable sequence of values that are created by ADS Notifications.</td>
</tr>
</tbody>
</table>

Reference

AnyTypeExtensions Class [1075]
TwinCAT.Ads.Reactive Namespace [1056]

AnyTypeExtensions.WhenNotification.T. Method (IAdsConnection, String, NotificationSettings)

Creates an observable sequence of values that are created by ADS Notifications.

**Namespace:** TwinCAT.Ads.Reactive [1056]
**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static IObservable<T> WhenNotification<T>(
    this IAdsConnection connection,
    string instancePath,
    NotificationSettings settings
)
```

**Parameters**

- **connection**
  Type: TwinCAT.Ads.IAdsConnection [765]
  The connection.

- **instancePath**
  Type: System.String
  The instance path.

- **settings**
  Type: TwinCAT.Ads(NotificationSettings) [979]
  The settings.

**Type Parameters**

- **T**
  The .NET Type representation of the specified symbols type.
Return Value

Type: IObservable<T>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Remarks

The values will be cast to the specified type. The .NET type must fit the Symbol type like all ANYTYPES.

Examples

The following sample shows how to observe Value changed Notifications with the reactive 
AnyTypeExtensions

Observe changing ADS Symbols with reactive extensions (Extended AdsNotification, ANY_TYPE)

// To Test the Observer run a project on the local PLC System (Port 851) 
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));
    // Reactive Notification Handler
    var valueObserver = Observer.Create<ushort>(val =>
    {
        Console.WriteLine(string.Format("Value: {0}", val.ToString()));
    });
    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values) 
    // and subscribe to them.
    IDisposable subscription = client.WhenNotification<ushort>("TwinCAT_SystemInfoVarList._TaskInfo.
        CycleCount", NotificationSettings.Default).Take(20).Subscribe(valueObserver);
    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}

Reference

AnyTypeExtensions Class [1075]
WhenNotification Overload [1098]
TwinCAT.Ads.Reactive Namespace [1056]

AnyTypeExtensions.WhenNotification Method (IAdsConnection, String, Type, NotificationSettings)

Creates an observable sequence of values that are created by ADS Notifications.

Namespace: TwinCAT.Ads.Reactive [1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

```csharp
public static IObservable<Object> WhenNotification(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    NotificationSettings settings
)
```

Parameters

- **connection**
  - Type: `TwinCAT.Ads.IAdsConnection` [765]
  - The connection.
- **instancePath**
  - Type: `System.String`
  - The instance path.
- **type**
  - Type: `System.Type`
  - The type.
- **settings**
  - Type: `TwinCAT.Ads.NotificationSettings` [979]
  - The settings.

Return Value

- Type: `IObservable<Object>`
- `IObservable<T>`.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type `IAdsConnection` [765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Remarks

The values will be cast to the specified type. The .NET type must fit be one of the compatible 'ANYTYPES'.

Examples

The following sample shows how to observe Value changed Notifications with the reactive `AnyTypeExtensions` [1075]

```
Observe changing ADS Symbols with reactive extensions (Extended AdsNotifications, ANY_TYPE)

// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Reactive Notification Handler
    var valueObserver = Observer.Create<ushort>(val =>
    {
        Console.WriteLine(string.Format("Value: {0}", val.ToString()));
    });

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = client.WhenNotification<ushort>("TwinCAT_SystemInfoVarList._TaskInfo.
CycleCount", NotificationSettings.Default).Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```
### AnyTypeExtensions.WriteValues Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValues.T. (IAdsConnection, String, IObservable.T)</td>
<td>Writes the sequence of values to the symbol specified by the instance path.</td>
</tr>
<tr>
<td>WriteValues.T. (IAdsConnection, String, IObservable.T, Action.Exception)</td>
<td>Writes the sequence of values to the symbol specified by the instance path.</td>
</tr>
</tbody>
</table>

**Reference**

- **AnyTypeExtensions Class [1075]**
- **TwinCAT.Ads.Reactive Namespace [1056]**

#### AnyTypeExtensions.WriteValues.T. Method (IAdsConnection, String, IObservable.T.)

Writes the sequence of values to the symbol specified by the instance path.

**Namespace:** TwinCAT.Ads.Reactive [1056]

**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static IDisposable WriteValues<T>(
    this IAdsConnection connection,
    string instancePath,
    IObservable<T> valueSequence
)
```

**Parameters**

- **connection**
  - Type: TwinCAT.Ads.IAdsConnection [765]
  - The connection.

- **instancePath**
  - Type: System.String
  - The instance path.
valueSequence

Type: System.IObserverable<T>
Value sequence (Any type).

Type Parameters

T

Return Value

Type: IDisposable
IDisposable.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

Write sequences of values to the target (ANY_TYPE)

```csharp
using {AdsClient client = new AdsClient();}
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information (Symbol 'i : INT' in PLC Global Variables list.
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol gvlIntSymbol = (IValueSymbol)symbolLoader.Symbols["GVL.i"];

    // Produces object (short) Values 0,1,2,3 ... in seconds period
    IObservable<object> timerObservable = Observable.Interval(TimeSpan.FromSeconds(1.0)).Select(i => (object)(short)i);

    // Take 10 Values (0..9) and write them to GVL.i
    IDisposable dispose = gvlIntSymbol.WriteValues(timerObservable.Take(10));

    Console.ReadKey(); // Wait for Key press
    dispose.Dispose(); // Dispose the Subscription
}
```

Reference

AnyTypeExtensions Class [1075]

WriteValues Overload [1101]

TwinCAT.Ads.Reactive Namespace [1056]

AnyTypeExtensions.WriteValues.T. Method (IAdsConnection, String, IObserverable.T., Action.Exception.)

Writes the sequence of values to the symbol specified by the instance path.

Namespace: TwinCAT.Ads.Reactive [1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public static IDisposable WriteValues<T>(
    this IAdsConnection connection,
    string instancePath,
    IObservable<T> valueSequence,
    Action<Exception> errorHandler
)
```

Parameters

- **connection**
  - Type: `TwinCAT.Ads.IAdsConnection` [765]
  - The connection.

- **instancePath**
  - Type: `System.String`
  - The instance path.

- **valueSequence**
  - Type: `System.IObservable<T>`
  - Value sequence (Any type).

- **errorHandler**
  - Type: `System.Action<Exception>`
  - The error handler.

Type Parameters

- **T**

Return Value

- Type: `IDisposable`
  - `IDisposable`.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type `IAdsConnection` [765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

Write values sequentially.

Write sequences of values to the target (ANY_TYPE)

```csharp
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information (Symbol 'i : INT' in PLC Global Variables list.
    IValueSymbol gvlIntSymbol = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);

    // Produces object (short) Values 0,1,2,3 ... in seconds period
    IObservable<object> timerObservable = Observable.Interval(TimeSpan.FromSeconds(1.0)).Select(i =>
        (object)(short)i);

    // Take 10 Values (0..9) and write them to GVL.i
    IDisposable dispose = gvlIntSymbol.WriteValues(timerObservable.Take(10));

    Console.ReadKey(); // Wait for Key press
    dispose.Dispose(); // Dispose the Subscription
}
```
6.3.3 SymbolValueNotification Class

Symbol Notification class

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.Notification [974]
    ValueNotification
      TwinCAT.Ads.Reactive.SymbolValueNotification

Namespace: TwinCAT.Ads.Reactive [1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public sealed class SymbolValueNotification : ValueNotification

The SymbolValueNotification type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data [977]</td>
<td>The notification Data. (Inherited from Notification [974].)</td>
</tr>
<tr>
<td>Handle [977]</td>
<td>The notification handle (Inherited from Notification [974].)</td>
</tr>
<tr>
<td>Symbol [1105]</td>
<td>Gets the symbol of the SymbolValueNotification.</td>
</tr>
<tr>
<td>TimeStamp [978]</td>
<td>Gets the time stamp of the INotification [969] (Inherited from Notification [974].)</td>
</tr>
<tr>
<td>UserData [978]</td>
<td>Attached UserData/Tag at the INotification [969] (Inherited from Notification [974].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Remarks

Extends the ValueNotification class by symbol specific information.
6.3.3.1 SymbolValueNotification Properties

The SymbolValueNotification [1104] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data [977]</td>
<td>The notification Data. (Inherited from Notification [974].)</td>
</tr>
<tr>
<td>Handle [977]</td>
<td>The notification handle (Inherited from Notification [974].)</td>
</tr>
<tr>
<td>Symbol [1105]</td>
<td>Gets the symbol of the SymbolValueNotification [1104].</td>
</tr>
<tr>
<td>TimeStamp [978]</td>
<td>Gets the time stamp of the INotification [969] (Inherited from Notification [974].)</td>
</tr>
<tr>
<td>UserData [978]</td>
<td>Attached UserData/Tag at the INotification [969] (Inherited from Notification [974].)</td>
</tr>
</tbody>
</table>

6.3.3.2 SymbolValueNotification Methods

The SymbolValueNotification [1104] type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Reference

SymbolValueNotification Class [1104]

TwinCAT.Ads.Reactive Namespace [1056]

### 6.3.4 ValueSymbolExtensions Class

Extension class for IAdsConnection [765] to provide reactive ADS extensions for accessing symbols that are loaded by the IAdsSymbolLoaderFactory

### Inheritance Hierarchy

System.Object
   TwinCAT.Ads.Reactive.ValueSymbolExtensions

Namespace: TwinCAT.Ads.Reactive [1056]

Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

C#

```csharp
public static class ValueSymbolExtensions
```

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollValuesAnnotateValueChangedEventArgs(T)</td>
<td>Polls the values as ValueChangedEventArgs [2439] sequence annotated value on trigger sequence</td>
</tr>
<tr>
<td>PollValuesAnnotateValueChangedEventArgs(T, TimeSpan)</td>
<td>Polls the values as ValueChangedEventArgs [2439] sequence with a specified period time.</td>
</tr>
<tr>
<td>WhenValueChanged(IValueSymbol)</td>
<td>Gets an observable sequence when the value of the IValueSymbol [2254] has changed.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WhenValueChanged(IAdsConnection, IEnumerable.ISymbol.)</td>
<td>Observable sequence of Values driven by ADS Notifications on the specified symbol.</td>
</tr>
<tr>
<td>WriteValues(IValueSymbol, IObservable.Object.)</td>
<td>Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol.</td>
</tr>
<tr>
<td>WriteValues(IValueSymbol, IObservable.Object, Action.Exception.)</td>
<td>Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol.</td>
</tr>
<tr>
<td>WriteValues(IValueSymbol, IObservable.Object, CancellationToken)</td>
<td>Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol.</td>
</tr>
<tr>
<td>WriteValues(IValueSymbol, IObservable.Object, Action.Exception, CancellationToken)</td>
<td>Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol.</td>
</tr>
</tbody>
</table>

**Remarks**

Reactive Extensions (Rx) are a library for composing asynchronous and event-based programs using observable sequences and LINQ-style query operators. Using Rx, developers represent asynchronous data streams with Observables, query asynchronous data streams using LINQ operators, and parameterize the concurrency in the asynchronous data streams using Schedulers. Simply put, Rx = Observables + LINQ + Schedulers. The ADS reactive extensions are build on top of this library to enable ADS Symbol and State Observables, seamlessly bound to the reactive extensions. To use the ADS reactive extensions the TwinCAT.Ads.Reactive Nuget package (or the included TwinCAT.Ads.Reactive.dll) must be referenced from All types within are contained in the ADS companion package "Beckhoff.TwinCAT.Ads.Reactive" which must be referenced separately. (Beckhoff.TwinCAT.Ads.Reactive package on Nuget).

**Examples**

The following sample shows how to observe Value changed Notifications with the reactive ValueSymbolExtensions from an IValueSymbol.

**Observe a single changing ADS Symbol (ADS Notifications)**

```csharp
// To Test the Observer run a project on the local PLC System (Port 851) using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
```
Examples

The following sample shows how to observe Value changed Notifications with the reactive ValueSymbolExtensions from an DynamicSymbol.

Observe a single changing ADS Symbol (ADS Notifications) with the dynamic language runtime (.NET DLR)

```csharp
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
        Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToString()));
    });

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransModeOnChange, 500, 5000); // optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.WhenValueChanged().Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Examples

The same for more than one IValueSymbol ▶ 2254.

Observe changing ADS Symbols (ADS Notifications)

```csharp
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
        Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToString()));
    });

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransModeOnChange, 500, 5000); // optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.WhenValueChanged().Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```
Examples

Here, the values are polled in a specific time period and sequential Reads are triggered (in opposite to ADS Notification in the latter example)

Observe changing ADS Symbols by polling (Read Polling)

// To Test the Observer run a project on the local PLC System (Port 851) using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInfo[1].CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
        Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToString()));
    });

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransModeOnChange, 500, 5000); // optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.PollValues(TimeSpan.FromMilliseconds(500)).Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}

Examples

In the following example it is demonstrated how to write Values sequentially to a IValueSymbol with the help of the reactive extensions.

Write sequences of values to the target

using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information (Symbol 'i : INT' in PLC Global Variables list.
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["GVL.i"];
// Produces object Values 0,1,2,3 ... in seconds period
IObservable<object> timerObservable = Observable.Interval(TimeSpan.FromSeconds(1.0)).Select(i => (object)(short)i);

// Take 10 Values (0..9) and write them to GVL.i
IDisposable dispose = cycleCount.WriteValues(timerObservable.Take(10));

Console.ReadKey(); // Wait for Key press
dispose.Dispose(); // Dispose the Subscription

Reference

TwinCAT.Ads.Reactive Namespace [1 056]
TwinCAT.Ads.Reactive.AdsClientExtensions [1 056]
TwinCAT.Ads.Reactive.AnyTypeExtensions [1 075]

6.3.4.1 ValueSymbolExtensions Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollValuesAnnotated(IValueSymbol, IObservable&lt;Unit&gt;) [1 111]</td>
<td>Polls the values as ValueChangedEventArgs [2 439] sequence annotated value on trigger sequence</td>
</tr>
<tr>
<td>PollValuesAnnotated(IValueSymbol, TimeSpan) [1 112]</td>
<td>Polls the values as ValueChangedEventArgs [2 439] sequence with a specified period time.</td>
</tr>
<tr>
<td>WhenValueChanged(IValueSymbol) [1 113]</td>
<td>Gets an observable sequence when the value of the IValueSymbol [2 254] has changed.</td>
</tr>
<tr>
<td>WhenValueChanged(IAdsConnection, IEnumerable.ISymbol) [1 114]</td>
<td>Observable sequence of Values driven by ADS Notifications on the specified symbol.</td>
</tr>
<tr>
<td>WriteValues(IValueSymbol, IObservable&lt;Object&gt;) [1 117]</td>
<td>Subscribes the IValueSymbol [2 254] to an observable sequence of values and writes them to the IValueSymbol [2 254].</td>
</tr>
<tr>
<td>WriteValues(IValueSymbol, IObservable&lt;Object, Action.Exception&gt;) [1 118]</td>
<td>Subscribes the IValueSymbol [2 254] to an observable sequence of values and writes them to the IValueSymbol [2 254].</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteValues(IValueSymbol, IObservable.Object, CancellationToken)</td>
<td>Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol.</td>
</tr>
<tr>
<td>WriteValues(IValueSymbol, IObservable.Object, Action.Exception, CancellationToken)</td>
<td>Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol.</td>
</tr>
</tbody>
</table>

Reference

ValueSymbolExtensions Class [1106]
TwinCAT.Ads.Reactive Namespace [1056]

6.3.4.1.1 ValueSymbolExtensions.PollValuesAnnotated Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollValuesAnnotated(IValueSymbol, IObservable.Unit)</td>
<td>Polls the values as ValueChangedEventArgs sequence annotated value on trigger sequence</td>
</tr>
<tr>
<td>PollValuesAnnotated(IValueSymbol, TimeSpan)</td>
<td>Polls the values as ValueChangedEventArgs sequence with a specified period time.</td>
</tr>
</tbody>
</table>

Reference

ValueSymbolExtensions Class [1106]
TwinCAT.Ads.Reactive Namespace [1056]

ValueSymbolExtensions.PollValuesAnnotated Method (IValueSymbol, IObservable.Unit.)

Polls the values as ValueChangedEventArgs sequence annotated value on trigger sequence

Namespace: TwinCAT.Ads.Reactive [1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public static IObservable<ValueChangedEventArgs> PollValuesAnnotated(
    this IValueSymbol symbol,
    IObservable<Unit> trigger
)
```

Parameters

- **symbol**: Type: `TwinCAT.TypeSystem.IValueSymbol` [2254]
The symbol.

- **trigger**: Type: `System.IObservable<Unit>`
The polling Trigger.

Return Value

Type: `IObservable<ValueChangedEventArgs>` [2439].

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type `IValueSymbol` [2254]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

- `ValueSymbolExtensions Class` [1106]
- `PollValuesAnnotated Overload` [1111]
- `TwinCAT.Ads.Reactive Namespace` [1056]

### ValueSymbolExtensions.PollValuesAnnotated Method (IValueSymbol, TimeSpan)

Polls the values as `ValueChangedEventArgs` [2439] sequence with a specified period time.

**Namespace:** `TwinCAT.Ads.Reactive` [1056]

**Assembly:** `TwinCAT.Ads.Reactive` (in `TwinCAT.Ads.Reactive.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static IObservable<ValueChangedEventArgs> PollValuesAnnotated(
    this IValueSymbol symbol,
    TimeSpan period
)
```

Parameters

- **symbol**: Type: `TwinCAT.TypeSystem.IValueSymbol` [2254]
The symbol.

- **period**: Type: `System.TimeSpan`
The polling period/interval.
Return Value

Type: IObservable_ValueChangedEventArgs |2439 |
IObservable_ValueChangedArgs>.  

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IValueSymbol |2254 |. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).  

Reference

ValueSymbolExtensions Class |1106 |
PollValuesAnnotated Overload |1111 |
TwinCAT.Ads.Reactive Namespace |1056 |  

6.3.4.1.2 ValueSymbolExtensions.WhenValueChanged Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ] WhenValueChanged(IValueSymbol)</td>
<td>Gets an observable sequence when the value of the IValueSymbol</td>
</tr>
<tr>
<td>![ ] WhenValueChanged(IAdsConnection, IEnumerable.ISymbol)</td>
<td>Observable sequence of Values driven by ADS Notifications on the specified symbol.</td>
</tr>
</tbody>
</table>

Reference

ValueSymbolExtensions Class |1106 |
TwinCAT.Ads.Reactive Namespace |1056 |  

ValueSymbolExtensions.WhenValueChanged Method (IValueSymbol)

Gets an observable sequence when the value of the IValueSymbol |2254 | has changed.  

Namespace: TwinCAT.Ads.Reactive |1056 |  
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14  

Syntax

C#  

```csharp
public static IObservable<Object> WhenValueChanged(
    this IValueSymbol symbol
)
```
Parameters

symbol Type: TwinCAT>TypeSystem.IValueSymbol
The symbol.

Return Value

Type: IObservable<Object, IObservable<System.Object>>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IValueSymbol. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

The following sample shows how to observe Value changed Notifications with the reactive ValueSymbolExtensions from an IValueSymbol.

Observe a single changing ADS Symbols (ADS Notifications)

```csharp
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInfo[1].CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
        Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToString()));
    });

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransModeOnChange, 500, 5000); // optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.WhenValueChanged().Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

ValueSymbolExtensions Class

WhenValueChanged Overload

TwinCAT.Ads Reactive Namespace

ValueSymbolExtensions.WhenValueChanged Method (IAdsConnection, IEnumerable.ISymbol.)

Observable sequence of Values driven by ADS Notifications on the specified symbol.
Namespace: TwinCAT.Ads.Reactive [1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static IObserver<Object> WhenValueChanged(
    this IAdsConnection connection,
    IEnumerable<ISymbol> symbols
)
```

Parameters

- **connection**
  - Type: TwinCAT.Ads.IAdsConnection [765]
  - The ADS connection / ADS Client
- **symbols**
  - Type: System.Collections.Generic.IEnumerable.ISymbol [2176]
  - The symbols to observe.

Return Value

- Type: IObserver Object.
  - IObserver<ValueChangedEventArgs>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [765]. When you use instance method syntax to call this method, omit the first parameter.

For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

The same for more than one IValueSymbol [2254].

Observe changing ADS Symbols (ADS Notifications)

```csharp
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInfo.CycleCount"];// UShort Type
    IValueSymbol lastExecTime = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInfo.LastExecTime"];// UInt Type

    SymbolCollection symbols = new SymbolCollection();
    symbols.Add(cycleCount);
    symbols.Add(lastExecTime);

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    { Console.WriteLine(string.Format("Instance: {0}, Value: {1}"), cycleCount.InstancePath, val.ToString()));
    });

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransModeOnChange, 500, 5000); // optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = client.WhenValueChanged(symbols).Take(20).Subscribe(valueObserver);
```
Examples

The following sample shows how to observe Value changed Notifications with the reactive ValueSymbolExtensions [1106] from an DynamicSymbol.

Observe a single changing ADS Symbol (ADS Notifications) with the dynamic language runtime (.NET DLR)

```csharp
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf o[1].CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
        Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToString()));
    });

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransModeOnChange, 500, 5000); // optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.WhenValueChanged().Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

ValueSymbolExtensions Class [1106]

WhenValueChanged Overload [1113]

TwinCAT.Ads.Reactive Namespace [1056]

Observable

6.3.4.1.3 ValueSymbolExtensions.WriteValues Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValues(<a href="#valueSymbol">ValueSymbol</a> <img src="#observableObject" alt="Observable Object" />) ![1117]</td>
<td>Subscribes the <a href="#valueSymbol">ValueSymbol</a> to an observable sequence of values and writes them to the <a href="#valueSymbol">ValueSymbol</a>.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteValues(IValueSymbol, IObservable.Object, Action.Exception.) [1118]</td>
<td>Subscribes the IValueSymbol [2254] to an observable sequence of values and writes them to the IValueSymbol [2254].</td>
</tr>
<tr>
<td>WriteValues(IValueSymbol, IObservable.Object, CancellationToken) [1119]</td>
<td>Subscribes the IValueSymbol [2254] to an observable sequence of values and writes them to the IValueSymbol [2254].</td>
</tr>
<tr>
<td>WriteValues(IValueSymbol, IObservable.Object, Action.Exception, CancellationToken) [1120]</td>
<td>Subscribes the IValueSymbol [2254] to an observable sequence of values and writes them to the IValueSymbol [2254].</td>
</tr>
</tbody>
</table>

**Reference**

ValueSymbolExtensions Class [1106]

TwinCAT.Ads.Reactive Namespace [1056]

**ValueSymbolExtensions.WriteValues Method (IValueSymbol, IObservable.Object.)**

Subscribes the IValueSymbol [2254] to an observable sequence of values and writes them to the IValueSymbol [2254].

**Namespace:** TwinCAT.Ads.Reactive [1056]

**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public static IDisposable WriteValues(
    this IValueSymbol symbol,
    IObservable<Object> valueObservable
)
```

**Parameters**

- **symbol**
  
  Type: TwinCAT.TypeSystem.IValueSymbol [2254]
  
  The symbol.

- **valueObservable**
  
  Type: System.IObservable<Object>
  
  Observable of Values.

**Return Value**

Type: IDisposable

IDisposable.
Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type `IValueSymbol` [2254]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

In the following example it is demonstrated how to write Values sequentially to a `IValueSymbol` [2254] with the help of the reactive extensions.

Write sequences of values to the target

```
using {AdClient client = new AdClient()}
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information (Symbol 'i : INT' in PLC Global Variables list.
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["GVL.i"];

    // Produces object Values 0,1,2,3 ... in seconds period
    IObservable<object> timerObservable = Observable.Interval(TimeSpan.FromSeconds(1.0)).Select(i => (object)(short)i);

    // Take 10 Values (0..9) and write them to GVL.i
    IDisposable dispose = cycleCount.WriteValues(timerObservable.Take(10));

    Console.ReadKey(); // Wait for Key press
    dispose.Dispose(); // Dispose the Subscription
}
```

Reference

ValueSymbolExtensions Class [1106]

WriteValues Overload [1116]

TwinCAT.Ads.Reactive Namespace [1056]


Subscribes the `IValueSymbol` [2254] to an observable sequence of values and writes them to the `IValueSymbol` [2254].

Namespace: TwinCAT.Ads.Reactive [1056]

Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IDisposable WriteValues(
    this IValueSymbol symbol,
    IObservable<object> valueObservable,
    Action<Exception> errorHandler
)
```

Parameters

- **symbol**
  - Type: TwinCAT.TypeSystem.IValueSymbol [2254]
  - The symbol.
valueObservable  Type: System.IObservable<Object>
Observable of Values.

errorHandler  Type: System.Action<Exception>
The error handler or NULL.

Return Value
Type: IDisposable
IDisposable.

Usage Note
In Visual Basic and C#, you can call this method as an instance method on any object of type IValueSymbol
[2254]. When you use instance method syntax to call this method, omit the first parameter. For more
information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference
ValueSymbolExtensions Class [1106]
WriteValues Overload [1116]
TwinCAT.Ads.Reactive Namespace [1056]

ValueSymbolExtensions.WriteValues Method (IValueSymbol, IObservable<Object>, CancellationToken)
Subscribes the IValueSymbol [2254] to an observable sequence of values and writes them to the
IValueSymbol [2254].

Namespace: TwinCAT.Ads.Reactive [1056]
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#

public static void WriteValues(
    this IValueSymbol symbol,
    IObservable<Object> valueObservable,
    CancellationToken cancel
)

Parameters
symbol  Type: TwinCAT.TypeSystem.IValueSymbol [2254]
The symbol.

valueObservable  Type: System.IObservable<Object>
Observable of Values.

cancel  Type: System.Threading.CancellationToken
The cancellation token.

Return Value
Type: IDisposable.
**Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type `IValueSymbol` [\ref{2254}]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

**Reference**

`ValueSymbolExtensions Class` [\ref{1106}]

`WriteValues Overload` [\ref{1116}]

`TwinCAT.Ads Reactive Namespace` [\ref{1056}]


Subscribes the `IValueSymbol` [\ref{2254}] to an observable sequence of values and writes them to the `IValueSymbol` [\ref{2254}].

**Namespace:** TwinCAT.Ads.Reactive [\ref{1056}]

**Assembly:** TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static void WriteValues(
    this IValueSymbol symbol,
    IObservable<Object> valueObservable,
    Action<Exception> errorHandler,
    CancellationToken cancel
)
```

**Parameters**

- `symbol` Type: `TwinCAT.TypeSystem.IValueSymbol` [\ref{2254}]
The symbol.

- `valueObservable` Type: `System.IObservable` Object.
  Observable of Values.

- `errorHandler` Type: `System.Action` Exception.
  The error handler.

- `cancel` Type: `System.Threading.CancellationToken`
  The cancellation token.

**Return Value**

Type: `IDisposable`.

**Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type `IValueSymbol` [\ref{2254}]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).
Reference

ValueSymbolExtensions Class [1106]
WriteValues Overload [1116]
TwinCAT.Ads.Reactive Namespace [1056]

6.4 TwinCAT.Ads.Server Namespace

Namespace for the TwinCAT ADS Server Component.

Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsServer [1121]</td>
<td>Base implementation for an ADS Server.</td>
</tr>
<tr>
<td>ErrorEventArgs [1186]</td>
<td>This class implements the event arguments passed by the TcAdsServerExEvent.</td>
</tr>
<tr>
<td>LoopbackNotRegisteredException [1188]</td>
<td>The Tcp Loopback client is not registered. Implements the AdsServerException [1181]</td>
</tr>
<tr>
<td>NotificationDataSample [1191]</td>
<td>This class implements an ADS Notification Sample. It contains the notification handle and the variable data.</td>
</tr>
<tr>
<td>NotificationSamplesStamp [1195]</td>
<td>This class implements an ADS Stamp Header containing multiple ADS Notification Samples (TcAdsStampHeader)</td>
</tr>
<tr>
<td>ServerConnectionStateChangedEventArgs [1199]</td>
<td>Class ConnectionStateChangedEventArgs (Server Connections)</td>
</tr>
<tr>
<td>ServerNotConnectedException [1202]</td>
<td>The AdsServer is not connected. Implements the AdsServerException [1181]</td>
</tr>
</tbody>
</table>

Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServerConnectionState [1198]</td>
<td>The Server Connection State</td>
</tr>
</tbody>
</table>

6.4.1 AdsServer Class

Base implementation for an ADS Server.
Inheritance Hierarchy

- System.Object
  - TwinCAT.Ads.Server.AdsServer

**Namespace:** TwinCAT.Ads.Server [1121]

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

**C#**

```csharp
public abstract class AdsServer : IDisposable
```

The AdsServer type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsConnected [1136]</td>
<td>Gets a value indicating whether AdsServer is connected.</td>
</tr>
<tr>
<td>IsDisconnecting</td>
<td>Indicates, that the AdsServer is actually disconnecting.</td>
</tr>
<tr>
<td>IsDisposed [1137]</td>
<td>Gets a value indicating whether this instance is disposed.</td>
</tr>
<tr>
<td>Logger [1137]</td>
<td>Gets the logger object.</td>
</tr>
<tr>
<td>ServerAddress</td>
<td>The AMS address of this server.</td>
</tr>
</tbody>
</table>

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDeviceNotificationConfirmationAsync [1142]</td>
<td>Called when an ADS Add Device Notification confirmation is received.</td>
</tr>
<tr>
<td>AddDeviceNotificationIndicationAsync [1143]</td>
<td>Called when an ADS Add Device Notification indication is received.</td>
</tr>
<tr>
<td>AddDeviceNotificationRequest [1144]</td>
<td>Sends an ADS Add Device Notification request (synchronous).</td>
</tr>
<tr>
<td>AddDeviceNotificationRequestAsync [1144]</td>
<td>Sends an ADS Add Device Notification request (async)</td>
</tr>
<tr>
<td>ConnectServer [1146]</td>
<td>Connect this ADS server to the local ADS router.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>ConnectServerAnd</strong>&lt;br&gt;<strong>WaitAsync</strong> [1147]</td>
<td>Registers the AdsServer at the router asynchronously.</td>
</tr>
<tr>
<td><strong>DeleteDeviceNotificationConfirmationAsync</strong> [1147]</td>
<td>Called when an ADS Delete Device Notification confirmation is received.</td>
</tr>
<tr>
<td><strong>DeleteDeviceNotificationIndicationAsync</strong> [1148]</td>
<td>Called when an ADS Delete Device Notification indication is received.</td>
</tr>
<tr>
<td><strong>DeleteDeviceNotificationRequest</strong> [1149]</td>
<td>Sends an ADS Delete Device Notification request (synchronous).</td>
</tr>
<tr>
<td><strong>DeleteDeviceNotificationRequestAsync</strong> [1149]</td>
<td>Sends an ADS Delete Device Notification request (async).</td>
</tr>
<tr>
<td><strong>DeleteDeviceNotificationResponseAsync</strong> [1150]</td>
<td>Sends an ADS Delete Device Notification response.</td>
</tr>
<tr>
<td><strong>DeviceNotificationRequestAsync</strong> [1151]</td>
<td>Sends an ADS Device Notification request asynchronously</td>
</tr>
<tr>
<td><strong>DeviceNotificationRequestSync</strong> [1152]</td>
<td>Sends an ADS Device Notification request (sync)</td>
</tr>
<tr>
<td><strong>Disconnect</strong> [1152]</td>
<td>Disconnects this ADS server from the local ADS router.</td>
</tr>
<tr>
<td><strong>Dispose</strong> [1153]</td>
<td>Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.</td>
</tr>
<tr>
<td><strong>Dispose(Boolean)</strong> [1154]</td>
<td>Releases unmanaged and - optionally - managed resources.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td><strong>Finalize</strong> [1154]</td>
<td>Finalizes an instance of the AdsServer class. (Overrides <strong>Object.Finalize</strong>)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td><strong>OnAddDeviceNotificationConfirmationAsync</strong> [1155]</td>
<td>Called when Add device Notification Confirmation is send.</td>
</tr>
<tr>
<td><strong>OnConnected</strong> [1155]</td>
<td>Handler function that is called, when the AdsServer is connected.</td>
</tr>
<tr>
<td><strong>OnDisconnect</strong> [1156]</td>
<td>Called when the AdsServer is about to be disconnected.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>OnRouterNotification</td>
<td>Handler Function for a Router Notification.</td>
</tr>
<tr>
<td>OnServerConnectionStateChanged</td>
<td>Handles the ServerConnectionStateChanged [¶ 1181] event.</td>
</tr>
<tr>
<td>ReadConfirmationAsync</td>
<td>Called when an ADS Read Device Info confirmation is received.</td>
</tr>
<tr>
<td>ReadDeviceInfoConfirmationAsync</td>
<td>Called when an ADS Read Device Info indication is received.</td>
</tr>
<tr>
<td>ReadDeviceInfoIndicationAsync</td>
<td>Sends an ADS Read Device Info request asynchronously.</td>
</tr>
<tr>
<td>ReadDeviceInfoRequestAsync</td>
<td>Sends an ADS Read Device Info request synchronously.</td>
</tr>
<tr>
<td>ReadDeviceInfoResponseStatusAsync</td>
<td>Sends an ADS Read Device Info response.</td>
</tr>
<tr>
<td>ReadDeviceStateConfirmationAsync</td>
<td>Called when an ADS Read State confirmation is received.</td>
</tr>
<tr>
<td>ReadDeviceStateIndicationAsync</td>
<td>Called when an ADS Read State indication is received.</td>
</tr>
<tr>
<td>ReadDeviceStateRequestAsync</td>
<td>Sends an ADS Read State request (asynchronous)</td>
</tr>
<tr>
<td>ReadDeviceStateRequestSync</td>
<td>Sends an ADS Read State request (synchronous)</td>
</tr>
<tr>
<td>ReadDeviceStateResponseAsync</td>
<td>Sends an ADS Read State response.</td>
</tr>
<tr>
<td>ReadIndicationAsync</td>
<td>Called when an ADS Read indication is received.</td>
</tr>
<tr>
<td>ReadRequest</td>
<td>Sends an ADS Read Request.</td>
</tr>
<tr>
<td>ReadRequestAsync</td>
<td>Sends an ADS Read Request asynchronously.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadResponseAsync</td>
<td></td>
</tr>
<tr>
<td>ReadWriteConfirmationAsync</td>
<td></td>
</tr>
<tr>
<td>ReadWriteIndicationAsync</td>
<td></td>
</tr>
<tr>
<td>ReadWriteRequestAsync</td>
<td></td>
</tr>
<tr>
<td>ReadWriteRequestSync</td>
<td></td>
</tr>
<tr>
<td>ReadWritegetResponseAsync</td>
<td></td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>WriteConfirmationAsync</td>
<td>Called when an ADS Write confirmation is received. Overwrite this method in derived classes to react on ADS Write confirmations.</td>
</tr>
<tr>
<td>WriteControlConfirmationAsync</td>
<td>Called when an ADS Write Control confirmation is received.</td>
</tr>
<tr>
<td>WriteControlIndicationAsync</td>
<td></td>
</tr>
<tr>
<td>WriteControlRequest</td>
<td></td>
</tr>
<tr>
<td>WriteControlRequestAsync</td>
<td></td>
</tr>
<tr>
<td>WriteControlRequestSync</td>
<td></td>
</tr>
<tr>
<td>WriteControlResponseAsync</td>
<td>Sends an ADS Write Control response.</td>
</tr>
<tr>
<td>WriteIndicationAsync</td>
<td></td>
</tr>
<tr>
<td>WriteRequest</td>
<td></td>
</tr>
<tr>
<td>WriteRequestAsync</td>
<td></td>
</tr>
<tr>
<td>WriteResponseAsync</td>
<td>Sends an ADS Write response.</td>
</tr>
</tbody>
</table>
Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServerConnectionState Changed</td>
<td>The connection status has changed</td>
</tr>
</tbody>
</table>

Remarks

Derived classes should overwrite the indication methods to react on incoming requests. The confirmation methods should be overwritten to receive replies on asynchronous requests sent by this ADS server.

Examples

The following sample shows how to derive from the AdsServer class and create your own Customized ADS Server.

AdsSampleServer

C# class Program
{
    public static void Main(string[] args)
    {
        CreateHostBuilder(args).Build().Run();
    }

    public static IHostBuilder CreateHostBuilder(string[] args) =>
        Host.CreateDefaultBuilder(args)
            .ConfigureServices((hostContext, services) =>
            {
                services.AddHostedService<ServerWorker>();
            });
}

C# public class ServerWorker : BackgroundService
{
    private readonly ILogger<ServerWorker> _logger;

    public ServerWorker(ILogger<ServerWorker> logger)
    {
        _logger = logger;
    }

    protected override async Task ExecuteAsync(CancellationToken cancel)
    {
        // Instantiate the server
        AdsSampleServer server = new AdsSampleServer(_logger);
        // Connect the server and wait for cancel
        await server.ConnectServerAndWaitAsync(cancel);
    }
}

C# /*
 * Extend the AdsServer class to implement your own ADS server.
 */
public class AdsSampleServer : AdsServer
{
    /// <summary>
    /// Fixed ADS Port (to be changed ...)
    /// </summary>
    const ushort ADS_PORT = 42;

    /// <summary>
    /// Fixed Name for the ADS Port (change this ...)
    /// </summary>
}
const string ADS_PORT_NAME = "AdsSampleServer_Port42";

private byte[] _dataBuffer = {1, 2, 3, 4};

private AdsState _adsState = AdsState.Config;

private ushort _deviceState = 0;

private ConcurrentDictionary<uint, NotificationRequestEntry> _notificationTable = new ConcurrentDictionary<uint, NotificationRequestEntry>();

private uint _currentNotificationHandle = 0;

public AdsSampleServer()
: this(null)
{
}

public AdsSampleServer(ILogger logger) : base(ADS_PORT, ADS_PORT_NAME)
{
_logger = logger;
}

public AdsSampleServer(ILogger logger) : base(ADS_PORT, ADS_PORT_NAME)
{
_logger = logger;
}

private void LogTrace(string message, params object[] args)
{
if (_logger != null)
    _logger.LogTrace(message, args);
}

private void LogInformation(string message, params object[] args)
{
if (_logger != null)
    _logger.LogInformation(message, args);
}

private void LogError(string message, params object[] args)
{
if (_logger != null)
    _logger.LogError(message, args);
}
/// AdsServer Version
/// </summary>
static AdsVersion s_version = new AdsVersion(0, 0, 1);

/* Overwrite the indication methods of the TCAdsServer class for the services your ADS server
* provides. They are called upon incoming requests. All indications that are not overwritten in
* this class return the ADS DeviceServiceNotSupported error code to the requester.
*/

/// Called when an ADS Read Device Info indication is received by your <see cref="AdsSampleServer"/>
/// </summary>
/// <remarks>
/// Overwrite this method in derived classes to react on ADS Read Device Info indications.
/// The default implementation replies with an <see AdsErrorCode.DeviceServiceNotSupported>error code (0x701).
/// </remarks>
/// <param name="sender">The sender's / requester's AMS address</param>
/// <param name="invokeId">The invokeId provided by the sender</param>
/// <param name="cancel">The cancellation token.</param>
/// <returns>A task that represents the asynchronous <see cref="OnReadDeviceInfoIndicationAsync" /> parameter contains the <see cref="AdsErrorCode"/> as
/// <see cref="Task(Task).Result"/>.
/// </returns>
protected override Task<AdsErrorCode> ReadDeviceInfoIndicationAsync(AmsAddress sender, uint invokeId, CancellationToken cancel)
{
    LogTrace("ReadDeviceINfoIndication(Address:{0}, ID: {1})", sender, invokeId);
    return ReadDeviceInfoResponseAsync(sender, // requester's AMS address
                                             invokeId, // invoke id provided by requester
                                             AdsErrorCode.NoError, // ADS error code
                                             "C#_TestServer", // name of this server
                                             s_version, // version of this server
                                             cancel); // Cancellation Token
}

/// Called when an ADS Write indication is received.
/// </summary>
/// <remarks>
/// Overwrite this method in derived classes to react on ADS Write indications.
/// The default implementation replies with an ADS ServiceNotSupported error code (0x701).
/// </remarks>
/// <param name="sender">The sender's AMS address</param>
/// <param name="invokeId">The invokeId provided by the sender</param>
/// <param name="indexGroup">The index group of the requested ADS service</param>
/// <param name="indexOffset">The index offset of the requested ADS service</param>
/// <param name="writeData">The data to be written</param>
/// <param name="cancel">The cancellation token.</param>
/// <returns>A task that represents the asynchronous <see cref="WriteIndicationAsync" /> operation. The <see cref="Task(Task).Result"/> parameter contains the <see cref="AdsErrorCode"/> as
/// <see cref="Task(Task).Result"/>.
/// </returns>
protected override Task<AdsErrorCode> WriteIndicationAsync(AmsAddress sender, uint invokeId, uint indexGroup, uint indexOffset, ReadOnlyMemory<byte> writeData, CancellationToken cancel)
{
    AdsErrorCode adsError = AdsErrorCode.NoError;
    LogTrace("WriteIndicationAsync(Address:(0), ID:{1}, IG:{2}, IO:{3}, Length: {4})", sender, invokeId, indexGroup, indexOffset, writeData.Length);
    switch (indexGroup) /* use index group (and offset) to distinguish between the services
    of this server */
    {
        case 0x10000:
            if (writeData.Length == 4)
            {
                writeData.CopyTo(_dataBuffer);
            }
            else
            {
                adsError = AdsErrorCode.DeviceInvalidParam;
            }
            break;
        case 0x20000: /* used for the PLC Sample */
            if (writeData.Length == 4)
            {
                break;
            }
    }
uint value = BinaryPrimitives.ReadUInt32LittleEndian(writeData.Span);
LogInformation(String.Format("PLC Counter: {0}"; value));
}
break;
default: /* other services are not supported */
adsError = AdsErrorCode.DeviceServiceNotSupported;
break;
}
// Send a response to the requester
return WriteResponseAsync(sender, // requester's AMS address
invokeId, // invoke id provided by requester
adsError, // ADS error code
cancel);
}
/// <summary>
/// Called when an ADS Read indication is received.
/// </summary>
/// <returns>A task that represents the asynchronous <see cref="ReadIndicationAsync"/> operation. The <see cref="Task{T}"/> parameter contains the <see cref="AdsErrorCode"/> as <see cref="Task{Task}.Result"/>.</returns>
protected override Task<AdsErrorCode> ReadIndicationAsync(AmsAddress sender, uint invokeId, uint indexGroup, uint indexOffset, int readLength, CancellationToken cancel) {
LogTrace("ReadIndicationAsync(Address:{0}, ID:{1}, IG:{2}, IO:{3}, Length:{4})", sender, invokeId, indexGroup, indexOffset, readLength);
/* Distinguish between services like in AdsWriteInd */
// Send a response to the requester
return ReadResponseAsync(sender, // requester's AMS address
invokeId, // invoke id provided by requester
AdsErrorCode.NoError, // ADS error code
dataBuffer.AsMemory(), // data buffer
cancel);
}
/// <summary>
/// Called when an ADS Read State indication is received.
/// </summary>
/// <returns>A task that represents the asynchronous <see cref="ReadDeviceStateIndicationAsync"/> operation. The <see cref="Task{T}"/> parameter contains the <see cref="AdsErrorCode"/> as <see cref="Task{Task}.Result"/>.</returns>
protected override Task<AdsErrorCode> ReadDeviceStateIndicationAsync(AmsAddress sender, uint invokeId, CancellationToken cancel) {
LogTrace("ReadDeviceStateIndicationAsync(Address:{0}, ID:{1}, IG:{2}, IO:{3}, Length:{4})", sender, invokeId, indexGroup, indexOffset, readLength);
/* Distinguish between services like in AdsWriteInd */
return ReadDeviceStateResponseAsync(sender, // requester's AMS address
invokeId, // invoke id provided by requester
AdsErrorCode.NoError, // ADS error code
_deviceState, // device state
cancel);
/// <summary>
/// Overwrite this method in derived classes to react on ADS Write Control indications.
/// The default implementation replies with an ADS ServiceNotSupported error code (0x701).
/// </summary>
/// <param name="sender">The sender's AMS address</param>
/// <param name="invokeId">The invokeId provided by the sender</param>
/// <param name="deviceState">The requested new device state of this ADS device</param>
/// <param name="cbLength">The length in bytes of the additional data buffer</param>
/// <param name="data">An additional data buffer of cbLength bytes</param>
/// <param name="cancel">The cancellation token</param>
protected override Task<AdsErrorCode> WriteControlIndicationAsync(AmsAddress sender, uint invokeId, AdsState adsState, ushort deviceState, ReadOnlyMemory<byte> data, CancellationToken cancel) {
    LogTrace("WriteControlIndication(Address:{0}, ID:{1}, AdsState:{2}, DeviceState:{3}, Length:{4})", sender, invokeId, adsState, deviceState, data.Length);
    // Set requested ADS and device status
    _adsState = adsState;
    _deviceState = deviceState;
    // Send a response to the requester
    return WriteControlResponseAsync(sender, invokeId, AdsErrorCode.NoError, cancel);
}

/// <summary>
/// Called when an ADS Add Device Notification indication is received.
/// </summary>
/// <param name="sender">The sender's AMS address</param>
/// <param name="invokeId">The invokeId provided by the sender</param>
/// <param name="indexGroup">The index group of the requested ADS service</param>
/// <param name="indexOffset">The index offset of the requested ADS service</param>
/// <param name="dataLength">Number of bytes to be transmitted</param>
/// <param name="settings">The Notification settings.</param>
/// <param name="cancel">The cancellation token.</param>
protected override Task<AdsErrorCode> AddDeviceNotificationIndicationAsync(AmsAddress sender, uint invokeId, uint indexGroup, uint indexOffset, int dataLength, NotificationSettings settings, CancellationToken cancel) {
    LogTrace("AddDeviceNotificationIndication(Address:{0}, ID:{1}, IG:{2}, IO:{3}, Length:{4})", sender, invokeId, indexGroup, indexOffset, dataLength);
    /* Create a new notification entry an store it in the notification table */
    NotificationRequestEntry notEntry = new NotificationRequestEntry(sender, indexGroup, indexOffset, dataLength, settings);
    _notificationTable.AddOrUpdate(_currentNotificationHandle, notEntry,(key,value) => notEntry);
    _currentNotificationHandle++;
    // Send a response to the requester
    return AddDeviceNotificationResponseAsync(sender, invokeId, AdsErrorCode.NoError, _currentNotificationHandle++, cancel);
}

/// <summary>
/// Called when an ADS Delete Device Notification indication is received.
/// </summary>
/// <param name="sender">The sender's AMS address</param>
/// <param name="invokeId">The invokeId provided by the sender</param>
/// <param name="hNotification">The notification handle to be deleted</param>
/// <param name="cancel">The cancellation token.</param>
/// <returns>A task that represents the asynchronous <see cref="DeleteDeviceNotificationIndicationAsync" /> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as <see cref="Task{Task}.Result" />.
protected override Task<AdsErrorCode> DeleteDeviceNotificationIndicationAsync(AmsAddress sender, uint invokeId, uint hNotification, CancellationToken cancel) {
    LogTrace("DeleteDeviceNotification(Address:{0}, ID:{1}, Handle: {2}");sender, invokeId, hNotification);
    AdsErrorCode errorCode = AdsErrorCode.NoError;
    /* check if the requested notification handle is still in the notification table */
    if (_notificationTable.ContainsKey(hNotification)) {
        NotificationRequestEntry entry = null;
        _notificationTable.TryRemove(hNotification, out entry);
    } else { // notification handle is not in the notification table -> return an error code
        errorCode = AdsErrorCode.DeviceNotifyHandleInvalid;
    }
    // Send a response to the requester
    return DeleteDeviceNotificationResponseAsync(sender, invokeId, errorCode, cancel);
}

/// <summary>
/// Called when an ADS Device Notification indication is received.
/// </summary>
/// <remarks>
/// Overwrite this method in derived classes to react on ADS Device Notification indications.
/// </remarks>
/// <param name="sender">The sender's AMS address</param>
/// <param name="invokeId">The invokeId provided by the sender</param>
/// <param name="numStampHeaders">The number of ADS Stamp Headers contained in stampHeaders</param>
/// <param name="stampHeaders">The array of received ADS Stamp Headers.</param>
/// <param name="cancel">The cancellation token.</param>
/// <returns>A task that represents the asynchronous <see cref="DeviceNotificationIndicationAsync(AmsAddress, uint, uint, NotificationSamplesStamp[], CancellationToken)" /> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as <see cref="Task{Task}.Result" />.
protected override Task<AdsErrorCode> DeviceNotificationIndicationAsync(AmsAddress sender, uint invokeId, uint numStampHeaders, NotificationSamplesStamp[] stampHeaders, CancellationToken cancel) {
    LogTrace("DeviceNotificationIndication(Address:{0}, ID:{1}, NumStampHeaders: {2}");sender, invokeId, numStampHeaders);
    LogInformation("Received Device Notification Request");
    /*
    * Call notification handlers.
    */
    return Task.FromResult(AdsErrorCode.Succeeded);
}

/// <summary>
/// Called when an ADS Read Write indication is received.
/// </summary>
/// <remarks>
/// Overwrite this method in derived classes to react on ADS Read Write indications.
/// The default implementation replies with an ADS ServiceNotSupported error code (0x701).
/// </remarks>
/// <param name="sender">The sender's AMS address</param>
/// <param name="invokeId">The invokeId provided by the sender</param>
/// <param name="indexGroup">The index group of the requested ADS service</param>
/// <param name="indexOffset">The index offset of the requested ADS service</param>
/// <param name="readLength">Number of bytes to be read</param>
/// <param name="writeData">The data to be written</param>
/// <param name="cancel">The cancellation token.</param>
/// <returns>A task that represents the asynchronous 'ReadWriteIndication' operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as <see cref="Task{Task}.Result" />.
protected async override Task<AdsErrorCode> ReadWriteIndicationAsync(AmsAddress sender, uint invokeId, uint indexGroup, uint indexOffset, byte[] readData, byte[] writeData, CancellationToken cancel) {
    LogTrace("ReadWriteIndication(Address:{0}, ID:{1}, IndexGroup: {2}, IndexOffset: {3}, ReadLength: {4}, WriteData: {5}, Cancel: {6});
    LogInformation("Sending Read Write Request");
    /*
    * Call read/write handlers.
    */
    return Task.FromResult(AdsErrorCode.Succeeded);
}
LogTrace("ReadWriteIndication(Address:{0}, ID:{1}, IG:{2}, IO:{3}, ReadLen:{4}, WriteLen:
(5))", sender, invokeId, indexGroup, indexOffset, readLength, writeData.Length);
/* Distinguish between services like in AdsWriteInd */
// Send a response to the requester
AdsErrorCode errorCode = AdsErrorCode.None;
if (readLength == 4 && writeData.Length == 4)
{
    errorCode = await ReadWriteResponseAsync(sender, // requester's AMS address
        invokeId, // invoke id provided by requester
        AdsErrorCode.NoError, // ADS error code
        _dataBuffer.AsMemory(), cancel).ConfigureAwait(false);
    writeData.CopyTo(_dataBuffer.AsMemory(0, 4));
} else
{
    errorCode = await ReadWriteResponseAsync(sender, // requester's AMS address
        invokeId, // invoke id provided by requester
        AdsErrorCode.DeviceInvalidSize, // ADS error code
        Memory<byte>.Empty, cancel).ConfigureAwait(false);
}
return errorCode;
/* Overwrite the confirmation methods of the TcAdsServer class for the requests your ADS server
* sends. They are called upon incoming responses. These sample implementations only add a log
* message to the sample form.
*/

/// <summary>
/// Called when an ADS Read State confirmation is received.
/// </summary>
/// <remarks>
/// Overwrite this method in derived classes to react on ADS Read State confirmations.
/// </remarks>
/// <param name="sender">The sender's AMS address</param>
/// <param name="invokeId">The invoke id provided by this server during the corresponding reques
/// t</param>
/// <param name="result">The ADS error code provided by the sender</param>
/// <param name="adsState">The ADS state of the sender</param>
/// <param name="deviceState">The device state of the sender</param>
/// <param name="cancel">The cancellation token.</param>
/// <returns>A task that represents the asynchronous <see cref="ReadDeviceStateConfirmationAsync"
 /> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as 
/// a task</returns>
protected override Task<AdsErrorCode> ReadDeviceStateConfirmationAsync(AmsAddress sender, uint i
vokeId, AdsErrorCode result, AdsState adsState, ushort deviceState, CancellationToken cancel)
{
    LogTrace("ReadDeviceStateConfirmation(Address:{0}, ID:{1}, Result:{2}, AdsState:
(3), DeviceState:{4})", sender, invokeId, result, adsState, deviceState);
    LogInformation("Received Read State Confirmation");
    return Task.FromResult(AdsErrorCode.Succeeded);
}

/// <summary>
/// Called when an ADS Read confirmation is received.
/// </summary>
/// <remarks>
/// Overwrite this method in derived classes to react on ADS Read confirmations.
/// </remarks>
/// <param name="sender">The sender's AMS address</param>
/// <param name="invokeId">The invoke id provided by this server during the corresponding reques
/// t</param>
/// <param name="result">The ADS error code provided by the sender</param>
/// <param name="cbLength">The number of read bytes</param>
/// <param name="data">The read data buffer</param>
/// <param name="cancel">The cancellation token.</param>
/// <returns>A task that represents the asynchronous <see cref="ReadConfirmationAsync"/>
/// operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as 
/// a task</returns>
protected override Task<AdsErrorCode> ReadConfirmationAsync(AmsAddress sender, uint invokeId, Ad
sErrorCode result, ReadOnlyMemory<byte> readData, CancellationToken cancel)
TwinCAT.Ads Namespaces

LogTrace("ReadConfirmation(Address:{0}, ID:{1}, IG:{2}, Result:{3}, Length:{4})", sender, invokeId, result, readData.Length);
LogInformation("Received Read Confirmation");
return Task.FromResult(AdsErrorCode.Succeeded);
}

/// <summary>
/// Called when an ADS Write confirmation is received.
/// </summary>
/// <param name="sender">The sender's AMS address</param>
/// <returns></returns>
protected override Task<AdsErrorCode> WriteConfirmationAsync(AmsAddress sender, uint invokeId, AdsErrorCode result, CancellationToken cancel)
{
LogTrace("WriteConfirmation(Address:{0}, ID:{1}, Result:{2})", sender, invokeId, result);
LogInformation("Received Write Confirmation");
return Task.FromResult(AdsErrorCode.Succeeded);
}

/// <summary>
/// Called when an ADS Read Device Info confirmation is received.
/// </summary>
/// <param name="target">The sender's AMS address</param>
/// <returns></returns>
protected override Task<AdsErrorCode> ReadDeviceInfoConfirmationAsync(AmsAddress sender, uint invokeId, AdsErrorCode result, string name, AdsVersion version, CancellationToken cancel)
{
LogTrace("ReadDeviceInfoConfirmation(Address:{0}, ID:{1}, Result:{2}, Name:{3}, Version:{4})", sender, invokeId, result, name, version);
LogInformation("Received Read Device Info Confirmation");
return Task.FromResult(AdsErrorCode.Succeeded);
}

/// <summary>
/// Called when an ADS Write Control confirmation is received.
/// </summary>
/// <param name="sender">The sender's AMS address</param>
/// <returns></returns>
protected override Task<AdsErrorCode> WriteControlConfirmationAsync(AmsAddress sender, uint invokeId, AdsErrorCode result, CancellationToken cancel)
{
LogTrace("WriteControlConfirmation(Address:{0}, ID:{1}, Result:{2})", sender, invokeId, result);
LogInformation("Received Write Control Confirmation");
return Task.FromResult(AdsErrorCode.Succeeded);
}
/// <summary>
/// Called when an ADS Add Device Notification confirmation is received.
/// </summary>
/// <remarks>
/// Overwrite this method in derived classes to react on ADS Add Device Notification confirmations.
/// </remarks>
/// <param name="sender">The sender's AMS address</param>
/// <param name="invokeId">The invoke id provided by this server during the corresponding request</param>
/// <param name="result">The ADS error code provided by the sender</param>
/// <param name="cancel">The cancellation token</param>
/// <returns>A task that represents the asynchronous \(<see cref="AddDeviceNotificationConfirmationAsync\)> operation. The \(<see cref="Task{Task}\)> parameter contains the \(<see cref="AdsErrorCode\> as \(<see cref="Task{Task}.Result\> result.</returns>
protected override Task<AdsErrorCode> AddDeviceNotificationConfirmationAsync(AmsAddress sender, uint invokeId, AdsErrorCode result, uint notificationHandle, CancellationToken cancel)
{
    _serverLogger.ServerNotificationHandle = notificationHandle;
    LogTrace("AddDeviceNotificationConfirmation(Address:{0}, ID:{1}, Result:{2}, Handle:{3})", sender, invokeId, result, notificationHandle);
    LogInformation("Received Add Device Notification Confirmation. Notification handle: " + notificationHandle);
    return Task.FromResult(AdsErrorCode.Succeeded);
}

/// <summary>
/// Called when an ADS Delete Device Notification confirmation is received.
/// </summary>
/// <remarks>
/// Overwrite this method in derived classes to react on ADS Delete Device Notification confirmations.
/// </remarks>
/// <param name="sender">The sender's AMS address</param>
/// <param name="invokeId">The invoke id provided by this server during the corresponding request</param>
/// <param name="result">The ADS error code provided by the sender</param>
/// <param name="cancel">The cancellation token</param>
/// <returns>A task that represents the asynchronous \(<see cref="DeleteDeviceNotificationConfirmationAsync\)> operation. The \(<see cref="Task{Task}\)> parameter contains the \(<see cref="AdsErrorCode\> as \(<see cref="Task{Task}.Result\> result.</returns>
protected override Task<AdsErrorCode> DeleteDeviceNotificationConfirmationAsync(AmsAddress sender, uint invokeId, AdsErrorCode result, CancellationToken cancel)
{
    LogTrace("DeleteDeviceNotificationConfirmation(Address:{0}, ID:{1}, Result:{2})", sender, invokeId, result);
    LogInformation("Received Delete Device Notification Confirmation");
    return Task.FromResult(AdsErrorCode.Succeeded);
}

/// <summary>
/// Called when an ADS Read Write confirmation is received.
/// </summary>
/// <remarks>
/// Overwrite this method in derived classes to react on ADS Read Write confirmations.
/// </remarks>
/// <param name="sender">The sender's AMS address</param>
/// <param name="invokeId">The invoke id provided by this server during the corresponding request</param>
/// <param name="result">The ADS error code provided by the sender</param>
/// <param name="cbLength">The number of read bytes</param>
/// <param name="data">The read data buffer</param>
/// <param name="cancel">The cancellation token</param>
/// <returns>A task that represents the asynchronous \(<see cref="ReadWriteConfirmationAsync\)> operation. The \(<see cref="Task{Task}\)> parameter contains the \(<see cref="AdsErrorCode\> as \(<see cref="Task{Task}.Result\> result.</returns>
protected override Task<AdsErrorCode> ReadWriteConfirmationAsync(AmsAddress sender, uint invokeId, AdsErrorCode result, ReadOnlyMemory<byte> readData, CancellationToken cancel)
{
    LogTrace("ReadWriteConfirmationConfirmation(Address:{0}, ID:{1}, Result:{2}, Length:{3})", sender, invokeId, result, readData.Length);
    LogInformation("Received Read Write Confirmation");
    return Task.FromResult(AdsErrorCode.Succeeded);
// <summary>
// AdsSampleServer Notification request entry
// </summary>
internal class NotificationRequestEntry
{
    private AmsAddress _rAddr; // the AmsNetId of the requester
    private uint _indexGroup; // the requested index group
    private uint _indexOffset; // the requested index offset
    private int _cbLength; // the number of bytes to send
    NotificationSettings _settings; // the notification settings

    internal NotificationRequestEntry(AmsAddress rAddr,
        uint indexGroup,
        uint indexOffset,
        int cbLength,
        NotificationSettings settings)
    {
        _rAddr = rAddr;
        _indexGroup = indexGroup;
        _indexOffset = indexOffset;
        _cbLength = cbLength;
        _settings = settings;
    }
}

Reference

TwinCAT.Ads.Server Namespace [1121]

6.4.1.1 AdsServer Properties

The AdsServer [1121] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmsServer [1135]</td>
<td>Gets the the internal AmsServer [1135] object.</td>
</tr>
<tr>
<td>IsConnected [1136]</td>
<td>Gets a value indicating whether AdsServer [1121] is connected.</td>
</tr>
<tr>
<td>IsDisconnecting</td>
<td>Indicates, that the AdsServer [1121] is actually disconnecting.</td>
</tr>
<tr>
<td>IsDisposed [1137]</td>
<td>Gets a value indicating whether this instance is disposed.</td>
</tr>
<tr>
<td>Logger [1137]</td>
<td>Gets the logger object.</td>
</tr>
<tr>
<td>ServerAddress [1138]</td>
<td>The AMS address of this server.</td>
</tr>
</tbody>
</table>

Reference

AdsServer Class [1121]

TwinCAT.Ads.Server Namespace [1121]

6.4.1.1 AdsServer.AmsServer Property

 Gets the the internal AmsServer object.
Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected AmsServer AmsServer { get; }

Property Value

Type: AmsServer
The ams server.

Reference

AdsServer Class
TwinCAT.Ads.Server Namespace

6.4.1.1.2 AdsServer.IsConnected Property

Gets a value indicating whether AdsServer is connected.

Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool IsConnected { get; }

Property Value

Type: Boolean
true if this instance is connected; otherwise, false.

Reference

AdsServer Class
TwinCAT.Ads.Server Namespace

6.4.1.1.3 AdsServer.IsDisconnecting Property

Indicates, that the AdsServer is actually disconnecting.

Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected bool IsDisconnecting { get; }
Property Value
Type: Boolean

Reference
AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.1.4 AdsServer.IsDisposed Property
Gets a value indicating whether this instance is disposed.

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
public bool IsDisposed { get; }

Property Value
Type: Boolean
true if this instance is disposed; otherwise, false.

Reference
AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.1.5 AdsServer.Logger Property
Gets the logger object.

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
protected ILogger Logger { get; }

Property Value
Type: ILogger
The logger.

Reference
AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]
6.4.1.6 AdsServer.ServerAddress Property

The AMS address of this server.

**Namespace:** TwinCAT.Ads.Server [1121]
**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public virtual AmsAddress ServerAddress { get; }
```

**Property Value**

Type: AmsAddress [648]

**Remarks**

The Address consists of AmsNetId [665] and AmsPort [693]. While the AmsNetId [665] is defined by the System, where the AmsTcpIPRouter is running, the AmsPort [693] is specified by the AdsServer [1121] constructor. By default, the router is running on the same system, but can be configured by RouterEndPoint [1546].

**Reference**

- AdsServer Class [1121]
- TwinCAT.Ads.Server Namespace [1121]
- TwinCAT.Ads.AmsNetId [665]
- TwinCAT.Ads.AmsPort [693]
- AmsConfiguration.RouterEndPoint [1546]

6.4.1.2 AdsServer Methods

The AdsServer [1121] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="1142" alt="AddDeviceNotificationConfirmationAsync" /></td>
<td>Called when an ADS Add Device Notification confirmation is received.</td>
</tr>
<tr>
<td><img src="1143" alt="AddDeviceNotificationIndicationAsync" /></td>
<td>Called when an ADS Add Device Notification indication is received.</td>
</tr>
<tr>
<td><img src="1144" alt="AddDeviceNotificationRequest" /></td>
<td>Sends an ADS Add Device Notification request (synchronous).</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>AddDeviceNotificationRequestAsync [1144]</td>
<td>Sends an ADS Add Device Notification request (async)</td>
</tr>
<tr>
<td>ConnectServer [1146]</td>
<td>Connect this ADS server to the local ADS router.</td>
</tr>
<tr>
<td>ConnectServerAndWaitAsync [1147]</td>
<td>Registers the AdsServer [1121] at the router asynchronously.</td>
</tr>
<tr>
<td>DeleteDeviceNotificationConfirmationAsync [1147]</td>
<td>Called when an ADS Delete Device Notification confirmation is received.</td>
</tr>
<tr>
<td>DeleteDeviceNotificationIndicationAsync [1148]</td>
<td>Called when an ADS Delete Device Notification indication is received.</td>
</tr>
<tr>
<td>DeleteDeviceNotificationRequest [1149]</td>
<td>Sends an ADS Delete Device Notification request (synchronous).</td>
</tr>
<tr>
<td>DeleteDeviceNotificationRequestAsync [1149]</td>
<td>Sends an ADS Delete Device Notification request (async).</td>
</tr>
<tr>
<td>DeviceNotificationRequestAsync [1151]</td>
<td>Sends an ADS Device Notification request asynchronously</td>
</tr>
<tr>
<td>DeviceNotificationRequestSync [1152]</td>
<td>Sends an ADS Device Notification request (sync)</td>
</tr>
<tr>
<td>Disconnect [1152]</td>
<td>Disconnects this ADS server from the local ADS router.</td>
</tr>
<tr>
<td>Dispose [1153]</td>
<td>Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.</td>
</tr>
<tr>
<td>Dispose(Boolean) [1154]</td>
<td>Releases unmanaged and - optionally - managed resources.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>OnAddDeviceNotificationConfirmationAsync</td>
<td>Called when Add device Notification Confirmation is send.</td>
</tr>
<tr>
<td>OnConnected</td>
<td>Handler function that is called, when the <code>AdsServer</code> is connected.</td>
</tr>
<tr>
<td>OnDisconnect</td>
<td>Called when the <code>AdsServer</code> is about to be disconnected.</td>
</tr>
<tr>
<td>OnRouterNotification</td>
<td>Handler Function for a Router Notification.</td>
</tr>
<tr>
<td>OnServerConnectionStateChanged</td>
<td>Handles the <code>ServerConnectionStateChanged</code> event.</td>
</tr>
<tr>
<td>ReadConfirmationAsync</td>
<td>Called when an ADS Read Device Info confirmation is received.</td>
</tr>
<tr>
<td>ReadDeviceInfoConfirmationAsync</td>
<td>Called when an ADS Read Device Info indication is received.</td>
</tr>
<tr>
<td>ReadDeviceInfoIndicationAsync</td>
<td>Sends an ADS Read Device Info request asynchronously</td>
</tr>
<tr>
<td>ReadDeviceInfoRequestAsync</td>
<td>Sends an ADS Read Device Info request synchronously.</td>
</tr>
<tr>
<td>ReadDeviceInfoResponseAsync</td>
<td>Sends an ADS Read Device Info response.</td>
</tr>
<tr>
<td>ReadDeviceStateConfirmationAsync</td>
<td>Called when an ADS Read State confirmation is received.</td>
</tr>
<tr>
<td>ReadDeviceStateIndicationAsync</td>
<td>Called when an ADS Read State indication is received.</td>
</tr>
<tr>
<td>ReadDeviceStateRequestAsync</td>
<td>Sends an ADS Read State request (asynchronous)</td>
</tr>
<tr>
<td>ReadDeviceStateRequestSync</td>
<td>Sends an ADS Read State request (synchronous)</td>
</tr>
<tr>
<td>ReadDeviceStateResponseAsync</td>
<td>Sends an ADS Read State response.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadIndicationAsync</td>
<td>Called when an ADS Read indication is received.</td>
</tr>
<tr>
<td>ReadRequest</td>
<td>Sends an ADS Read Request.</td>
</tr>
<tr>
<td>ReadRequestAsync</td>
<td>Sends an ADS Read Request asynchronously.</td>
</tr>
<tr>
<td>ReadResponseAsync</td>
<td></td>
</tr>
<tr>
<td>ReadWriteConfirmationAsync</td>
<td></td>
</tr>
<tr>
<td>ReadWriteIndicationAsync</td>
<td></td>
</tr>
<tr>
<td>ReadWriteRequestAsync</td>
<td></td>
</tr>
<tr>
<td>ReadWriteRequestSync</td>
<td></td>
</tr>
<tr>
<td>ReadWriteResponseAsync</td>
<td></td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>WriteConfirmationAsync</td>
<td>Called when an ADS Write confirmation is received. Overwrite this method in</td>
</tr>
<tr>
<td>WriteControlConfirmationAsync</td>
<td>derived classes to react on ADS Write confirmations.</td>
</tr>
<tr>
<td>WriteControlIndicationAsync</td>
<td></td>
</tr>
<tr>
<td>WriteControlRequest</td>
<td></td>
</tr>
<tr>
<td>WriteControlRequestAsync</td>
<td></td>
</tr>
<tr>
<td>WriteControlResponse</td>
<td>Sends an ADS Write Control response.</td>
</tr>
<tr>
<td>WriteIndicationAsync</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>WriteRequest</td>
<td></td>
</tr>
<tr>
<td>WriteRequestAsync</td>
<td></td>
</tr>
<tr>
<td>WriteResponseAsync</td>
<td>Sends an ADS Write response.</td>
</tr>
</tbody>
</table>

**Reference**

AdsServer Class [1121]

TwinCAT.Ads.Server Namespace [1121]

### 6.4.1.2.1 AdsServer.AddDeviceNotificationConfirmationAsync Method

Called when an ADS Add Device Notification confirmation is received.

**Namespace:** TwinCAT.Ads.Server [1121]

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected virtual Task<AdsErrorCode> AddDeviceNotificationConfirmationAsync(
    AmsAddress sender,
    uint invokeId,
    AdsErrorCode result,
    uint notificationHandle,
    CancellationToken cancel)
```

**Parameters**

- **sender**
  - Type: TwinCAT.Ads.AmsAddress [648]
  - The sender's AMS address

- **invokeId**
  - Type: System.UInt32
  - The invoke id provided by this server during the corresponding request

- **result**
  - Type: TwinCAT.Ads.AdsErrorCode [575]
  - The ADS error code provided by the sender

- **notificationHandle**
  - Type: System.UInt32
  - The notification handle provided by the sender

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

**Return Value**

Type: Task<AdsErrorCode> [575].
A task that represents the asynchronous AddDeviceNotificationConfirmationAsync(AmsAddress, UInt32, AdsErrorCode, UInt32, CancellationToken) operation. The Task.TResult parameter contains the AdsErrorCode [575] as Result.

**Remarks**

Overwrite this method in derived classes to react on ADS Add Device Notification confirmations.
6.4.1.2.2 AdsServer.AddDeviceNotificationIndicationAsync Method

Called when an ADS Add Device Notification indication is received.

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```c#
protected virtual Task<AdsErrorCode> AddDeviceNotificationIndicationAsync(
    AmsAddress sender,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int dataLength,
    NotificationSettings settings,
    CancellationToken cancel
)
```

**Parameters**

- **sender**
  - Type: `TwinCAT.Ads.AmsAddress`
  - The sender's AMS address

- **invokeId**
  - Type: `System.UInt32`
  - The invokeId provided by the sender

- **indexGroup**
  - Type: `System.UInt32`
  - The index group of the requested ADS service

- **indexOffset**
  - Type: `System.UInt32`
  - The index offset of the requested ADS service

- **dataLength**
  - Type: `System.Int32`
  - Number of bytes to be transmitted

- **settings**
  - Type: `TwinCAT.Ads.NotificationSettings`
  - The Notification settings.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.

**Return Value**

Type: `Task<AdsErrorCode>`

A task that represents the asynchronous 'AddDeviceNotificationIndication' operation. The `Task<TResult>` parameter contains the `AdsErrorCode` as `Result`.

**Remarks**

Overwrite this method in derived classes to react on ADS Add Device Notification indications. The default implementation replies with an ADS ServiceNotSupported error code (0x701).

**Reference**

AdsServer Class [1121]
**6.4.1.2.3 AdsServer.AddDeviceNotificationRequest Method**

Sends an ADS Add Device Notification request (synchronous).

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
protected AdsErrorCode AddDeviceNotificationRequest(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int dataLength,
    NotificationSettings settings
)
```

**Parameters**

- **target**  
  Type: TwinCAT.Ads.AmsAddress
  The receiver's AMS address

- **invokeId**  
  Type: System.UInt32
  The invokeId for this call. Used to find the matching ADS Confirmation

- **indexGroup**  
  Type: System.UInt32
  The index group of the requested ADS service

- **indexOffset**  
  Type: System.UInt32
  The index group of the requested ADS service

- **dataLength**  
  Type: System.Int32
  The number of bytes to be transmitted

- **settings**  
  Type: TwinCAT.Ads.NotificationSettings
  The notification settings.

**Return Value**

Type: AdsErrorCode
The ADS error code for this call.

**Reference**

- AdsServer Class
- TwinCAT.Ads.Server Namespace

**6.4.1.2.4 AdsServer.AddDeviceNotificationRequestAsync Method**

Sends an ADS Add Device Notification request (async)

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
protected Task<AdsErrorCode> AddDeviceNotificationRequestAsync(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int dataLength,
    NotificationSettings settings,
    CancellationToken cancel
)
```

Parameters

target
Type: TwinCAT.Ads.AmsAddress
The receiver's AMS address

invokeId
Type: System.UInt32
The invokeId for this call. Used to find the matching ADS Confirmation

indexGroup
Type: System.UInt32
The index group of the requested ADS service

indexOffset
Type: System.UInt32
The index group of the requested ADS service

dataLength
Type: System.Int32
The number of bytes to be transmitted

settings
Type: TwinCAT.Ads.NotificationSettings
The notification settings.

cancel
Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<AdsErrorCode>
A task that represents the asynchronous AddDeviceNotificationRequestAsync(AmsAddress, UInt32, UInt32, UInt32, Int32, NotificationSettings, CancellationToken) operation. The Task.TResult parameter contains the AdsErrorCode as Result.

Reference

AdsServer Class
TwinCAT.Ads.Server Namespace

6.4.1.2.5 AdsServer.AddDeviceNotificationResponseAsync Method

 Sends an ADS Add Device Notification response.

Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected Task<AdsErrorCode> AddDeviceNotificationResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
)
```
uint handle,
CancellationToken cancel
)

Parameters

target Type: TwinCAT.Ads.AmsAddress
The receiver’s AMS address

invokeld Type: System.UInt32
The invoke ID provided by the receiver

result Type: TwinCAT.Ads.AdsErrorCode
The ADS error code for the response

handle Type: System.UInt32
The notification handle for the added notification

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task.TwinCAT.Ads.AdsErrorCode
A task that represents the asynchronous AddDeviceNotificationResponseAsync(AmsAddress, UInt32, AdsErrorCode, UInt32, CancellationToken) operation. The Task.TResult parameter contains the AdsErrorCode as Result.

Reference

AdsServer Class
TwinCAT.Ads.Server Namespace

6.4.1.2.6 AdsServer.ConnectServer Method

Connect this ADS server to the local ADS router.

Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public virtual uint ConnectServer()

Return Value

Type: UInt32
The AmsServer Port.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsServerException</td>
<td>The connect call has failed!</td>
</tr>
</tbody>
</table>

Reference

AdsServer Class
6.4.1.2.7 AdsServer.ConnectServerAndWaitAsync Method

Registers the AdsServer at the router asynchronously.

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Task<AdsErrorCode> ConnectServerAndWaitAsync(
    CancellationToken cancel)
```

**Parameters**

`cancel` Type: `System.Threading.CancellationToken`

**Return Value**

Type: `Task<AdsErrorCode>`

Returns a task object that represents the ConnectServerAndWaitAsync(CancellationToken) operation which returns an AdsErrorCode as result.

**Remarks**

The connection is hold until a cancel is requested, that means this method will wait until disconnect.

**Reference**

AdsServer Class

TwinCAT.Ads.Server Namespace

6.4.1.2.8 AdsServer.DeleteDeviceNotificationConfirmationAsync Method

Called when an ADS Delete Device Notification confirmation is received.

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected virtual Task<AdsErrorCode> DeleteDeviceNotificationConfirmationAsync(
    AmsAddress sender,
    uint invokeId,
    AdsErrorCode result,
    CancellationToken cancel)
```
Parameters

sender Type: TwinCAT.Ads.AmsAddress [648]
The sender’s AMS address

invokeId Type: System.UInt32
The invoke id provided by this server during the corresponding request

result Type: TwinCAT.Ads.AdsErrorCode [575]
The ADS error code provided by the sender

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<AdsErrorCode [575]>
A task that represents the asynchronous DeleteDeviceNotificationConfirmationAsync(AmsAddress, UInt32, AdsErrorCode, CancellationToken) operation. The Task.TResult parameter contains the AdsErrorCode [575] as Result.

Remarks

Overwrite this method in derived classes to react on ADS Delete Device Notification confirmations.

Reference

AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.9 AdsServer.DeleteDeviceNotificationIndicationAsync Method

Called when an ADS Delete Device Notification indication is received.

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

protected virtual Task<AdsErrorCode> DeleteDeviceNotificationIndicationAsync(
    AmsAddress sender,
    uint invokeId,
    uint hNotification,
    CancellationToken cancel
)

Parameters

sender Type: TwinCAT.Ads.AmsAddress [648]
The sender’s AMS address

invokeId Type: System.UInt32
The invoke id provided by this server during the corresponding request

hNotification Type: System.UInt32
The notification handle to be deleted

cancel Type: System.Threading.CancellationToken
The cancellation token.
Return Value

Type: Task.AdsErrorCode [575].
A task that represents the asynchronous DeleteDeviceNotificationIndicationAsync(AmsAddress, UInt32, UInt32, CancellationToken) operation. The Task.TResult parameter contains the AdsErrorCode [575] as Result.

Remarks

Overwrite this method in derived classes to react on ADS Delete Device Notification indications. The default implementation replies with an ADS ServiceNotSupported error code (0x701).

Reference

AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.10 AdsServer.DeleteDeviceNotificationRequest Method

Sends an ADS Delete Device Notification request (synchronous).

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

protected AdsErrorCode DeleteDeviceNotificationRequest(
    AmsAddress target,
    uint invokeId,
    uint hNotification)

Parameters

target Type: TwinCAT.Ads.AmsAddress [648]
The receiver's AMS address

invokeId Type: System.UInt32
The invokeId for this call. Used to find the matching ADS Confirmation

hNotification Type: System.UInt32
The notification ID to be deleted

Return Value

Type: AdsErrorCode [575]
The ADS error code for this call.

Reference

AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.11 AdsServer.DeleteDeviceNotificationRequestAsync Method

Sends an ADS Delete Device Notification request (async).
Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected Task<AdsErrorCode> DeleteDeviceNotificationRequestAsync(
    AmsAddress target,
    uint invokeId,
    uint hNotification,
    CancellationToken cancel)
```

Parameters

target Type: TwinCAT.Ads.AmsAddress [648]
The receiver's AMS address
invokeId Type: System.UInt32
The invokeId for this call. Used to find the matching ADS Confirmation
hNotification Type: System.UInt32
The notification ID to be deleted
cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<AdsErrorCode> [575]
A task that represents the asynchronous DeleteDeviceNotificationRequestAsync(AmsAddress, UInt32, UInt32, CancellationToken) operation. The Task.TResult parameter contains the AdsErrorCode [575] as Result.

Reference

AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.12 AdsServer.DeleteDeviceNotificationResponseAsync Method

Sends an ADS Delete Device Notification response.

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected Task<AdsErrorCode> DeleteDeviceNotificationResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    CancellationToken cancel)
```
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>TwinCAT.Ads.AmsAddress [648]</td>
<td>The receiver's AMS address</td>
</tr>
<tr>
<td>invokeId</td>
<td>System.UInt32</td>
<td>The invoke ID provided by the receiver</td>
</tr>
<tr>
<td>result</td>
<td>TwinCAT.Ads.AdsErrorCode [575]</td>
<td>The ADS error code for the response</td>
</tr>
<tr>
<td>cancel</td>
<td>System.Threading.CancellationToken</td>
<td>The cancellation token</td>
</tr>
</tbody>
</table>

Return Value

Type: Task<AdsErrorCode [575]>
A task that represents the asynchronous DeleteDeviceNotificationResponseAsync(AmsAddress, UInt32, AdsErrorCode, CancellationToken) operation. The Task<TResult> parameter contains the AdsErrorCode [575] as Result.

Reference

AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.13 AdsServer.DeviceNotificationRequestAsync Method
Sends an ADS Device Notification request asynchronously

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```c#
protected Task<AdsErrorCode> DeviceNotificationRequestAsync(
    AmsAddress target,
    uint invokeId,
    uint numStampHeaders,
    NotificationSamplesStamp[] notificationHeaders,
    CancellationToken cancel
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>TwinCAT.Ads.AmsAddress [648]</td>
<td>The receiver's AMS address</td>
</tr>
<tr>
<td>invokeId</td>
<td>System.UInt32</td>
<td>The invokeId for this call. Used to find the matching ADS Confirmation</td>
</tr>
<tr>
<td>numStampHeaders</td>
<td>System.UInt32</td>
<td>The number of ADS Stamp Headers to be sent</td>
</tr>
<tr>
<td>notificationHeaders</td>
<td>.TwinCAT.Ads.Server.NotificationSamplesStamp [1195]</td>
<td>The array of ADS Stamp Headers to be sent</td>
</tr>
<tr>
<td>cancel</td>
<td>System.Threading.CancellationToken</td>
<td>The cancellation token</td>
</tr>
</tbody>
</table>
Return Value

Type: Task<AdsErrorCode>

The ADS error code for this call.

Reference

AdsServer Class

TwinCAT.Ads.Server Namespace

### 6.4.1.2.14 AdsServer.DeviceNotificationRequestSync Method

Sends an ADS Device Notification request (sync)

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
protected AdsErrorCode DeviceNotificationRequestSync(
    AmsAddress target,
    uint invokeId,
    uint numStampHeaders,
    NotificationSamplesStamp[] notificationHeaders
)
```

**Parameters**

- **target**
  
  Type: TwinCAT.Ads.AmsAddress
  
  The receiver's AMS address

- **invokeId**
  
  Type: System.UInt32
  
  The invokeId for this call. Used to find the matching ADS Confirmation

- **numStampHeaders**
  
  Type: System.UInt32
  
  The number of ADS Stamp Headers to be sent

- **notificationHeaders**
  
  Type: TwinCAT.Ads.Server.NotificationSamplesStamp
  
  The array of ADS Stamp Headers to be sent

**Return Value**

Type: AdsErrorCode

The ADS error code for this call.

Reference

AdsServer Class

TwinCAT.Ads.Server Namespace

### 6.4.1.2.15 AdsServer.Disconnect Method

Disconnects this ADS server from the local ADS router.

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public virtual bool Disconnect()
```

Return Value

Type: Boolean
true if disconnected, false if the AdsServer was disconnected before.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsServerException</td>
<td>Thrown if the disconnect call fails.</td>
</tr>
</tbody>
</table>

Reference

AdsServer Class

TwinCAT.Ads.Server Namespace

6.4.1.2.16 AdsServer.Dispose Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Performs application-defined tasks associated with freeing, releasing, or</td>
</tr>
<tr>
<td></td>
<td>resetting unmanaged resources.</td>
</tr>
<tr>
<td>Dispose(Boolean)</td>
<td>Releases unmanaged and - optionally - managed resources.</td>
</tr>
</tbody>
</table>

Reference

AdsServer Class

TwinCAT.Ads.Server Namespace

AdsServer.Dispose Method

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void Dispose()
```

Implements

IDisposable Dispose.
AdsServer.Dispose Method (Boolean)

Releases unmanaged and - optionally - managed resources.

**Namespace:** TwinCAT.Ads.Server [1121]

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected virtual void Dispose(
    bool disposing
)
```

**Parameters**

- **disposing**
  - Type: System.Boolean
  - true to release both managed and unmanaged resources; false to release only unmanaged resources.

**Reference**

- AdsServer Class [1121]
- Dispose Overload [1153]
- TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.17 AdsServer.Finalize Method

Finalizes an instance of the AdsServer class.

**Namespace:** TwinCAT.Ads.Server [1121]

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected override void Finalize()
```

**Implements**

Object.Finalize.

**Reference**

- AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.18 AdsServer.OnAddDeviceNotificationConfirmationAsync Method

Called when Add device Notification Confirmation is send.

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

```csharp
protected Task<AdsErrorCode> OnAddDeviceNotificationConfirmationAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    uint notificationHandle,
    CancellationToken cancel
)
```

Parameters

target
Type: TwinCAT.Ads.AmsAddress [648]
The r addr.

invokeId
Type: System.UInt32
The invoke identifier.

result
Type: TwinCAT.Ads.AdsErrorCode [575]
The result.

notificationHandle
Type: System.UInt32
The notification handle.

cancel
Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<AdsErrorCode> [575]
A task that represents the asynchronous OnAddDeviceNotificationConfirmationAsync(AmsAddress, UInt32, AdsErrorCode, UInt32, CancellationToken) operation. The Task.TResult parameter contains the AdsErrorCode as Result.

Reference

AdsServer Class [1121]

TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.19 AdsServer.OnConnected Method

Handler function that is called, when the AdsServer [1121] is connected.

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
6.4.1.2.20 AdsServer.OnDisconnect Method

Called when the AdsServer is about to be disconnected.

Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected virtual bool OnDisconnect()
Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

AdsServer Class [1121]
TwinCAT.Ads Server Namespace [1121]

6.4.1.2.22 AdsServer.OnServerConnectionStateChanged Method

Handles the ServerConnectionStateChanged [1181] event.

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected virtual void OnServerConnectionStateChanged(
    Object sender,
    ServerConnectionStateChangedEventArgs e
)

Parameters

sender Type: System.Object
The sender.

e Type: TwinCAT.Ads.Server.ServerConnectionStateChangedEventArgs [1199]
The ServerConnectionStateChangedEventArgs [1199] instance containing the event data.

Reference

AdsServer Class [1121]
TwinCAT.Ads Server Namespace [1121]

6.4.1.2.23 AdsServer.ReadConfirmationAsync Method

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected virtual Task<AdsErrorCode> ReadConfirmationAsync(
    AmsAddress targetAddress,
    uint invokeId,
    AdsErrorCode result,
Parameters

- `targetAddress` Type: TwinCAT.Ads.AmsAddress [648]
- `invokeId` Type: System.UInt32
- `result` Type: TwinCAT.Ads.AdsErrorCode [575]
- `readData` Type: ReadOnlyMemory
- `cancel` Type: System.Void

Return Value
Type: Task<TwinCAT.Ads.AdsErrorCode [575].>

Reference
AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.24 AdsServer.ReadDeviceInfoConfirmationAsync Method

Called when an ADS Read Device Info confirmation is received.

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C# protected virtual Task<TwinCAT.Ads.AdsErrorCode> ReadDeviceInfoConfirmationAsync(
    AmsAddress sender,
    uint invokeId,
    TwinCAT.Ads.AdsErrorCode result,
    string name,
    TwinCAT.Ads.AdsVersion version,
    CancellationToken cancel
)

Parameters

- `sender` Type: TwinCAT.Ads.AmsAddress [648]
The sender's AMS address
- `invokeId` Type: System.UInt32
  The invoke id provided by this server during the corresponding request
- `result` Type: TwinCAT.Ads.AdsErrorCode [575]
The ADS error code provided by the sender
- `name` Type: System.String
  The sender's name
version Type: TwinCAT.Ads.AdsVersion [642]
The sender's version

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<AdsErrorCode [575]>
A task that represents the asynchronous ReadDeviceInfoConfirmationAsync(AmsAddress, UInt32, AdsErrorCode, String, AdsVersion, CancellationToken) operation. The Task.TResult parameter contains the AdsErrorCode [575] as Result.

Remarks

Overwrite this method in derived classes to react on ADS Read Device Info confirmations.

Reference

AdsServer Class [1121]

TwinCAT.Ads Server Namespace [1121]

6.4.1.2.25 AdsServer.ReadDeviceInfoIndicationAsync Method

Called when an ADS Read Device Info indication is received.

Namespace: TwinCAT.Ads.Server [1121]

Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

protected virtual Task<AdsErrorCode> ReadDeviceInfoIndicationAsync(
    AmsAddress sender,
    uint invokeId,
    CancellationToken cancel
)

Parameters

sender Type: TwinCAT.Ads.AmsAddress [648]
The sender's / requester's AMS address

invokeId Type: System.UInt32
The invokeId provided by the sender

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<AdsErrorCode [575]>
A task that represents the asynchronous OnReadDeviceInfoIndicationAsync(AmsAddress, UInt32, CancellationToken) operation. The Task.TResult parameter contains the AdsErrorCode [575] as Result.

Remarks

Overwrite this method in derived classes to react on ADS Read Device Info indications. The default implementation replies with an ADS ServiceNotSupported error code (0x701).
6.4.1.2.26  AdsServer.ReadDeviceInfoRequestAsync Method

Sends an ADS Read Device Info request asynchronously

**Namespace:**  
TwinCAT.Ads.Server

**Assembly:**  
TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
protected Task<AdsErrorCode> ReadDeviceInfoRequestAsync(
    AmsAddress target,
    uint invokeId,
    CancellationToken cancel
)
```

**Parameters**

- **target**  
  Type: TwinCAT.Ads.AmsAddress
  The receiver’s AMS address

- **invokeId**  
  Type: System.UInt32
  The invokeId for this call. Used to find the matching ADS Confirmation

- **cancel**  
  Type: System.Threading.CancellationToken
  The cancellation token.

**Return Value**

Type: Task<AdsErrorCode>
A task that represents the asynchronous ReadDeviceInfoRequestAsync(AmsAddress, UInt32, CancellationToken) operation. The `Task<TResult>` parameter contains the `AdsErrorCode` as `Result`.

**Reference**

AdsServer Class
TwinCAT.Ads.Server Namespace

6.4.1.2.27  AdsServer.ReadDeviceInfoRequestSync Method

Sends an ADS Read Device Info request synchronously.

**Namespace:**  
TwinCAT.Ads.Server

**Assembly:**  
TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
### Syntax

**C#**

```csharp
protected AdsErrorCode ReadDeviceInfoRequestSync(
    AmsAddress target,
    uint invokeId
)
```

#### Parameters

- **target**
  - Type: `TwinCAT.Ads.AmsAddress` (648)
  - The receiver's AMS address

- **invokeId**
  - Type: `System.UInt32`
  - The invokeId for this call. Used to find the matching ADS Confirmation

#### Return Value

- Type: `AdsErrorCode` (575)
  - The ADS error code for this call.

### Reference

- **AdsServer Class** (1121)
- **TwinCAT.Ads.Server Namespace** (1121)

#### 6.4.1.2.28 AdsServer.ReadDeviceInfoResponseAsync Method

Sends an ADS Read Device Info response.

**Namespace:** TwinCAT.Ads.Server (1121)

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected Task<AdsErrorCode> ReadDeviceInfoResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    string name,
    AdsVersion version,
    CancellationToken cancel
)
```

#### Parameters

- **target**
  - Type: `TwinCAT.Ads.AmsAddress` (648)
  - The receiver's AMS address

- **invokeId**
  - Type: `System.UInt32`
  - The invoke ID provided by the receiver

- **result**
  - Type: `TwinCAT.Ads.AdsErrorCode` (575)
  - The ADS error code for the response

- **name**
  - Type: `System.String`
  - The name of this ADS server

- **version**
  - Type: `TwinCAT.Ads.AdsVersion` (642)
  - The version of this ADS server
cancel
Type: System.Threading.CancellationToken
The cancellation token.

Return Value
Type: Task<TwinCAT.Ads.AdsErrorCode>
A task that represents the asynchronous ReadDeviceInfoResponseAsync(AmsAddress, UInt32, AdsErrorCode, String, AdsVersion, CancellationToken) operation. The Task<TResult> parameter contains the AdsErrorCode as Result.

Reference
AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.29 AdsServer.ReadDeviceStateConfirmationAsync Method
Called when an ADS Read State confirmation is received.

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
protected virtual Task<TwinCAT.Ads.AdsErrorCode> ReadDeviceStateConfirmationAsync(
    AmsAddress sender,
    uint invokeId,
    TwinCAT.Ads.AdsErrorCode result,
    TwinCAT.Ads.AdsState adsState,
    ushort deviceState,
    CancellationToken cancel)

Parameters
sender
Type: TwinCAT.Ads.AmsAddress [648]
The sender's AMS address

invokeId
Type: System.UInt32
The invoke id provided by this server during the corresponding request

result
Type: TwinCAT.Ads.AdsErrorCode [575]
The ADS error code provided by the sender

adsState
Type: TwinCAT.Ads.AdsState [626]
The ADS state of the sender

deviceState
Type: System.UInt16
The device state of the sender

cancel
Type: System.Threading.CancellationToken
The cancellation token.

Return Value
Type: Task<TwinCAT.Ads.AdsErrorCode>
A task that represents the asynchronous ReadDeviceStateConfirmationAsync(AmsAddress, UInt32, AdsErrorCode, AdsState, UInt16, CancellationToken) operation. The Task<TResult> parameter contains the AdsErrorCode as Result.
Remarks
Overwrite this method in derived classes to react on ADS Read State confirmations.

Reference
AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.30 AdsServer.ReadDeviceStateIndicationAsync Method
Called when an ADS Read State indication is received.

Namespace: TwinCAT.Ads.Server [1121]
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
protected virtual Task<AdsErrorCode> ReadDeviceStateIndicationAsync(
    AmsAddress sender,
    uint invokeId,
    CancellationToken cancel
)

Parameters
sender Type: TwinCAT.Ads.AmsAddress [648]
The sender's AMS address
invokeId Type: System.UInt32
The invokeId provided by the sender
cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value
Type: Task<AdsErrorCode> [575].
A task that represents the asynchronous ReadDeviceStateIndicationAsync(AmsAddress, UInt32,

Remarks
Overwrite this method in derived classes to react on ADS Read State indications. The default
implementation replies with an ADS ServiceNotSupported error code (0x701).

Reference
AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.31 AdsServer.ReadDeviceStateRequestAsync Method
Sends an ADS Read State request (asynchronous)
TwinCAT.Ads Namespaces

Namespace:  TwinCAT.Ads.Server  [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected Task<AdsErrorCode> ReadDeviceStateRequestAsync(
    AmsAddress target,
    uint invokeId,
    CancellationToken cancel
)
```

Parameters

target Type: TwinCAT.Ads.AmsAddress [648]
The receiver's AMS address

invokeId Type: System.UInt32
The invokeId for this call. Used to find the matching ADS Confirmation

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<AdsErrorCode> [575].
A task that represents the asynchronous ReadDeviceStateRequestAsync(AmsAddress, UInt32, CancellationToken) operation. The Task.TResult parameter contains the AdsErrorCode [575] as Result.

Reference

AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.32 AdsServer.ReadDeviceStateRequestSync Method

Sends an ADS Read State request (synchronous)

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected AdsErrorCode ReadDeviceStateRequestSync(
    AmsAddress target,
    uint invokeId
)
```

Parameters

target Type: TwinCAT.Ads.AmsAddress [648]
The receiver's AMS address

invokeId Type: System.UInt32
The invokeId for this call. Used to find the matching ADS Confirmation
Return Value

Type: AdsErrorCode [575]
The ADS error code for this call.

Reference

AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.33 AdsServer.ReadDeviceStateResponseAsync Method

Sends an ADS Read State response.

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected Task<AdsErrorCode> ReadDeviceStateResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    AdsState adsState,
    ushort deviceState,
    CancellationToken cancel
)
```

Parameters

- **target** Type: TwinCAT.Ads.AmsAddress [648]
The receiver's AMS address
- **invokeId** Type: System.UInt32
The invoke ID provided by the receiver
- **result** Type: TwinCAT.Ads.AdsErrorCode [575]
The ADS error code for the response
- **adsState** Type: TwinCAT.Ads.AdsState [626]
The current ADS state of this ADS server
- **deviceState** Type: System.UInt16
The device state of this ADS server
- **cancel** Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<AdsErrorCode [575]>
A task that represents the asynchronous ReadDeviceStateResponseAsync(AmsAddress, UInt32, AdsErrorCode, AdsState, UInt16, CancellationToken) operation. The Task<TResult> parameter contains the AdsErrorCode [575] as Result.

Reference

AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]
6.4.1.2.34   AdsServer.ReadIndicationAsync Method

Called when an ADS Read indication is received.

**Namespace:** TwinCAT.Ads.Server 
**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
protected virtual Task<AdsErrorCode> ReadIndicationAsync(
    AmsAddress sender,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int readLength,
    CancellationToken cancel
)
```

**Parameters**

- **sender**  
  Type: TwinCAT.Ads.AmsAddress  
  The sender’s AMS address

- **invokeId**  
  Type: System.UInt32  
  The invokeId provided by the sender

- **indexGroup**  
  Type: System.UInt32  
  The index group of the requested ADS service

- **indexOffset**  
  Type: System.UInt32  
  The index offset of the requested ADS service

- **readLength**  
  Type: System.Int32  
  The number of bytes to be read

- **cancel**  
  Type: System.Threading.CancellationToken  
  The cancellation token.

**Return Value**

Type: Task<AdsErrorCode>  
A task that represents the asynchronous ReadIndicationAsync(AmsAddress, UInt32, UInt32, UInt32, Int32, CancellationToken) operation. The **Task TResult** parameter contains the AdsErrorCode as Result.

**Remarks**

Overwrite this method in derived classes to react on ADS Read indications. The default implementation replies with an ADS ServiceNotSupported error code (0x701).

**Reference**

- AdsServer Class
- TwinCAT.Ads.Server Namespace

6.4.1.2.35   AdsServer.ReadRequest Method

Sends an ADS Read Request.

**Namespace:** TwinCAT.Ads.Server  
**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
protected AdsErrorCode ReadRequest(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int readLength
)
```

Parameters

target Type: `TwinCAT.Ads.AmsAddress`  
The receiver's AMS address

invokeId Type: `System.UInt32`  
The invokeId for this call. Used to find the matching ADS Confirmation

indexGroup Type: `System.UInt32`  
The index group of the requested ADS service

indexOffset Type: `System.UInt32`  
The index group of the requested ADS service

readLength Type: `System.Int32`  
The number of bytes to be read

Return Value

Type: `AdsErrorCode`  
The ADS error code for this call.

Reference

- `AdsServer Class`  
- `TwinCAT.Ads.Server Namespace`

6.4.1.2.36 AdsServer.ReadRequestAsync Method

Sends an ADS Read Request asynchronously.

Namespace: `TwinCAT.Ads.Server`  
Assembly: `TwinCAT.Ads.Server`  
Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected Task<AdsErrorCode> ReadRequestAsync(
    AmsAddress target,
    uint invokeId,
    uint indexId,
    uint indexOffset,
    int readLength,
    CancellationToken cancel
)
```

Parameters

target Type: `TwinCAT.Ads.AmsAddress`  
The receiver's AMS address
invokeId Type: System.UInt32
The invokeId for this call. Used to find the matching ADS Confirmation

indexGroup Type: System.UInt32
The index group of the requested ADS service

indexOffset Type: System.UInt32
The index group of the requested ADS service

readLength Type: System.Int32
The number of bytes to be read

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value
Type: Task<AdsErrorCode>
A task that represents the asynchronous ReadRequestAsync(AmsAddress, UInt32, UInt32, UInt32, Int32, CancellationToken) operation. The Task<TResult> parameter contains the AdsErrorCode as Result.

Reference
AdsServer Class
TwinCAT.Ads.Server Namespace

6.4.1.2.37 AdsServer.ReadResponseAsync Method

Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
protected Task<AdsErrorCode> ReadResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    Memory data,
    void cancel
)

Parameters

target Type: TwinCAT.Ads.AmsAddress

invokeId Type: System.UInt32

result Type: TwinCAT.Ads.AdsErrorCode

data Type: Memory

cancel Type: System.Void

Return Value
Type: Task<AdsErrorCode>.
Reference

AdsServer Class [► 1121]

TwinCAT Ads Server Namespace [► 1121]

6.4.1.2.38 AdsServer.ReadWriteConfirmationAsync Method

Namespace: TwinCAT.Ads.Server [► 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected virtual Task<AdsErrorCode> ReadWriteConfirmationAsync(AmsAddress sender, uint invokeId, AdsErrorCode result, ReadOnlyMemory readData, void cancel)
```

Parameters

- sender Type: TwinCAT.Ads.AmsAddress [► 648]
- invokeId Type: System.UInt32
- result Type: TwinCAT.Ads.AdsErrorCode [► 575]
- readData Type: ReadOnlyMemory
- cancel Type: System.Void

Return Value

Type: Task<AdsErrorCode> [► 575].

Reference

AdsServer Class [► 1121]
TwinCAT Ads Server Namespace [► 1121]

6.4.1.2.39 AdsServer.ReadWriteIndicationAsync Method

Namespace: TwinCAT.Ads.Server [► 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

protected virtual Task<AdsErrorCode> ReadWriteIndicationAsync(
    AmsAddress sender,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int readLength,
    ReadOnlyMemory writeData,
    void cancel
)

Parameters

sender Type: TwinCAT.Ads.AmsAddress [648]
invokeId Type: System.UInt32
indexGroup Type: System.UInt32
indexOffset Type: System.UInt32
readLength Type: System.Int32
writeData Type: ReadOnlyMemory
cancel Type: System.Void

Return Value

Type: Task<AdsErrorCode> [575].

Reference

AdsServer Class [1121]

TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.40 AdsServer.ReadWriteRequestAsync Method

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

protected Task<AdsErrorCode> ReadWriteRequestAsync(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int readLength,
    ReadOnlySpan writeData,
    void cancel
)
Parameters

target Type: TwinCAT.Ads.AmsAddress

invokeId Type: System.UInt32

indexGroup Type: System.UInt32

indexOffset Type: System.UInt32

readLength Type: System.Int32

writeData Type: ReadOnlySpan

cancel Type: System.Void

Return Value
Type: Task<AdsErrorCode>

Reference
AdsServer Class
TwinCAT.Ads.Server Namespace

6.4.1.2.41 AdsServer.ReadWriteRequestSync Method

Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected AdsErrorCode ReadWriteRequestSync(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int readLength,
    ReadOnlySpan data
)

Parameters

target Type: TwinCAT.Ads.AmsAddress

invokeId Type: System.UInt32

indexGroup Type: System.UInt32

indexOffset Type: System.UInt32

readLength Type: System.Int32
data  
Type: ReadOnlySpan

Return Value
Type: AdsErrorCode [575]

Reference
AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.42 AdsServer.ReadWriteResponseAsync Method

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#

```csharp
protected Task<AdsErrorCode> ReadWriteResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    ReadOnlyMemory data,
    void cancel
)
```

Parameters

target  
Type: TwinCAT.Ads.AmsAddress [648]

invokeId  
Type: System.UInt32

result  
Type: TwinCAT.Ads.AdsErrorCode [575]

data  
Type: ReadOnlyMemory

cancel  
Type: System.Void

Return Value
Type: Task<AdsErrorCode>[575]

Reference
AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.43 AdsServer.WriteConfirmationAsync Method

Called when an ADS Write confirmation is received. Overwrite this method in derived classes to react on ADS Write confirmations.
Syntax

C#

```csharp
protected virtual Task<AdsErrorCode> WriteConfirmationAsync(
    AmsAddress sender,
    uint invokeId,
    AdsErrorCode result,
    CancellationToken cancel
)
```

Parameters

- **sender**
  - Type: `TwinCAT.Ads.AmsAddress` (648)
  - The sender's AMS address

- **invokeId**
  - Type: `System.UInt32`
  - The invoke id provided by this server during the corresponding request

- **result**
  - Type: `TwinCAT.Ads.AdsErrorCode` (575)
  - The ADS error code provided by the sender

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.

Return Value

- **Type:** `Task<AdsErrorCode>` (575)
  - A task that represents the asynchronous `WriteConfirmationAsync(AmsAddress, UInt32, AdsErrorCode, CancellationToken)` operation. The `Task.TResult` parameter contains the `AdsErrorCode` (575) as `Result`.

Reference

- [AdsServer Class](#)
- [TwinCAT.Ads.Server Namespace](#)

### 6.4.1.2.44 AdsServer.WriteControlConfirmationAsync Method

Called when an ADS Write Control confirmation is received.

Syntax

C#

```csharp
protected virtual Task<AdsErrorCode> WriteControlConfirmationAsync(
    AmsAddress sender,
    uint invokeId,
    AdsErrorCode result,
    CancellationToken cancel
)
```

Parameters

- **sender**
  - Type: `TwinCAT.Ads.AmsAddress` (648)
  - The sender's AMS address
invokeId Type: System.UInt32
The invoke id provided by this server during the corresponding request

result Type: TwinCAT.Ads.AdsErrorCode [575]
The ADS error code provided by the sender

cancel Type: System.Threading.CancellationToken
The cancellation token.

**Return Value**

Type: Task<AdsErrorCode [575]>
A task that represents the asynchronous WriteControlConfirmationAsync(AmsAddress, UInt32, AdsErrorCode, CancellationToken) operation. The Task<TResult> parameter contains the AdsErrorCode [575] as Result.

**Remarks**

Overwrite this method in derived classes to react on ADS Write Control confirmations.

**Reference**

AdsServer Class [1121]

TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.45 AdsServer.WriteControlIndicationAsync Method

**Namespace:** TwinCAT.Ads.Server [1121]

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

protected virtual Task<AdsErrorCode> WriteControlIndicationAsync(
    AmsAddress sender,
    uint invokeId,
    AdsState adsState,
    ushort deviceState,
    ReadOnlyMemory data,
    void cancel)

**Parameters**

sender Type: TwinCAT.Ads.AmsAddress [648]

invokeId Type: System.UInt32

adsState Type: TwinCAT.Ads.AdsState [626]

deviceState Type: System.UInt16

data Type: ReadOnlyMemory

cancel Type: System.Void
Return Value

Type: Task<AdsErrorCode> [575].

Reference

AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.46 AdsServer.WriteControlRequest Method

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

protected AdsErrorCode WriteControlRequest(
    AmsAddress target,
    uint invokeId,
    AdsState adsState,
    ushort deviceState,
    ReadOnlySpan data
)

Parameters

target Type: TwinCAT.Ads.AmsAddress [648]
invokeId Type: System.UInt32
adsState Type: TwinCAT.Ads.AdsState [626]
deviceState Type: System.UInt16
data Type: ReadOnlySpan

Return Value

Type: AdsErrorCode [575]

Reference

AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.47 AdsServer.WriteControlRequestAsync Method

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

```csharp
protected Task<AdsErrorCode> WriteControlRequestAsync(
    AmsAddress target,
    uint invokeId,
    AdsState adsState,
    ushort deviceState,
    ReadOnlySpan data,
    void cancel
)
```

Parameters

- **target**
  Type: `TwinCAT.Ads.AmsAddress` [648]

- **invokeId**
  Type: `System.UInt32`

- **adsState**
  Type: `TwinCAT.Ads.AdsState` [626]

- **deviceState**
  Type: `System.UInt16`

- **data**
  Type: `ReadOnlySpan`

- **cancel**
  Type: `System.Void`

Return Value

Type: `Task<AdsErrorCode>` [575].

Reference

- **AdsServer Class [1121]**
- **TwinCAT.Ads.Server Namespace [1121]**

6.4.1.2.48 AdsServer.WriteControlRequestSync Method

Namespace: `TwinCAT.Ads.Server` [1121]

Assembly: `TwinCAT.Ads.Server` (in `TwinCAT.Ads.Server.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected AdsErrorCode WriteControlRequestSync(
    AmsAddress target,
    uint invokeId,
    AdsState adsState,
    ushort deviceState,
    ReadOnlySpan data
)
```

Parameters

- **target**
  Type: `TwinCAT.Ads.AmsAddress` [648]
invokeId Type: System.UInt32
adsState Type: TwinCAT.Ads.AdsState [626]
deviceState Type: System.UInt16
data Type: ReadOnlySpan

Return Value
Type: AdsErrorCode [575]

Reference
AdsServer Class [1121]
TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.49 AdsServer.WriteControlResponseAsync Method
Sends an ADS Write Control response.

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
protected Task<AdsErrorCode> WriteControlResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    CancellationToken cancel
)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>TwinCAT.Ads.AmsAddress [648]</td>
<td>The receiver's AMS address</td>
</tr>
<tr>
<td>invokeId</td>
<td>System.UInt32</td>
<td>The invoke ID provided by the receiver</td>
</tr>
<tr>
<td>result</td>
<td>TwinCAT.Ads.AdsErrorCode [575]</td>
<td>The ADS error code for the response</td>
</tr>
<tr>
<td>cancel</td>
<td>System.Threading.CancellationToken</td>
<td>The cancellation token.</td>
</tr>
</tbody>
</table>

Return Value
Type: Task<AdsErrorCode> [575],
A task that represents the asynchronous WriteControlResponseAsync(AmsAddress, UInt32, AdsErrorCode, CancellationToken) operation. The Task<TResult> parameter contains the AdsErrorCode [575] as Result.

Reference
AdsServer Class [1121]
### 6.4.1.2.50 AdsServer.WriteIndicationAsync Method

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected virtual Task<AdsErrorCode> WriteIndicationAsync(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeData,
    void cancel
)
```

**Parameters**

- `target` Type: TwinCAT.Ads.AmsAddress
- `invokeId` Type: System.UInt32
- `indexGroup` Type: System.UInt32
- `indexOffset` Type: System.UInt32
- `writeData` Type: ReadOnlyMemory
- `cancel` Type: System.Void

**Return Value**

Type: Task<AdsErrorCode>

**Reference**

- AdsServer Class
- TwinCAT.Ads.Server Namespace

### 6.4.1.2.51 AdsServer.WriteRequest Method

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected AdsErrorCode WriteRequest(
    AmsAddress target,
    uint invokeId,

```
Parameters

- **target**: Type: `TwinCAT.Ads.AmsAddress` [648]
- **invokeId**: Type: `System.UInt32`
- **indexGroup**: Type: `System.UInt32`
- **indexOffset**: Type: `System.UInt32`
- **data**: Type: `ReadOnlySpan`

Return Value

Type: `AdsErrorCode` [575]

Reference

- AdsServer Class [1121]
- TwinCAT.Ads.Server Namespace [1121]

### 6.4.1.2.52 AdsServer.WriteRequestAsync Method

**Namespace**: TwinCAT.Ads.Server [1121]

**Assembly**: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
protected Task<AdsErrorCode> WriteRequestAsync(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    ReadOnlySpan data,
    void cancel
)
```

Parameters

- **target**: Type: `TwinCAT.Ads.AmsAddress` [648]
- **invokeId**: Type: `System.UInt32`
- **indexGroup**: Type: `System.UInt32`
- **indexOffset**: Type: `System.UInt32`
- **data**: Type: `ReadOnlySpan`
Return Value

Type: Task<AdsErrorCode> [575].

Reference

AdsServer Class [1121]

TwinCAT.Ads.Server Namespace [1121]

6.4.1.2.53 AdsServer.WriteResponseAsync Method

Sends an ADS Write response.

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

protected Task<AdsErrorCode> WriteResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    CancellationToken cancel)

Parameters

target
  Type: TwinCAT.Ads.AmsAddress [648]
The receiver's AMS address

invokeId
  Type: System.UInt32
  The invoke ID provided by the receiver

result
  Type: TwinCAT.Ads.AdsErrorCode [575]
The ADS error code for the response

cancel
  Type: System.Threading.CancellationToken
  The cancellation token.

Return Value

Type: Task<AdsErrorCode> [575].
A task that represents the asynchronous WriteResponseAsync(AmsAddress, UInt32, AdsErrorCode, CancellationToken) operation. The Task.TResult parameter contains the AdsErrorCode [575] as Result.

Reference

AdsServer Class [1121]

TwinCAT.Ads.Server Namespace [1121]

6.4.1.3 AdsServer Events

The AdsServer [1121] type exposes the following members.
Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServerConnectionStateChanged</td>
<td>The connection status has changed</td>
</tr>
</tbody>
</table>

Reference

AdsServer Class

TwinCAT.Ads.Server Namespace

6.4.1.3.1 AdsServer.ServerConnectionStateChanged Event

The connection status has changed

Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public event EventHandler<ServerConnectionStateChangedEventArgs> ServerConnectionStateChanged
```

Value

Type: System.EventHandler<ServerConnectionStateChangedEventArgs>

Reference

AdsServer Class

TwinCAT.Ads.Server Namespace

6.4.2 AdsServerException Class

An AdsServerException is thrown on communication errors in the AdsServer class.

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.Ads.Server.AdsServerException
      TwinCAT.Ads.Server.LoopbackNotRegisteredException
      TwinCAT.Ads.Server.ServerNotConnectedException

Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
[SerializableAttribute]
public class AdsServerException : Exception
```
The AdsServerException type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsServerException</td>
<td>Initializes a new instance of the AdsServerException class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets or sets the error code.</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.Server Namespace [1121]

6.4.2.1 AdsServerException Constructor

Initializes a new instance of the AdsServerException [1181] class.

Namespace: TwinCAT.Ads.Server [1121]  
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected AdsServerException(  
    SerializationInfo serializationInfo,  
    StreamingContext streamingContext
)
```

Parameters

- serializationInfo  
  Type: System.Runtime.Serialization.SerializationInfo  
  The serialization information.
- streamingContext  
  Type: System.Runtime.Serialization.StreamingContext  
  The streaming context.

Reference

AdsServerException Class [1181]  
TwinCAT.Ads.Server Namespace [1121]

6.4.2.2 AdsServerException Properties

The AdsServerException [1181] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ErrorCode [1184]</td>
<td>Gets or sets the error code.</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### AdsServerException Class [1181]

**TwinCAT.Ads.Server Namespace [1121]**

### 6.4.2.2.1 AdsServerException.ErrorCode Property

Gets or sets the error code.

**Namespace:** TwinCAT.Ads.Server [1121]
**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AmsServerErrorCode ErrorCode { get; set; }
```

**Property Value**

Type: AmsServerErrorCode [1548]
The error code.

**Reference**

AdsServerException Class [1181]

TwinCAT.Ads.Server Namespace [1121]

### 6.4.2.3 AdsServerException Methods

The AdsServerException [1181] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.     (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. ( Overrides Exception.GetObjectData(SerializationInfo, StreamingContext) )</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

#### Reference

AdsWithException Class [1181]

TwinCAT.Ads.Server Namespace [1121]

#### 6.4.2.3.1 AdsServerException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

**Namespace:** TwinCAT.Ads.Server [1121]

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

**Parameters**

- **info**
  
  Type: System.Runtime.Serialization.SerializationInfo
  
  The SerializationInfo that holds the serialized object data about the exception being thrown.

- **context**
  
  Type: System.Runtime.Serialization.StreamingContext
  
  The StreamingContext that contains contextual information about the source or destination.

**Implements**

- ISerializable.GetObjectData(SerializationInfo, StreamingContext)
- Exception.GetObjectData(SerializationInfo, StreamingContext)

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td></td>
</tr>
<tr>
<td>ArgumentNullException</td>
<td></td>
</tr>
</tbody>
</table>

TC1000 Version: 1.1
6.4.2.4 AdsServerException Events

The AdsServerException type exposes the following members.

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

6.4.3 ErrorEventArgs Class

This class implements the event arguments passed by the TcAdsServerExEvent.

**Inheritance Hierarchy**

- System.Object
  - System.EventArgs
    - TwinCAT.Ads.Server.ErrorEventArgs

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public class ErrorEventArgs : EventArgs
```

The ErrorEventArgs type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception</td>
<td>Returns the exception that caused the event.</td>
</tr>
<tr>
<td>Message</td>
<td>Returns the exception message.</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.Server Namespace [1121]

6.4.3.1 ErrorEventArgs Properties

The ErrorEventArgs [1186] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception</td>
<td>Returns the exception that caused the event.</td>
</tr>
<tr>
<td>Message</td>
<td>Returns the exception message.</td>
</tr>
</tbody>
</table>

Reference

ErrorEventArgs Class [1186]

TwinCAT.Ads.Server Namespace [1121]

6.4.3.1.1 ErrorEventArgs.Exception Property

Returns the exception that caused the event.

Namespace:  TwinCAT.Ads.Server [1121]
Assembly:  TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Exception Exception { get; }
```

Return Value

Type: Exception
The Exception that caused the event.

Reference

ErrorEventArgs Class [1186]

TwinCAT.Ads.Server Namespace [1121]
6.4.3.1.2 ErrorEventArgs.Message Property

Returns the exception message.

Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public string Message { get; }
```

Return Value

Type: String
The Exception message.

Reference

ErrorEventArgs Class
TwinCAT.Ads.Server Namespace

6.4.3.2 ErrorEventArgs Methods

The ErrorEventArgs type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

ErrorEventArgs Class
TwinCAT.Ads.Server Namespace

6.4.4 LoopbackNotRegisteredException Class

The Tcp Loopback client is not registered. Implements the AdsServerException.
Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.Ads.Server.AdsServerException
    TwinCAT.Ads.Server.LoopbackNotRegisteredException

Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class LoopbackNotRegisteredException : AdsServerException

The LoopbackNotRegisteredException type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets or sets the error code. (Inherited from AdsServerException.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from AdsServerException.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object</td>
</tr>
<tr>
<td></td>
<td>that contains serialized data about the exception. (Inherited from</td>
</tr>
<tr>
<td></td>
<td><code>Exception</code>.)</td>
</tr>
</tbody>
</table>

### Reference

- **TwinCAT.Ads.Server Namespace** [1121]
- **TwinCAT.Ads.Server.AdsServerException** [1181]

### 6.4.4.1 LoopbackNotRegisteredException Properties

The `LoopbackNotRegisteredException` [1188] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets or sets the error code. (Inherited from <code>AdsServerException</code> [1181].)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

### Reference

- **LoopbackNotRegisteredException Class** [1188]
- **TwinCAT.Ads.Server Namespace** [1121]
6.4.4.2 LoopbackNotRegisteredException Methods

The LoopbackNotRegisteredException type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root</td>
</tr>
<tr>
<td></td>
<td>cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with</td>
</tr>
<tr>
<td></td>
<td>information about the exception. (Inherited from AdsServerException)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

LoopbackNotRegisteredException Class

TwinCAT.Ads.Server Namespace

6.4.4.3 LoopbackNotRegisteredException Events

The LoopbackNotRegisteredException type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object</td>
</tr>
<tr>
<td></td>
<td>that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

LoopbackNotRegisteredException Class

TwinCAT.Ads.Server Namespace

6.4.5 NotificationDataSample Class

This class implements an ADS Notification Sample. It contains the notification handle and the variable data.
Inheritance Hierarchy

System.Object
  TwinCAT.Ads.Server.NotificationDataSample

Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public sealed class NotificationDataSample
```

The NotificationDataSample type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotificationDataSample</td>
<td></td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotificationHandle</td>
<td>Gets the notification handle.</td>
</tr>
<tr>
<td>SampleData</td>
<td>Gets the Sample Data.</td>
</tr>
<tr>
<td>SampleSize</td>
<td>Gets the Sample Size.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from System.Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from System.Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from System.Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from System.Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.Server Namespace [1121]

6.4.5.1 NotificationDataSample Constructor

Namespace: TwinCAT.Ads.Server
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#
public NotificationDataSample(
    uint handle,
    ReadOnlyMemory data
)

Parameters

handle Type: System.UInt32
data Type: ReadOnlyMemory

Reference

NotificationDataSample Class [1191]
TwinCAT.Ads.Server Namespace [1121]

6.4.5.2 NotificationDataSample Properties

The NotificationDataSample [1191] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotificationHandle</td>
<td>Gets the notification handle.</td>
</tr>
<tr>
<td>SampleData</td>
<td>Gets the Sample Data.</td>
</tr>
<tr>
<td>SampleSize</td>
<td>Gets the Sample Size.</td>
</tr>
</tbody>
</table>

Reference

NotificationDataSample Class [1191]
TwinCAT.Ads.Server Namespace [1121]

6.4.5.2.1 NotificationDataSample.NotificationHandle Property

Gets the notification handle.

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public uint NotificationHandle { get; }

Property Value

Type: UInt32
6.4.5.2.2 NotificationDataSample.SampleData Property

Gets the Sample Data.

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public ReadOnlyMemory SampleData { get; }
```

**Property Value**

Type: `ReadOnlyMemory`

Reference

NotificationDataSample Class [1191]

TwinCAT.Ads.Server Namespace [1121]

6.4.5.2.3 NotificationDataSample.SampleSize Property

Gets the Sample Size.

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public int SampleSize { get; }
```

**Property Value**

Type: `Int32`

Reference

NotificationDataSample Class [1191]

TwinCAT.Ads.Server Namespace [1121]

6.4.5.3 NotificationDataSample Methods

The `NotificationDataSample` [1191] type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

NotificationDataSample Class  [ ]  1191
TwinCAT.Ads.Server Namespace  [ ]  1121

6.4.6 NotificationSamplesStamp Class

This class implements an ADS Stamp Header containing multiple ADS Notification Samples (TcAdsStampHeader)

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.Server.NotificationSamplesStamp

Namespace: TwinCAT.Ads.Server  [ ]  1121
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public sealed class NotificationSamplesStamp

The NotificationSamplesStamp type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotificationSamplesStamp</td>
<td>Initializes a new instance of the NotificationSamplesStamp class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotificationSamples</td>
<td>Get the array of notification samples contained in this header.</td>
</tr>
<tr>
<td>NumSamples</td>
<td>Get the number of notification samples contained in this header.</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Get or set the time stamp of this header.</td>
</tr>
</tbody>
</table>
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Inherited from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
<td>Object</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
<td>Object</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
<td>Object</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
<td>Object</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.Server Namespace [1121]

6.4.6.1 NotificationSamplesStamp Constructor

Initializes a new instance of the NotificationSamplesStamp [1195] class.

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public NotificationSamplesStamp(
    long timestamp,
    NotificationDataSample[] samples
)
```

Parameters

timestamp
  Type: System.Int64
  The timestamp.
samples
  Type: .TwinCAT.Ads.Server.NotificationDataSample [1191]
  The notification samples.

Reference

NotificationSamplesStamp Class [1195]
TwinCAT.Ads.Server Namespace [1121]

6.4.6.2 NotificationSamplesStamp Properties

The NotificationSamplesStamp [1195] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotificationSamples</td>
<td>Get the array of notification samples contained in this header.</td>
</tr>
<tr>
<td>NumSamples</td>
<td>Get the number of notification samples contained in this header.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Get or set the time stamp of this header.</td>
</tr>
</tbody>
</table>

**Reference**

NotificationSamplesStamp Class  [1195]

TwinCAT.Ads.Server Namespace  [1121]

### 6.4.6.2.1 NotificationSamplesStamp.NotificationSamples Property

Get the array of notification samples contained in this header.

**Namespace:**  TwinCAT.Ads.Server  [1121]

**Assembly:**  TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public NotificationDataSample[] NotificationSamples { get; }
```

**Property Value**

Type:  NotificationDataSample  [1191].

**Reference**

NotificationSamplesStamp Class  [1195]

TwinCAT.Ads.Server Namespace  [1121]

### 6.4.6.2.2 NotificationSamplesStamp.NumSamples Property

Get the number of notification samples contained in this header.

**Namespace:**  TwinCAT.Ads.Server  [1121]

**Assembly:**  TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int NumSamples { get; }
```

**Property Value**

Type:  Int32

**Reference**

NotificationSamplesStamp Class  [1195]

TwinCAT.Ads.Server Namespace  [1121]
6.4.6.3 NotificationSamplesStamp.TimeStamp Property

Get or set the time stamp of this header.

**Namespace:** TwinCAT.Ads.Server [1121]

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public long TimeStamp { get; }
```

**Property Value**

Type: Int64

**Reference**

NotificationSamplesStamp Class [1195]

TwinCAT.Ads.Server Namespace [1121]

6.4.6.3 NotificationSamplesStamp Methods

The NotificationSamplesStamp [1195] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Reference**

NotificationSamplesStamp Class [1195]

TwinCAT.Ads.Server Namespace [1121]

6.4.7 ServerConnectionState Enumeration

The Server Connection State

**Namespace:** TwinCAT.Ads.Server [1121]

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public enum ServerConnectionState
```
Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>None / Uninitialized / Disconnect</td>
</tr>
<tr>
<td>Disconnected</td>
<td>1</td>
<td>Disconnected State.</td>
</tr>
<tr>
<td>Connected</td>
<td>2</td>
<td>Connected state.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.Server Namespace [1121]

6.4.8 ServerConnectionStateChangedEventArgs Class

Class ConnectionStateChangedEventArgs (Server Connections)

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.Server.ServerConnectionStateChangedEventArgs

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class ServerConnectionStateChangedEventArgs : EventArgs

The ServerConnectionStateChangedEventArgs type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServerConnectionStateChangedEventArgs [1200]</td>
<td>Initializes a new instance of the ServerConnectionStateChangedEventArgs class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address [1201]</td>
<td>The Server address</td>
</tr>
<tr>
<td>State [1201]</td>
<td>The Connection state</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### ServerConnectionStateChangedEventArgs Constructor

Initializes a new instance of the ServerConnectionStateChangedEventArgs class.

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**C#**

```csharp
public ServerConnectionStateChangedEventArgs(
    AmsAddress address,
    ServerConnectionState state
)
```

**Parameters**

- **address**
  - Type: TwinCAT.Ads.AmsAddress
  - The address.

- **state**
  - Type: TwinCAT.Ads.Server.ServerConnectionState
  - The state.

### ServerConnectionStateChangedEventArgs Properties

The ServerConnectionStateChangedEventArgs type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>The Server address</td>
</tr>
<tr>
<td>State</td>
<td>The Connection state</td>
</tr>
</tbody>
</table>
6.4.8.2.1 ServerConnectionStateChangedEventArgs.Address Property

The Server address

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AmsAddress Address { get; }
```

**Property Value**

Type: AmsAddress

**Reference**

ServerConnectionStateChangedEventArgs Class

TwinCAT.Ads.Server Namespace

6.4.8.2.2 ServerConnectionStateChangedEventArgs.State Property

The Connection state

**Namespace:** TwinCAT.Ads.Server

**Assembly:** TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ServerConnectionState State { get; }
```

**Property Value**

Type: ServerConnectionState

**Reference**

ServerConnectionStateChangedEventArgs Class

TwinCAT.Ads.Server Namespace

6.4.8.3 ServerConnectionStateChangedEventArgs Methods

The ServerConnectionStateChangedEventArgs type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
</tbody>
</table>

Reference

ServerConnectionStateChangedEventArgs Class [1199]
TwinCAT.Ads.Server Namespace [1121]

6.4.9 ServerNotConnectedException Class

The AdsServer is not connected. Implements the AdsServerException [1181]

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.Ads.Server.AdsServerException [1181]
    TwinCAT.Ads.Server.ServerNotConnectedException

Namespace: TwinCAT.Ads.Server [1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C# public class ServerNotConnectedException : AdsServerException

The ServerNotConnectedException type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets or sets the error code. (Inherited from AdsServerException [1181].)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from AdsServerException.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Reference

- TwinCAT.Ads.Server Namespace [1121]
- TwinCAT.Ads.Server.AdsServerException [1181]

#### 6.4.9.1 ServerNotConnectedException Properties

The ServerNotConnectedException [1202] type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode [1184]</td>
<td>Gets or sets the error code. (Inherited from AdsServerException [1181].)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Reference

ServerNotConnectedException Class [1202]

TwinCAT.Ads.Server Namespace [1121]

#### 6.4.9.2 ServerNotConnectedException Methods

The ServerNotConnectedException [1202] type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from AdsServerException [1181].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Reference

ServerNotConnectedException Class [1202]

TwinCAT.Ads.Server Namespace [1121]
6.4.9.3 ServerNotConnectedException Events

The ServerNotConnectedException [1202] type exposes the following members.

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

**Reference**

ServerNotConnectedException Class [1202]

TwinCAT.Ads.Server Namespace [1121]

### TwinCAT.Ads.SumCommand Namespace

ADS offers powerful and fast communication to exchange any kind of information. It's possible to read single variables or complete arrays and structures with each one single ADS-API call. ADS Sum-Commands offer to read/write with one single ADS call multiple variables which are not structured within a linear memory, effectively reducing roundtrips.

#### Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultSumCommand</td>
<td>Result class for an asynchronous TwinCAT.Ads.SumCommand:</td>
</tr>
<tr>
<td>ResultSumHandles</td>
<td>Result class for an asynchronous SumCreateHandles [1222].</td>
</tr>
<tr>
<td>ResultSumHandles2</td>
<td>Result class for an asynchronous SumCreateHandles [1222].</td>
</tr>
<tr>
<td>ResultSumReadRaw</td>
<td>Result class for an asynchronous SumRead.</td>
</tr>
<tr>
<td>ResultSumValues</td>
<td>Result class for an asynchronous SumSymbolRead [1242].</td>
</tr>
<tr>
<td>SumCreateHandles</td>
<td>SumCommandBase for getting variable handles by a set of InstancePaths</td>
</tr>
<tr>
<td>SumHandleRead</td>
<td>Read (primitive, Any) values by Handle SumCommandBase.</td>
</tr>
<tr>
<td>SumHandleWrite</td>
<td>Write any (primitive) values by Handle SumCommandBase.</td>
</tr>
<tr>
<td>SumReleaseHandles</td>
<td>Release Handles SumCommandBase.</td>
</tr>
</tbody>
</table>
## 6.5.1 ISumCommand Interface

Interface for SumCommands (Combined commands)

**Namespace:** TwinCAT.Ads.SumCommand

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public interface ISumCommand
```

The `ISumCommand` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executed</td>
<td>Gets a value indicating whether this <code>ISumCommand</code> was already executed.</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether this <code>ISumCommand</code> failed.</td>
</tr>
<tr>
<td>Result</td>
<td>Gets the <code>AdsErrorCode</code> of the main <code>SumCommandBase</code> ADS Request</td>
</tr>
<tr>
<td>SubResults</td>
<td>Gets the sub results of the single Sub Requests.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether this <code>ISumCommand</code> is succeeded.</td>
</tr>
</tbody>
</table>

### Remarks

ADS offers powerful and fast communication to exchange any kind of information. It's possible to read single variables or complete arrays and structures with each one single ADS-API call. The ADS Sum Command command offers to read with one single ADS call multiple variables which are not structured within a linear memory. As a result the ADS caller application (like scada Systems etc.) can extremely speed up cyclic polling: Sample:

- Until now: Polling 4000 single variables which are not in a linear area (like array / structure / fixed PLC address) would cause 4000 single Ads-ReadReq with each 1-2 ms protocol time. As a result the scanning of these variables take 4000ms-8000ms.
• New Ads-Command allows to read multiple variables with one single ADS-ReadReq: 4000 single variables are handled with e.g. 8 single Ads-ReadReq (each call requesting 500 variables) with each 1-2 ms protocol time. As a result the scanning of these variables take just few 10ms.

REQUIREMENTS AND IMPORTANT LIMITATIONS: Note that ADS is just a transport layer, but there could be important side effects. So read this requirements and take care on limitations:

• **Version of target ADS Device** - ADS itself is just the transport layer, but the requested ADS device has to support the ADS-Command.

• **Bytes length of requested data** - Requesting a large list of values from variables is fine, but the requested data of the Ads-response (the data-byte-length) have to pass the AMS Router (size by default a 2048kb) So the caller has to limit the requested variables based on calculation of requested data-byte-length.

• **Number of Sub-ADS calls**: Highly recommended to max. 500! - If the PLC is processing one ADS request, it will completely work on this single ADS request BEFORE starting next PLC cycle. As a result one single ADS request with 200.000 sub-Ads-requests would cause that PLC would collect and copy 200.000 variables into one single ADS response, before starting next PLC. So this large number of ads-sub-commands will jitter the PLC execution! **We highly recommend to not request more than 500 Ads-Sub commands**

Reference

TwinCAT.Ads.SumCommand Namespace [1205]

6.5.1.1 ISumCommand Properties

The ISumCommand type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executed</td>
<td>Gets a value indicating whether this ISumCommand was already executed.</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether this ISumCommand failed.</td>
</tr>
<tr>
<td>Result</td>
<td>Gets the AdsErrorCode of the main SumCommandBase ADS Request</td>
</tr>
<tr>
<td>SubResults</td>
<td>Gets the sub results of the single Sub Requests.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether this ISumCommand is succeeded.</td>
</tr>
</tbody>
</table>

Reference

ISumCommand Interface [1206]

TwinCAT.Ads.SumCommand Namespace [1205]

6.5.1.1.1 ISumCommand.Executed Property

Gets a value indicating whether this ISumCommand was already executed.

Namespace: TwinCAT.Ads.SumCommand [1205]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

bool Executed { get; }

Property Value

Type: Boolean
true if executed; otherwise, false.

Reference

ISumCommand Interface [1206]

TwinCAT.Ads.SumCommand Namespace [1205]

6.5.1.1.2 ISumCommand.Failed Property

Gets a value indicating whether this ISumCommand [1206] failed.

Namespace: TwinCAT.Ads.SumCommand [1206]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

bool Failed { get; }

Property Value

Type: Boolean
true if failed; otherwise, false.

Reference

ISumCommand Interface [1206]

TwinCAT.Ads.SumCommand Namespace [1205]

6.5.1.1.3 ISumCommand.Result Property

Gets the AdsErrorCode [575] of the main SumCommandBase ADS Request

Namespace: TwinCAT.Ads.SumCommand [1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

AdsErrorCode Result { get; }

Property Value

Type: AdsErrorCode [575]
The result.
Reference

ISumCommand Interface [1206]

TwinCAT.Ads.SumCommand Namespace [1205]

### 6.5.1.1.4 ISumCommand/SubResults Property

Gets the sub results of the single Sub Requests.

**Namespace:** TwinCAT.Ads.SumCommand [1205]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
AdsErrorCode[] SubResults { get; }
```

**Property Value**

Type: `AdsErrorCode` [575]

The sub results.

**Reference**

ISumCommand Interface [1206]

TwinCAT.Ads.SumCommand Namespace [1205]

### 6.5.1.1.5 ISumCommand/Succeeded Property

Gets a value indicating whether this `ISumCommand` [1206] is succeeded.

**Namespace:** TwinCAT.Ads.SumCommand [1205]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
bool Succeeded { get; }
```

**Property Value**

Type: `Boolean`

true if succeeded; otherwise, false.

**Reference**

ISumCommand Interface [1206]

TwinCAT.Ads.SumCommand Namespace [1205]
6.5.2 ResultSumCommand Class

Result class for an asynchronous TwinCAT.Ads.SumCommand:

Inheritance Hierarchy

System.Object
TwinCAT.Ads.ResultAds
TwinCAT.Ads.SumCommand.ResultSumCommand
TwinCAT.Ads.SumCommand.ResultSumHandles
TwinCAT.Ads.SumCommand.ResultSumReadRaw
TwinCAT.Ads.SumCommand.ResultSumValues

Namespace: TwinCAT.Ads.SumCommand
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class ResultSumCommand : ResultAds

The ResultSumCommand type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result object. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds state is failed. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>SubErrors</td>
<td>Gets the Error codes for the single SumCommands.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds state is succeeded. (Inherited from ResultAds.)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds (Inherited from ResultAds)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.SumCommand Namespace
6.5.2.1 ResultSumCommand Properties

The ResultSumCommand [1210] type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>SubErrors</td>
<td>Gets the Error codes [575] for the single SumCommands.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>

### Reference

ResultSumCommand Class [1210]
TwinCAT.Ads.SumCommand Namespace [1205]

6.5.2.1.1 ResultSumCommand/SubErrors Property

Gets the Error codes [575] for the single SumCommands.

Namespace: TwinCAT.Ads.SumCommand [1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

C#

```csharp
public AdsErrorCode[] SubErrors { get; protected set; }
```

### Property Value

Type: AdsErrorCode [575].

### Reference

ResultSumCommand Class [1210]
TwinCAT.Ads.SumCommand Namespace [1205]

6.5.2.2 ResultSumCommand Methods

The ResultSumCommand [1210] type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code> (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>SetError</code></td>
<td>Sets the error state of this <code>ResultAds</code> (Inherited from <code>ResultAds</code>).</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

Reference

`ResultSumCommand Class [1210]`

`TwinCAT.Ads.SumCommand Namespace [1205]`

### 6.5.3 ResultSumHandles Class

Result class for an asynchronous `SumCreateHandles [1222]`.

Inheritance Hierarchy

```
System.Object
  TwinCAT.Ads.ResultAds [989]
    TwinCAT.Ads.SumCommand.ResultSumCommand [1210]
    TwinCAT.Ads.SumCommand.ResultSumHandles [1211]
    TwinCAT.Ads.SumCommand.ResultSumHandles2 [1214]
```

**Namespace:** TwinCAT.Ads.SumCommand [1205]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

**C#**

```csharp
public class ResultSumHandles : ResultSumCommand
```

The `ResultSumHandles` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ErrorCode</code></td>
<td>Gets the ADS Error code bound to this <code>Result</code> object. (Inherited from <code>ResultAds</code>.)</td>
</tr>
<tr>
<td><code>Failed</code></td>
<td>Gets a value indicating whether the <code>ResultAds</code> state is failed. (Inherited from <code>ResultAds</code>.)</td>
</tr>
<tr>
<td><code>Handles</code></td>
<td>The registered Symbol handles.</td>
</tr>
<tr>
<td><code>SubErrors</code></td>
<td>Gets the <code>Error codes</code> for the single <code>SumCommands</code>. (Inherited from <code>ResultCommand</code>.)</td>
</tr>
<tr>
<td><code>Succeeded</code></td>
<td>Gets a value indicating whether the <code>ResultAds</code> state is succeeded. (Inherited from <code>ResultAds</code>.)</td>
</tr>
</tbody>
</table>
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.SumCommand Namespace [1205]

6.5.3.1 ResultSumHandles Properties

The ResultSumHandles [1212] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Handles</td>
<td>The registered Symbol handles.</td>
</tr>
<tr>
<td>SubErrors</td>
<td>Gets the Error codes [575] for the single SumCommands. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>ResultSumCommand [1210].)</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>

Reference

ResultSumHandles Class [1212]

TwinCAT.Ads.SumCommand Namespace [1205]

6.5.3.1.1 ResultSumHandles.Handles Property

The registered Symbol handles.

Namespace: TwinCAT.Ads.SumCommand [1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

**C#**

```csharp
public uint[] Handles { get; }
```

**Property Value**

Type: **UInt32**.

**Reference**

ResultadoHandles Class [1212]

TwinCAT.Ads.SumCommand Namespace [1205]

### 6.5.3.2 ResultSumHandles Methods

The ResultSumHandles [1212] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td><strong>Object</strong>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this <strong>ResultAds</strong> [989] (Inherited from <strong>ResultAds</strong> [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>

**Reference**

ResultadoHandles Class [1212]

TwinCAT.Ads.SumCommand Namespace [1205]

### 6.5.4 ResultSumHandles2 Class

Result class for an asynchronous **SumCreateHandles** [1222].

**Inheritance Hierarchy**

```
System.Object
  TwinCAT.Ads.ResultAds [989]
    TwinCAT.Ads.SumCommand.ResultSumCommand [1210]
      TwinCAT.Ads.SumCommand.ResultSumHandles [1212]
      TwinCAT.Ads.SumCommand.ResultSumHandles2
```
TwinCAT.Ads Namespaces

Namespace: TwinCAT.Ads.SumCommand
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class ResultSumHandles2 : ResultSumHandles
```

The ResultSumHandles2 type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultSumHandles2</td>
<td>Initializes a new instance of the ResultSumHandles2 class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result object. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds state is failed. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>Handles</td>
<td>The registered Symbol handles. (Inherited from ResultSumHandles.)</td>
</tr>
<tr>
<td>InstancePaths</td>
<td>The symbol/instance paths belonging to the handles and the subErrors.</td>
</tr>
<tr>
<td>SubErrors</td>
<td>Gets the Error codes for the single SumCommands. (Inherited from ResultSumCommand.)</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds state is succeeded. (Inherited from ResultAds.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds object. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads.SumCommand Namespace
6.5.4.1 ResultSumHandles2 Constructor

Initializes a new instance of the ResultSumHandles2 [1214] class.

Namespace: TwinCAT Ads SumCommand [1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public ResultSumHandles2(
    AdsErrorCode complete,
    AdsErrorCode[] subErrors,
    uint[] handles,
    string[] instancePaths
)
```

Parameters

- **complete**
  - Type: TwinCAT.Ads.AdsErrorCode [575]
  - The overall ADS error codes.

- **subErrors**
  - Type: TwinCAT.Ads.AdsErrorCode[] [575]
  - The single ADS SubErrors.

- **handles**
  - Type: System.UInt32
  - The registered handles.

- **instancePaths**
  - Type: System.String
  - The instance paths.

Reference

- ResultSumHandles2 Class [1214]
- TwinCAT.Ads.SumCommand Namespace [1205]

6.5.4.2 ResultSumHandles2 Properties

The ResultSumHandles2 [1214] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Handles</td>
<td>The registered Symbol handles. (Inherited from ResultSumHandles [1212].)</td>
</tr>
<tr>
<td>InstancePaths</td>
<td>The symbol-instance paths belonging to the handles and the subErrors.</td>
</tr>
<tr>
<td>SubErrors</td>
<td>Gets the Error codes [575] for the single SumCommands. (Inherited from ResultSumCommand [1210].)</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>
6.5.4.2.1  ResultSumHandles2.InstancePaths Property

The symbol/instance paths belonging to the handles and the subErrors.

**Namespace:** TwinCAT.Ads.SumCommand

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public string[] InstancePaths { get; }
```

**Property Value**

Type: `String`.

Reference

**ResultSumHandles2 Class**

**TwinCAT.Ads.SumCommand Namespace**

6.5.4.3  ResultSumHandles2 Methods

The `ResultSumHandles2` type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this <code>ResultAds</code> (Inherited from <code>ResultAds</code>).</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

Reference

**ResultSumHandles2 Class**

**TwinCAT.Ads.SumCommand Namespace**
6.5.5 ResultSumReadRaw Class

Result class for an asynchronous SumRead.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [989]
    TwinCAT.Ads.SumCommand.ResultSumCommand [1210]
      TwinCAT.Ads.SumCommand.ResultSumReadRaw

Namespace: TwinCAT.Ads.SumCommand

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

public class ResultSumReadRaw : ResultSumCommand

The ResultSumReadRaw type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ReadBlocks</td>
<td>List of the read raw memory blocks read.</td>
</tr>
<tr>
<td>SubErrors</td>
<td>Gets the Error codes [575] for the single SumCommands. (Inherited from ResultSumCommand [1210].)</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.SumCommand Namespace [1205]
6.5.5.1  ResultSumReadRaw Properties

The `ResultSumReadRaw` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ErrorCode</code></td>
<td>Gets the ADS Error code bound to this <code>Result</code> object. (Inherited from <code>ResultAds</code>.)</td>
</tr>
<tr>
<td><code>Failed</code></td>
<td>Gets a value indicating whether the <code>ResultAds</code> state is failed. (Inherited from <code>ResultAds</code>.)</td>
</tr>
<tr>
<td><code>ReadBlocks</code></td>
<td>List of the read raw memory blocks read.</td>
</tr>
<tr>
<td><code>SubErrors</code></td>
<td>Gets the Error codes for the single SumCommands. (Inherited from <code>ResultSumCommand</code>.)</td>
</tr>
<tr>
<td><code>Succeeded</code></td>
<td>Gets a value indicating whether the <code>ResultAds</code> state is succeeded. (Inherited from <code>ResultAds</code>.)</td>
</tr>
</tbody>
</table>

### Reference

- `ResultSumReadRaw Class` [1218]
- `TwinCAT.Ads.SumCommand Namespace` [1205]

6.5.5.1.1  ResultSumReadRaw.ReadBlocks Property

List of the read raw memory blocks read.

**Namespace**: TwinCAT.Ads.SumCommand [1205]

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public IList<byte[]> ReadBlocks { get; }
```

### Property Value

**Type**: `IList<byte[]>`

### Reference

- `ResultSumReadRaw Class` [1218]
- `TwinCAT.Ads.SumCommand Namespace` [1205]

6.5.5.2  ResultSumReadRaw Methods

The `ResultSumReadRaw` type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

ResultSumReadRaw Class [1218]
TwinCAT.Ads.SumCommand Namespace [1205]

6.5.6 ResultSumValues Class

Result class for an asynchronous SumSymbolRead [1242].

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [989]
    TwinCAT.Ads.SumCommand.ResultSumCommand [1210]
      TwinCAT.Ads.SumCommand.ResultSumValues

Namespace: TwinCAT.Ads.SumCommand [1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class ResultSumValues : ResultSumCommand
```

The ResultSumValues type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>SubErrors</td>
<td>Gets the Error codes [575] for the single SumCommands. (Inherited from ResultSumCommand [1210].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from ResultAds [989])</td>
</tr>
<tr>
<td>Values [1222]</td>
<td>The Read data as marshalled values</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError [997]</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989])</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.SumCommand Namespace [1205]

6.5.6.1 ResultSumValues Properties

The ResultSumValues [1220] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989])</td>
</tr>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989])</td>
</tr>
<tr>
<td>SubErrors [1211]</td>
<td>Gets the Error codes [575] for the single SumCommands. (Inherited from ResultSumCommand [1210])</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989])</td>
</tr>
<tr>
<td>Values [1222]</td>
<td>The Read data as marshalled values</td>
</tr>
</tbody>
</table>

Reference

ResultSumValues Class [1220]

TwinCAT.Ads.SumCommand Namespace [1205]
6.5.6.1.1 ResultSumValues.Values Property

The Read data as marshalled values

**Namespace:** TwinCAT.Ads.SumCommand
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Object[] Values { get; }
```

**Property Value**

Type: `Object`.

**Reference**

ResultSumValues Class [1220]

TwinCAT.Ads.SumCommand Namespace [1205]

6.5.6.2 ResultSumValues Methods

The `ResultSumValues` [1220] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>SetError [997]</td>
<td>Sets the error state of this <code>ResultAds</code> [989] (Inherited from <code>ResultAds</code> [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

**Reference**

ResultSumValues Class [1220]

TwinCAT.Ads.SumCommand Namespace [1205]

6.5.7 SumCreateHandles Class

SumCommandBase for getting variable handles by a set of InstancePaths
Inheritance Hierarchy

System.Object
  SumCommandWrapper.SumReadWrite.
   TwinCAT.Ads.SumCommand.SumCreateHandles

Namespace: TwinCAT.Ads.SumCommand
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class SumCreateHandles : SumCommandWrapper<SumReadWrite>

The SumCreateHandles type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SumCreateHandles(IAdsConnection, IList.String.)</td>
<td>Initializes a new instance of the SumCreateHandles class.</td>
</tr>
<tr>
<td>SumCreateHandles(IAdsConnection, String.)</td>
<td>Initializes a new instance of the SumCreateHandles class.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateHandles</td>
<td>Creates the ADS handles.</td>
</tr>
<tr>
<td>CreateHandlesAsync</td>
<td>Create handles asynchronously.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Examples

Usage of Sum commands with handles (CreateHandles, Read, Write, ReleaseHandles)

/// <summary>
/// Defines the entry point of the application.
/// </summary>
static async void Main(string[] args)
{

Console.WriteLine("\n");
Console.WriteLine("Press [Enter] for start:\n");
Console.ReadLine();

// Parse the command-line arguments
AmsAddress address = ArgParser.Parse(args);

CancellationTokenSource cancelSource = new CancellationTokenSource();
CancellationToken cancel = cancelSource.Token;

using (AdsClient client = new AdsClient())
{
// Connect the AdsClient to the device target.
client.Connect(address);

string[] instancePathList = {"GVL.bVar", "GVL.iCount", "TwinCAT_SystemInfoVarList._AppInfo.ProjectName"};
SumCreateHandles createHandlesCommand = new SumCreateHandles(client, instancePathList);

var resultCreateHandles = await createHandlesCommand.CreateHandlesAsync(cancel);
if (resultCreateHandles.Succeeded)
{
    uint[] handles = resultCreateHandles.Handles;
    Type[] valueTypes = new Type[] { typeof(bool), typeof(short), typeof(string) };
    SumHandleRead readCommand = new SumHandleRead(client, handles, valueTypes);
    var resultRead = await readCommand.ReadAsync(cancel);
    if (resultRead.Succeeded)
    {
        object[] readValues = resultRead.Values;
        for (int i = 0; i < instancePathList.Length; i++)
        {
            Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", instancePathList[i], readValues[i].ToString(), valueTypes[i].Name);
        }
        // Sum Command Write
        SumHandleWrite writeCommand = new SumHandleWrite(client, handles, valueTypes);
        object[] writeValues = new object[] { true, (short)42, "MyNewProjectName" };
        await writeCommand.WriteAsync(writeValues, cancel);
        SumReleaseHandles releaseCommand = new SumReleaseHandles(client, handles);
        await releaseCommand.ReleaseHandlesAsync(cancel);
    }
}

Console.WriteLine("\n");
Console.WriteLine("Press [Enter] for leave:\n");
Console.ReadLine();

Reference

TwinCAT.Ads.SumCommand Namespace [1205]
TwinCAT.Ads.SumCommand.ISumCommand [1206]
TwinCAT.Ads.SumCommand.SumReleaseHandles [1237]
TwinCAT.Ads.SumCommand.SumHandleRead [1228]
TwinCAT.Ads.SumCommand.SumHandleWrite [1232]
6.5.7.1 SumCreateHandles Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SumCreateHandles(IAdsConnection, IList&lt;String&gt;)</td>
<td>Initializes a new instance of the SumCreateHandles class.</td>
</tr>
<tr>
<td>SumCreateHandles(IAdsConnection, String)</td>
<td>Initializes a new instance of the SumCreateHandles class.</td>
</tr>
</tbody>
</table>

Reference

SumCreateHandles Class [1222]

TwinCAT.Ads.SumCommand Namespace [1205]

6.5.7.1.1 SumCreateHandles Constructor (IAdsConnection, IList<String.))

Initializes a new instance of the SumCreateHandles class.

Namespace: TwinCAT.Ads.SumCommand [1205]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public SumCreateHandles(IAdsConnection connection, 
                        IList<string> instancePaths)
```

Parameters

- `connection` Type: TwinCAT.Ads.IAdsConnection [765]
  The connection.
- `instancePaths` Type: System.Collections.Generic.IList<String>
  The instance paths.

Reference

SumCreateHandles Class [1222]

SumCreateHandles Overload [1225]

TwinCAT.Ads.SumCommand Namespace [1205]

6.5.7.1.2 SumCreateHandles Constructor (IAdsConnection, .String.)

Initializes a new instance of the SumCreateHandles class.
Namespace:  TwinCAT.Ads.SumCommand [ 1205]
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public SumCreateHandles(
    IAdsConnection connection,
    string[] instancePaths
)
```

Parameters

- **connection**  
  Type: TwinCAT.Ads.IAdsConnection [ 765]  
  The connection.

- **instancePaths**  
  Type: System.String  
  The instance paths.

Reference

SumCreateHandles Class [ 1222]
SumCreateHandles Overload [ 1225]
TwinCAT.Ads.SumCommand Namespace [ 1205]

### 6.5.7.2 SumCreateHandles Methods

The SumCreateHandles [ 1222] type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateHandles</td>
<td>Creates the ADS handles.</td>
</tr>
<tr>
<td>CreateHandlesAsync</td>
<td>Create handles asynchronously.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

SumCreateHandles Class [ 1222]
TwinCAT.Ads.SumCommand Namespace [ 1205]
### SumCreateHandles.CreateHandles Method

Creates the ADS handles.

**Namespace:** TwinCAT.Ads.SumCommand

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
public uint[] CreateHandles()
```

**Return Value**

Type: `System.UInt32[]`.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsSumCommandException <img src="https://example.com" alt="632" /></td>
<td>SumGetHandles failed!</td>
</tr>
<tr>
<td>AdsSumCommandException <img src="https://example.com" alt="632" /></td>
<td>SumGetHandlesCommand failed!</td>
</tr>
</tbody>
</table>

**Reference**

- SumCreateHandles Class ![1222](https://example.com)
- TwinCAT.Ads.SumCommand Namespace ![1205](https://example.com)
- SumCreateHandles.TryCreateHandles(String.., UInt32.., AdsErrorCode..)
- SumCreateHandles.CreateHandlesAsync(CancellationToken) ![1227](https://example.com)

### SumCreateHandles.CreateHandlesAsync Method

Create handles asynchronously.

**Namespace:** TwinCAT.Ads.SumCommand

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
public Task<ResultSumHandles> CreateHandlesAsync(
    CancellationToken cancel
)
```

**Parameters**

- `cancel`  
  Type: `System.Threading.CancellationToken`  
  The cancellation token.
Return Value

Type: Task<ResultSumHandles [1212].
An asynchronous task that represents the 'ReadWriteRaw' operation and returns a ResultSumHandles [1212]. The overall error return code is contained in the ErrorCode [992] property.

Reference

SumCreateHandles Class [1222]  
TwinCAT.Ads.SumCommand Namespace [1205]  
SumCreateHandles.CreateHandles. [1227]  
SumCreateHandles.TryCreateHandles (.String.., .UInt32.., .AdsErrorCode..)

6.5.8 SumHandleRead Class

Read (primitive, Any) values by Handle SumCommandBase.

Inheritance Hierarchy

System.Object  
SumCommandBase  
SumRead  
TwinCAT.Ads.SumCommand.SumHandleRead

Namespace: TwinCAT.Ads.SumCommand [1205]  
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class SumHandleRead : SumRead

The SumHandleRead type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Read [1230]</td>
<td>Reads the values.</td>
</tr>
<tr>
<td>ReadAsync [1231]</td>
<td>Read the values asynchronously.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryRead [1232]</td>
<td>Tries to read the values of the .</td>
</tr>
</tbody>
</table>
Remarks

This is an ADS Sum Command to access values by handle information. It is always used in combination with `CreateHandles` and related methods. By design (and in contrast to the symbolic access in `SumSymbolRead` and `SumSymbolWrite`), this access method can act only with ADS ANY Type (Primitive) values (disadvantage). The advantage is, that no symbolic information must be loaded before accessing the values, see samples:

Examples

**Usage of Sum commands with handles (CreateHandles, Read, Write, ReleaseHandles)**

```csharp
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    Console.WriteLine("\n\nPress [Enter] for start:\n");
    Console.ReadLine();

    // Parse the command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    using (AdsClient client = new AdsClient())
    {
        // Connect the AdsClient to the device target.
        client.Connect(address);

        string[] instancePathList = {"GVL.bVar", "GVL.iCount", "TwinCAT_SystemInfoVarList._AppInfo.ProjectName"};
        SumCreateHandles createHandlesCommand = new SumCreateHandles(client, instancePathList);
        var resultCreateHandles = await createHandlesCommand.CreateHandlesAsync(cancel);
        if (resultCreateHandles.Succeeded)
        {
            uint[] handles = resultCreateHandles.Handles;
            Type[] valueTypes = new Type[] { typeof(bool), typeof(short), typeof(string) };
            SumHandleRead readCommand = new SumHandleRead(client, handles, valueTypes);
            var resultRead = await readCommand.ReadAsync(cancel);
            if (resultRead.Succeeded)
            {
                object[] readValues = resultRead.Values;
                for (int i = 0; i < instancePathList.Length; i++)
                {
                    Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", instancePathList[i], readValues[i].ToString(), valueTypes[i].Name);
                }

                // Sum Command Write
                SumHandleWrite writeCommand = new SumHandleWrite(client, handles, valueTypes);
                object[] writeValues = new object[] { true, (short)42, "MyNewProjectName" };
                await writeCommand.WriteAsync(writeValues, cancel);
            }
        }

        SumReleaseHandles releaseCommand = new SumReleaseHandles(client, handles);
        await releaseCommand.ReleaseHandlesAsync(cancel);
    }
}
```

Reference

**TwinCAT.Ads.SumCommand Namespace**

[1205]
6.5.8.1 SumHandleRead Methods

The `SumHandleRead` type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Read</strong></td>
<td>Reads the values.</td>
</tr>
<tr>
<td><strong>ReadAsync</strong></td>
<td>Read the values asynchronously.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>TryRead</strong></td>
<td>Tries to read the values of the .</td>
</tr>
</tbody>
</table>

## Reference

- `SumHandleRead Class` [1228]
- `TwinCAT.Ads.SumCommand Namespace` [1205]

### 6.5.8.1.1 SumHandleRead.Read Method

Reads the values.

**Namespace:** TwinCAT.Ads.SumCommand [1205]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Object[] Read()
```
Return Value

Type: `Object`, `System.Object[]`.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsSumCommandException [632]</td>
<td>SumAnyReadByHandleCommand failed!</td>
</tr>
</tbody>
</table>

Reference

- `SumHandleRead Class [1228]`
- `TwinCAT.Ads.SumCommand Namespace [1205]`
- `SumHandleRead.TryRead(Object..., AdsErrorCode...) [1232]`
- `SumHandleRead.ReadAsync(CancellationToken) [1231]`

### 6.5.8.1.2 SumHandleRead.ReadAsync Method

Read the values asynchronously.

**Namespace:** `TwinCAT.Ads.SumCommand [1205]`

**Assembly:** `TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`

**Syntax**

**C#**

```csharp
public Task<ResultSumValues> ReadAsync(
    CancellationToken cancel
)
```

**Parameters**

- `cancel`  
  Type: `System.Threading.CancellationToken`  
  The cancellation token

**Return Value**

Type: `Task<ResultSumValues> [1220]`.

An asynchronous task that represents the 'ReadSymbols' operation and returns a `ResultSumValues [1220]`. The overall error return code is contained in the `ErrorCode [992]` property.

**Reference**

- `SumHandleRead Class [1228]`
- `TwinCAT.Ads.SumCommand Namespace [1205]`
- `SumHandleRead.Read [1230]`
- `SumHandleRead.TryRead(Object..., AdsErrorCode...) [1232]`
6.5.8.1.3 SumHandleRead.TryRead Method

Tries to read the values of the .

**Namespace:** TwinCAT.Ads.SumCommand [1205]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AdsErrorCode TryRead(
    out Object[] values,
    out AdsErrorCode[] returnCodes
)
```

**Parameters**

- **values**
  - Type: `System.Object`..  
  - The values.

- **returnCodes**
  - Type: `TwinCAT.Ads.AdsErrorCode`[575]..
  - The return codes.

**Return Value**

Type: `AdsErrorCode`[575]

`AdsErrorCode`.

**Reference**

- **SumHandleRead Class** [1228]

TwinCAT.Ads.SumCommand Namespace [1205]

TwinCAT.Ads.SumCommand.SumHandleRead [1228]

SumHandleRead.Read [1230]

SumHandleRead.ReadAsync(CancellationToken) [1231]

6.5.9 SumHandleWrite Class

Write any (primitive) values by Handle SumCommandBase.

**Inheritance Hierarchy**

- `System.Object`
  - `SumCommandBase`
  - `SumWrite`
  - `TwinCAT.Ads.SumCommand.SumHandleWrite`

**Namespace:** TwinCAT.Ads.SumCommand [1205]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public class SumHandleWrite : SumWrite
```
The SumHandleWrite type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryWrite</td>
<td>Tries to write the values.</td>
</tr>
<tr>
<td>Write</td>
<td>Writes the values to the Symbols.</td>
</tr>
<tr>
<td>WriteAsync</td>
<td>Write the values asynchronously.</td>
</tr>
</tbody>
</table>

### Remarks

This is an ADS Sum Command to access values by handle information. It is always used in combination with and . By design (and in contrast to the symbolic access in SumSymbolRead, SumSymbolWrite) this access method can act only with ADS ANY Type (Primitive) values (disadvantage). The Advantage is, that no symbolic information must be loaded before accessing the values, see samples:

### Examples

#### Usage of Sum commands with handles (CreateHandles, Read, Write, ReleaseHandles)

```csharp
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    Console.WriteLine(""");
    Console.WriteLine("Press [Enter] for start:");
    Console.ReadLine();

    // Parse the command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    using (AdsClient client = new AdsClient())
    {
        // Connect the AdsClient to the device target.
        client.Connect(address);

        string[] instancePathList = {"GVL.bVar", "GVL.iCount", "TwinCAT_SystemInfoVarList._AppState.Name"};
        SumCreateHandles createHandlesCommand = new SumCreateHandles(client, instancePathList);
        var resultCreateHandles = await createHandlesCommand.CreateHandlesAsync(cancel);

        if (resultCreateHandles.Succeeded)
        {
            uint[] handles = resultCreateHandles.Handles;
            Type[] valueTypes = new Type[] { typeof(bool), typeof(short), typeof(string) };
            SumHandleRead readCommand = new SumHandleRead(client, handles, valueTypes);
            var resultRead = await readCommand.ReadAsync(cancel);
        }
    }
}
```
if (resultRead.Succeeded)
{
    object[] readValues = resultRead.Values;
    for (int i = 0; i < instancePathList.Length; i++)
    {
        Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", instancePathList[i], readValues[i].ToString(), valueTypes[i].Name);
    }
    // Sum Command Write
    SumHandleWrite writeCommand = new SumHandleWrite(client, handles, valueTypes);
    object[] writeValues = new object[] { true, (short)42, "MyNewProjectName" };
    await writeCommand.WriteAsync(writeValues, cancel);
}

SumReleaseHandles releaseCommand = new SumReleaseHandles(client, handles);
await releaseCommand.ReleaseHandlesAsync(cancel);

Console.WriteLine("\nPress \[Enter\] for leave:\n");
Console.ReadLine();

Reference
TwinCAT.Ads SumCommand Namespace [1205]
TwinCAT.Ads SumCommand.SumCreateHandles [1222]
TwinCAT.Ads SumCommand.SumReleaseHandles [1237]
TwinCAT.Ads SumCommand.ISumCommand [1206]
TwinCAT.Ads SumCommand.SumReleaseHandles [1237]
TwinCAT.Ads SumCommand.SumReleaseHandles [1237]
TwinCAT.Ads SumCommand.SumHandleRead [1228]

6.5.9.1 SumHandleWrite Methods

The SumHandleWrite [1232] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryWrite</td>
<td>Tries to write the values.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>Write [1235]</td>
<td>Writes the values to the Symbols.</td>
</tr>
<tr>
<td>WriteAsync [1236]</td>
<td>Write the values asynchronously.</td>
</tr>
</tbody>
</table>

Reference

SumHandleWrite Class [1232]
TwinCAT.Ads.SumCommand Namespace [1205]

6.5.9.1.1 SumHandleWrite.TryWrite Method

Tries to write the values.

Namespace: TwinCAT.Ads.SumCommand [1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryWrite(
    Object[] values,
    out AdsErrorCode[] returnCodes
)
```

Parameters

values Type: .System.Object.
The values (ANY/Primitive types only).

returnCodes Type: .TwinCAT.Ads.AdsErrorCode [1205].
The return codes.

Return Value

Type: AdsErrorCode [1205]
AdsErrorCode.

Reference

SumHandleWrite Class [1232]
TwinCAT.Ads.SumCommand Namespace [1205]
SumHandleWrite.Write(.Object.) [1235]
SumHandleWrite.WriteAsync(.Object,. CancellationToken) [1236]

6.5.9.1.2 SumHandleWrite.Write Method

Writes the values to the Symbols.

Namespace: TwinCAT.Ads.SumCommand [1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public void Write(
    Object[] values
)
```

Parameters

- **values**
  - Type: `System.Object`
  - The Values (Any primitive types only):

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsSumCommandException [632]</td>
<td>SumAnyWriteByHandleCommand failed!</td>
</tr>
</tbody>
</table>

Reference

- **SumHandleWrite Class [1232]**
- **TwinCAT.Ads.SumCommand Namespace [1205]**
- **SumHandleWrite.TryWrite(Object, AdsErrorCode) [1235]**
- **SumHandleWrite.WriteAsync(Object, CancellationToken) [1236]**

### 6.5.9.1.3 SumHandleWrite.WriteAsync Method

Write the values asynchronously.

**Namespace:** TwinCAT.Ads.SumCommand [1205]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultSumCommand> WriteAsync(
    Object[] values,
    CancellationToken cancel
)
```

Parameters

- **values**
  - Type: `System.Object`
  - The values.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.

Return Value

- Type: `Task<ResultSumCommand> [1210]`

An asynchronous task that represents the 'ReadWriteRaw' operation and returns a `ResultSumCommand [1210]`. The overall error return code is contained in the `ErrorCode [992]` property.
6.5.10 **SumReleaseHandles Class**

Release Handles SumCommandBase.

**Inheritance Hierarchy**

System.Object
  SumCommandWrapper.SumWrite.
  TwinCAT.Ads.SumCommand.SumReleaseHandles

**Namespace:** TwinCAT.Ads.SumCommand

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public class SumReleaseHandles : SumCommandWrapper<SumWrite>
```

The `SumReleaseHandles` type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SumReleaseHandles</code></td>
<td>Initializes a new instance of the <code>SumReleaseHandles</code> class.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ReleaseHandles</code></td>
<td>Releases the handles.</td>
</tr>
<tr>
<td><code>ReleaseHandlesAsync</code></td>
<td>Releases the handles asynchronously.</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
### Remarks

Releases the specified ADS handles. Usually used in conjunction with the `SumCreateHandles` and the `SumHandleRead` / `SumHandleWrite` commands.

### Examples

#### Usage of Sum commands with handles (CreateHandles, Read, Write, ReleaseHandles)

```csharp
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    Console.WriteLine("\n");
    Console.WriteLine("Press [Enter] for start: ");
    Console.ReadLine();

    // Parse the command-line arguments
    AmsAddress address = ArgParser.Parse(args);
    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    using (AdsClient client = new AdsClient())
    {
        // Connect the AdsClient to the device target.
        client.Connect(address);

        string[] instancePathList = { "GVL.bVar", "GVL.iCount", "TwinCAT_SystemInfoVarList._AppName" };
        SumCreateHandles createHandlesCommand = new SumCreateHandles(client, instancePathList);

        var resultCreateHandles = await createHandlesCommand.CreateHandlesAsync(cancel);
        if (resultCreateHandles.Succeeded)
        {
            uint[] handles = resultCreateHandles.Handles;
            Type[] valueTypes = new Type[] { typeof(bool), typeof(short), typeof(string) };
            SumHandleRead readCommand = new SumHandleRead(client, handles, valueTypes);
            var resultRead = await readCommand.ReadAsync(cancel);
            if (resultRead.Succeeded)
            {
                object[] readValues = resultRead.Values;
                for (int i = 0; i < instancePathList.Length; i++)
                {
                    Console.WriteLine("Symbol: \{0\} (Value: \{1\}, Type: \{2\})", instancePathList[i], readValues[i].ToString(), valueTypes[i].Name);
                }

                // Sum Command Write
                SumHandleWrite writeCommand = new SumHandleWrite(client, handles, valueTypes);
                object[] writeValues = new object[] { true, (short)42, "MyNewProjectName" };
                await writeCommand.WriteAsync(writeValues, cancel);

                SumReleaseHandles releaseCommand = new SumReleaseHandles(client, handles);
                await releaseCommand.ReleaseHandlesAsync(cancel);
            }
        }
    }

    Console.WriteLine("\n");
    Console.WriteLine("Press [Enter] for leave: ");
    Console.ReadLine();
}
```
Reference

TwinCAT.Ads.SumCommand Namespace [1205]
TwinCAT.Ads.SumCommand.ISumCommand [1206]
TwinCAT.Ads.SumCommand.SumCreateHandles [1222]
TwinCAT.Ads.SumCommand.SumHandleRead [1228]
TwinCAT.Ads.SumCommand.SumHandleWrite [1232]

6.5.10.1 SumReleaseHandles Constructor

Initializes a new instance of the SumReleaseHandles [1237] class.

Namespace: TwinCAT.Ads.SumCommand [1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public SumReleaseHandles(
    IAdsConnection connection,
    uint[] serverHandles)
```

Parameters

- connection
  Type: TwinCAT.Ads.IAdsConnection [765]
  The connection.

- serverHandles
  Type: System.UInt32.
  The handles.

Reference

SumReleaseHandles Class [1237]
TwinCAT.Ads.SumCommand Namespace [1205]

6.5.10.2 SumReleaseHandles Methods

The SumReleaseHandles [1237] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ReleaseHandles</td>
<td>Releases the handles.</td>
</tr>
<tr>
<td>TryReleaseHandles</td>
<td>Tries to Release the Handles</td>
</tr>
<tr>
<td>ReleaseHandlesAsy</td>
<td>Releases the handles asynchronously.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Reference**

SumReleaseHandles Class [1237]

TwinCAT.Ads.SumCommand Namespace [1205]

### 6.5.10.2.1 SumReleaseHandles.ReleaseHandles Method

Releases the handles.

**Namespace:** TwinCAT.Ads.SumCommand [1205]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public void ReleaseHandles()
```

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsSumCommandException [632]</td>
<td>SumReleaseHandlesCommand failed!</td>
</tr>
</tbody>
</table>

**Reference**

SumReleaseHandles Class [1237]

TwinCAT.Ads.SumCommand Namespace [1205]

SumReleaseHandles.TryReleaseHandles(AdsErrorCode_) [1241]

SumReleaseHandles.ReleaseHandlesAsync(CancellationToken) [1240]

### 6.5.10.2.2 SumReleaseHandles.ReleaseHandlesAsync Method

Releases the handles asynchronously.

**Namespace:** TwinCAT.Ads.SumCommand [1205]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public Task<ResultSumCommand> ReleaseHandlesAsync(
    CancellationToken cancel
)
```

Parameters

cancel Type: `System.Threading.CancellationToken`
The cancellation token.

Return Value

Type: `Task`<`ResultSumCommand`>
An asynchronous task that represents the 'ReleaseHandles' operation and returns a `ResultSumCommand`

Reference

`SumReleaseHandles Class` [1237]

`TwinCAT.Ads.SumCommand Namespace` [1205]

6.5.10.2.3 `SumReleaseHandles.TryReleaseHandles Method`

Tries to Release the Handles

Namespace: `TwinCAT.Ads.SumCommand` [1205]
Assembly: `TwinCAT.Ads (in TwinCAT.Ads.dll)` Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryReleaseHandles(
    out AdsErrorCode[] returnCodes
)
```

Parameters

returnCodes Type: `.TwinCAT.Ads.AdsErrorCode` [575]
The return codes.

Return Value

Type: `AdsErrorCode` [575]
`AdsErrorCode`

Reference

`SumReleaseHandles Class` [1237]

`TwinCAT.Ads.SumCommand Namespace` [1205]

`SumReleaseHandles.ReleaseHandles` [1240]

`SumReleaseHandles.ReleaseHandlesAsync(CancellationToken)` [1240]
**6.5.11 SumSymbolRead Class**

Symbolic ADS Sum read access

**Inheritance Hierarchy**

```
System.Object
    SumCommandWrapper.SumRead.
    SumSymbolCommand.SumRead.
    TwinCAT.Ads.SumCommand.SumSymbolRead
```

**Namespace:** TwinCAT.Ads.SumCommand

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public class SumSymbolRead : SumSymbolCommand<SumRead>
```

The `SumSymbolRead` type exposes the following members.

**Constructors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SumSymbolRead</td>
<td>Initializes a new instance of the <code>SumSymbolRead</code> class.</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Read</td>
<td>Reads the Values.</td>
</tr>
<tr>
<td>ReadAsync</td>
<td>Reads Symbol values as an asynchronous operation.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>TryRead</td>
<td>Reads the specified symbols.</td>
</tr>
</tbody>
</table>

**Remarks**

The `SumSymbolRead` implements symbolic read access with automatic (dynamic) value marshalling. The advantage of the symbolic access is (in contrast to the handle access classes `SumHandleRead` [1228], `SumHandleWrite` [1232]) that all type information is available when using this ADS Sum Command. The disadvantage is, that the Symbolic information must be loaded beforehand, see examples.
Examples

Usage of SumSymbolRead/SumSymbolWrite with AdsSession

```csharp
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    Console.WriteLine("\n");
    Console.WriteLine("Press [Enter] for start:";
    Console.ReadLine();

    // Parse the command-line arguments
    AmsAddress address = ArgParser.Parse(args);
    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    using (AdsSession session = new AdsSession(address))
    {
        // Connect to the device target.
        AdsConnection connection = (AdsConnection)session.Connect();
        // Load symbolic information
        var resultSymbols = await session.SymbolServer.GetSymbolsAsync(cancel);
        resultSymbols.ThrowOnError();
        ISymbolCollection<ISymbol> symbols = resultSymbols.Symbols;
        ISymbol bVar1 = symbols["GVL.bVar1"]; // You can use the symbol name here.
        ISymbol bVar2 = symbols["GVL.bVar2"]; // You can use the symbol name here.
        ISymbol projectName = symbols["TwinCAT_SystemInfoVarList._AppInfo.ProjectName"]; // You can use the symbol name here.
        SymbolCollection coll = new SymbolCollection() {bVar1, bVar2, projectName};
        // Sum Command Read
        SumSymbolRead readCommand = new SumSymbolRead(connection, coll);
        var resultReadValues = await readCommand.ReadAsync(cancel);
        if (resultReadValues.Succeeded)
        {
            object[] values = resultReadValues.Values;
            for (int i = 0; i < coll.Count; i++)
            {
                Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", coll[i].InstancePath, values[i].ToString(), values[i].GetType().Name);
            }
        }
        // Sum Command Write
        SumSymbolWrite writeCommand = new SumSymbolWrite(connection, coll);
        object[] writeValues = new object[] {true, (short)42, "MyNewProjectName"};
        var resultWrite = await writeCommand.WriteAsync(writeValues, cancel);
    }
    Console.WriteLine("\n");
    Console.WriteLine("Press [Enter] for leave:";
    Console.ReadLine();
}
```

Examples

Usage of SumSymbolRead/SumSymbolWrite with AdsClient

```csharp
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    Console.WriteLine("\n");
    Console.WriteLine("Press [Enter] for start:";
    Console.ReadLine();
```
CancellationTokenSource cancelSource = new CancellationTokenSource();
CancellationToken cancel = cancelSource.Token;

// Parse the command-line arguments
AmsAddress address = ArgParser.Parse(args);

using (AdsClient client = new AdsClient())
{
    // Connect the AdsClient to the device target.
    client.Connect(address);

    // Load symbolic information
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    var resultReadSymbols = await loader.GetSymbolsAsync(cancel);
    resultReadSymbols.ThrowOnError();

    ISymbolCollection<ISymbol> allSymbols = resultReadSymbols.Symbols;
    ISymbol bVar1 = allSymbols["GVL.bVar1"]; // Symbol "GVL.bVar1"
    ISymbol bVar2 = allSymbols["GVL.iCount"]; // Symbol "GVL.iCount"
    ISymbol projectName = allSymbols["TwinCAT_SystemInfoVarList._AppInfo.ProjectName"]; // Symbol "TwinCAT_SystemInfoVarList._AppInfo.ProjectName"

    SymbolCollection symbols = new SymbolCollection() { bVar1, bVar2, projectName };

    // Sum Command Read
    SumSymbolRead readCommand = new SumSymbolRead(client, symbols);
    var resultSumRead = await readCommand.ReadAsync(cancel);
    if (resultSumRead.Succeeded)
    {
        object[] values = resultSumRead.Values;
        for (int i = 0; i < symbols.Count; i++)
        {
            Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", symbols[i].InstancePath, values[i].ToString(), values[i].GetType().Name);
        }

        // Sum Command Write
        SumSymbolWrite writeCommand = new SumSymbolWrite(client, symbols);
        object[] writeValues = new object[] { true, (short)42, "MyNewProjectName" };
        var resultSumWrite = await writeCommand.WriteAsync(writeValues, cancel);
    }
}

Console.WriteLine("Press [Enter] for leave: ");
Console.ReadLine();

Reference
TwinCAT.Ads SumCommand Namespace [1205]
TwinCAT.Ads SumCommand.ISumCommand [1206]
TwinCAT.Ads SumCommand.SumSymbolWrite [1248]

6.5.11.1 SumSymbolRead Constructor

Initializes a new instance of the SumSymbolRead [1242] class.

Namespace: TwinCAT.Ads.SumCommand [1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffddca3e72bc0ea15da1c14
Syntax

C#

public SumSymbolRead(
  IAdsConnection connection,
  IList<ISymbol> symbols)

Parameters

connection Type: TwinCAT.Ads.IAdsConnection [765]
  The ADS Connection.
symbols Type: System.Collections.Generic.IList<ISymbol> [2176]
  The symbols to read

Reference

SumSymbolRead Class [1242]
TwinCAT.Ads.SumCommand Namespace [1205]

6.5.11.2 SumSymbolRead Methods

The SumSymbolRead [1242] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Read [1245]</td>
<td>Reads the Values.</td>
</tr>
<tr>
<td>ReadAsync [1246]</td>
<td>Reads Symbol values as an asynchronous operation.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryRead [1247]</td>
<td>Reads the specified symbols.</td>
</tr>
</tbody>
</table>

Reference

SumSymbolRead Class [1242]
TwinCAT.Ads.SumCommand Namespace [1205]

6.5.11.2.1 SumSymbolRead.Read Method

Reads the Values.
Namespace: TwinCAT.Ads.SumCommand [1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public Object[] Read()

Return Value

Type: Object.
System.Object[].

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsSumCommandException [632]</td>
<td>SumSymbolRead failed!</td>
</tr>
</tbody>
</table>

Remarks

The return values are automatically marshalled to their appropriate .NET types.

Reference

SumSymbolRead Class [1242]
TwinCAT.Ads.SumCommand Namespace [1205]
SumSymbolRead.TryRead(Object, AdsErrorCode) [1247]
SumSymbolRead.ReadKeyAsync(CancellationToken) [1246]

6.5.11.2.2 SumSymbolRead.ReadKeyAsync Method

Reads Symbol values as an asynchronous operation.

Namespace: TwinCAT.Ads.SumCommand [1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public Task<ResultSumValues> ReadAsync(
    CancellationToken cancel
)

Parameters

cancel Type: System.Threading.CancellationToken
The cancellation token.
Return Value

Type: Task<ResultSumValues> [1220].
An asynchronous task that represents the 'Read' operation and returns a ResultSumValues [1220]. The overall error return code is contained in the ErrorCode [992] property.

Reference

SumSymbolRead Class [1242]
TwinCAT.Ads.SumCommand Namespace [1205]
SumSymbolRead.Read [1245]
SumSymbolRead.TryRead(Object.., AdsErrorCode..) [1247]

6.5.11.2.3 SumSymbolRead.TryRead Method

Reads the specified symbols.

Namespace: TwinCAT.Ads.SumCommand [1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryRead(
    out Object[] values,
    out AdsErrorCode[] returnCodes
)
```

Parameters

values Type: System.Object.. The values.
returnCodes Type: TwinCAT.Ads.AdsErrorCode [575].. The return codes.

Return Value


Remarks

The returned values are automatically marshalled to their appropriate .NET types.

Reference

SumSymbolRead Class [1242]
TwinCAT.Ads.SumCommand Namespace [1205]
SumSymbolRead.Read [1245]
SumSymbolRead.TryReadAsync(CancellationToken) [1246]
6.5.12 SumSymbolWrite Class

Class for ADS Sum symbolic Write Access.

Inheritance Hierarchy

System.Object
   SumCommandWrapper.SumWrite.
   SumSymbolCommand.SumWrite.
   TwinCAT.Ads.SumCommand.SumSymbolWrite

Namespace: TwinCAT.Ads.SumCommand
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public class SumSymbolWrite : SumSymbolCommand<SumWrite>

The SumSymbolWrite type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SumSymbolWrite</td>
<td>Initializes a new instance of the SumSymbolWrite class.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td>TryWrite</td>
<td>Writes the specified values.</td>
</tr>
<tr>
<td>Write</td>
<td>Writes the specified values.</td>
</tr>
<tr>
<td>WriteAsync</td>
<td>Reads the symbol values asynchronously.</td>
</tr>
</tbody>
</table>

Remarks

The SumSymbolWrite implements symbolic write access with automatic (dynamic) value marshalling. The advantage of the symbolic access is (in contrast to the handle access classes SumHandleRead, SumHandleWrite) that all type information is available when using this ADS Sum Command. The disadvantage is, that the Symbolic information must be loaded beforehand, see examples.
Examples

Usage of SumSymbolRead/SumSymbolWrite with AdsSession

```csharp
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    Console.WriteLine("\n");
    Console.WriteLine("Press \[Enter\] for start: \n");
    Console.ReadLine();

    // Parse the command-line arguments
    AmsAddress address = ArgParser.Parse(args);
    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    using (AdsSession session = new AdsSession(address))
    {
        // Connect to the device target.
        AdsConnection connection = (AdsConnection)session.Connect();

        // Load symbolic information
        var resultSymbols = await session.SymbolServer.GetSymbolsAsync(cancel);
        resultSymbols.ThrowOnError();

        ISymbolCollection<ISymbol> symbols = resultSymbols.Symbols;

        ISymbol bVar1 = symbols["GVL.bVar1"];  
        ISymbol bVar2 = symbols["GVL.bVar2"];  
        ISymbol projectName = symbols["TwinCAT_SystemInfoVarList._AppInfo.ProjectName"];  

        SymbolCollection coll = new SymbolCollection() {bVar1, bVar2, projectName};

        // Sum Command Read
        SumSymbolRead readCommand = new SumSymbolRead(connection, coll);
        var resultReadValues = await readCommand.ReadAsync(cancel);
        if (resultReadValues.Succeeded)
        {
            object[] values = resultReadValues.Values;
            for (int i = 0; i < coll.Count; i++)
            {
                Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", coll[i].InstancePath, values[i].ToString(), values[i].GetType().Name);
            }
        }

        // Sum Command Write
        SumSymbolWrite writeCommand = new SumSymbolWrite(connection, coll);
        object[] writeValues = new object[] {true, (short)42, "MyNewProjectName"};
        var resultWrite = await writeCommand.WriteAsync(writeValues, cancel);

        Console.WriteLine("\n");
        Console.WriteLine("Press \[Enter\] for leave: \n");
        Console.ReadLine();
    }
}
```

Examples

Usage of SumSymbolRead/SumSymbolWrite with AdsClient

```csharp
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    Console.WriteLine("\n");
    Console.WriteLine("Press \[Enter\] for start: \n");
    Console.ReadLine();
```
CancellationTokenSource cancelSource = new CancellationTokenSource();
CancellationToken cancel = cancelSource.Token;

// Parse the command-line arguments
AmsAddress address = ArgParser.Parse(args);

using (AdsClient client = new AdsClient())
{
    // Connect the AdsClient to the device target.
    client.Connect(address);

    // Load symbolic information
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    var resultReadSymbols = await loader.GetSymbolsAsync(cancel);
    resultReadSymbols.ThrowOnError();

    ISymbolCollection<ISymbol> allSymbols = resultReadSymbols.Symbols;
    ISymbol bVar1 = allSymbols["GVL.bVar1"];  
    ISymbol bVar2 = allSymbols["GVL.iCount"];  
    ISymbol projectName = allSymbols["TwinCAT_SystemInfoVarList._AppInfo.ProjectName"];  

    SymbolCollection symbols = new SymbolCollection() { bVar1, bVar2, projectName };

    // Sum Command Read
    SumSymbolRead readCommand = new SumSymbolRead(client, symbols);
    var resultSumRead = await readCommand.ReadAsync(cancel);
    if (resultSumRead.Succeeded)
    {
        object[] values = resultSumRead.Values;
        for (int i = 0; i < symbols.Count; i++)
        {
            Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", symbols[i].InstancePath, values[i].ToString(), values[i].GetType().Name);
        }

        // Sum Command Write
        SumSymbolWrite writeCommand = new SumSymbolWrite(client, symbols);
        object[] writeValues = new object[] { true, (short)42, "MyNewProjectName" };
        var resultSumWrite = await writeCommand.WriteAsync(writeValues, cancel);
    }
}

Console.WriteLine("\nPress [Enter] for leave:");
Console.ReadLine();

Reference

TwinCAT.Ads SumCommand Namespace [1205]
TwinCAT.Ads SumCommand ISumCommand [1206]
TwinCAT.Ads.SumCommand.SumSymbolWrite

6.5.12.1 SumSymbolWrite Constructor

Initializes a new instance of the SumSymbolWrite [1248] class.

Namespace: TwinCAT.Ads.SumCommand [1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public SumSymbolWrite(
    IAdsConnection connection,
    IList<ISymbol> symbols
)
```

Parameters

- `connection` Type: `TwinCAT.Ads.IAdsConnection` 
  The ADS Connection.
- `symbols` Type: `System.Collections.Generic.IList<ISymbol>`
  The symbols to read

Reference

- `SumSymbolWrite Class` [1248]
- `TwinCAT.Ads.SumCommand Namespace` [1205]

### 6.5.12.2 SumSymbolWrite Methods

The `SumSymbolWrite` [1248] type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>TryWrite</strong></td>
<td>Writes the specified values.</td>
</tr>
<tr>
<td><strong>Write</strong></td>
<td>Writes the specified values.</td>
</tr>
<tr>
<td><strong>WriteAsync</strong></td>
<td>Reads the symbol values asynchronously.</td>
</tr>
</tbody>
</table>

Reference

- `SumSymbolWrite Class` [1248]
- `TwinCAT.Ads.SumCommand Namespace` [1205]

### 6.5.12.2.1 SumSymbolWrite.TryWrite Method

Writes the specified values.
TwinCAT.Ads Namespaces

Namespace: TwinCAT.Ads.SumCommand

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AdsErrorCode TryWrite(
    Object[] values,
    out AdsErrorCode[] returnCodes
)
```

Parameters

values Type: System.Object
The values.

returnCodes Type: TwinCAT.Ads.AdsErrorCode
The return codes.

Return Value

Type: AdsErrorCode
AdsErrorCode.

Remarks

The written values will be marshalled automatically to their appropriate ADS types.

Reference

SumSymbolWrite Class

TwinCAT.Ads.SumCommand Namespace

SumSymbolWrite.Write(.Object.)

6.5.12.2.2 SumSymbolWrite.Write Method

Writes the specified values.

Namespace: TwinCAT.Ads.SumCommand

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void Write(
    Object[] values
)
```

Parameters

values Type: System.Object
The values.
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsSumCommandException</td>
<td>SumSymbolWrite failed!</td>
</tr>
</tbody>
</table>

### Remarks

The values will be marshalled automatically to their appropriate ADS types.

### Reference

- `SumSymbolWrite Class` [► 1248]
- `TwinCAT.Ads.SumCommand Namespace` [► 1205]
- `SumSymbolWrite.TryWrite(Object, AdsErrorCode)` [► 1251]
- `SumSymbolWrite.WriteAsync(Object, CancellationToken)` [► 1253]

### 6.5.12.2.3 `SumSymbolWrite.WriteAsync Method`

Reads the symbol values asynchronously.

#### Namespace
- `TwinCAT.Ads.SumCommand` [► 1205]

#### Assembly
- `TwinCAT.Ads (in TwinCAT.Ads.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public Task<ResultSumCommand> WriteAsync(
    Object[] values,
    CancellationToken cancel
)
```

#### Parameters

- **values**
  - Type: `System.Object`
  - The values.

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token

#### Return Value

Type: `Task<ResultSumCommand>`

An asynchronous task that represents the 'Write' operation and returns a `ResultSumCommand`. The overall error return code is contained in the `ErrorCode` property.

### Reference

- `SumSymbolWrite Class` [► 1248]
- `TwinCAT.Ads.SumCommand Namespace` [► 1205]
- `SumSymbolWrite.Write(Object)` [► 1252]
- `SumSymbolWrite.TryWrite(Object, AdsErrorCode)` [► 1251]
6.6 TwinCAT.Ads.TcpRouter Namespace

### Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmsTcpIpRouter ![1254]</td>
<td>ADS Router class</td>
</tr>
<tr>
<td>Route ![1281]</td>
<td>Class Route specifies a AMS/ADS Route object.</td>
</tr>
<tr>
<td>RouteCollection ![1292]</td>
<td>Collection of routes.</td>
</tr>
<tr>
<td>RouterException ![1306]</td>
<td>An RouterException ![1306] is thrown on communication errors in the AmsTcpIpRouter ![1254] class.</td>
</tr>
<tr>
<td>RouterNotInitializedException ![1310]</td>
<td>Class RouterNotInitializedException. Implements the RouterException ![1306]</td>
</tr>
<tr>
<td>RouterNotStartedException ![1313]</td>
<td>Class RouterNotStartedException. Implements the RouterException ![1306]</td>
</tr>
<tr>
<td>RouterStatusChangedEventArgs ![1317]</td>
<td>Event Arguments sent when the RouterStatusChanged ![1281] Event. Implements the EventArgs</td>
</tr>
<tr>
<td>StaticRoutesXmlConfigurationBuilderExtension ![1320]</td>
<td>Extension class adding StaticRoutes.xml file reading to the IConfigurationBuilder.</td>
</tr>
<tr>
<td>StaticRoutesXmlConfigurationProvider ![1321]</td>
<td>Class StaticRoutesXmlConfigurationProvider. Implements the IConfigurationProvider</td>
</tr>
<tr>
<td>StaticRoutesXmlConfigurationSource ![1326]</td>
<td>StaticRoutes Configuration Sources Implements the IConfigurationSource</td>
</tr>
</tbody>
</table>

### Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAmRouter ![1273]</td>
<td>Interface IAmRouter</td>
</tr>
</tbody>
</table>

### Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RouterStatus ![1316]</td>
<td>Enum RouterStatus</td>
</tr>
</tbody>
</table>

### 6.6.1 AmsTcpIpRouter Class

ADS Router class
Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TcpRouter.AmsTcpIpRouter

Namespace: TwinCAT.Ads.TcpRouter
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class AmsTcpIpRouter : IAmsRouter
```

The AmsTcpIpRouter type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmsTcpIpRouter(Configuration)</td>
<td>Initializes a new instance of the AmsTcpIpRouter class.</td>
</tr>
<tr>
<td>AmsTcpIpRouter(AmsNetId)</td>
<td>Initializes a new instance of the AmsTcpIpRouter class.</td>
</tr>
<tr>
<td>AmsTcpIpRouter(Log, IConfiguration)</td>
<td>Initializes a new instance of the AmsTcpIpRouter class.</td>
</tr>
<tr>
<td>AmsTcpIpRouter(AmsNetId, Int32, IPAddress, Int32, ILogger)</td>
<td>Initializes a new instance of the AmsTcpIpRouter class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsActive</td>
<td>Gets a value indicating whether the IAmsRouter is active (Running or in Starting / Stopping state).</td>
</tr>
<tr>
<td>IsRunning</td>
<td>Gets a value indicating whether the IAmsRouter is running (Start phase completely finished).</td>
</tr>
<tr>
<td>LocalNetld</td>
<td>Gets the local AmsNetld of this router.</td>
</tr>
<tr>
<td>Loopback</td>
<td>Gets the loopback alias</td>
</tr>
<tr>
<td>Netld</td>
<td>The Local AmsNetld of this router.</td>
</tr>
<tr>
<td>RouterStatus</td>
<td>Gets the router status.</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddRoute</td>
<td>Adds a dynamic Route</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IsRegisteredServer</td>
<td>Determines whether the Address specifies an locally registered AmsServer</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>RemoveRoute(String)</td>
<td>Removes a dynamic Route</td>
</tr>
<tr>
<td>RemoveRoute(AmsNetId)</td>
<td>Removes a dynamic Route</td>
</tr>
<tr>
<td>StartAsync</td>
<td>Starts the AmsTcpIpRouter asynchronously.</td>
</tr>
<tr>
<td>Stop</td>
<td>Stops the AmsTcpIpRouter.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryAddRoute</td>
<td>Adds a dynamic Route</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RouterStatusChanged</td>
<td>Occurs when the router status changes.</td>
</tr>
</tbody>
</table>

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT_TCP_PORT</td>
<td>The default TCP port (0xBF02, 48898)</td>
</tr>
<tr>
<td>TCP_TIMEOUT</td>
<td>The TCP timeout</td>
</tr>
</tbody>
</table>

### Remarks

The AmsTcpIpRouter implements a simple ADS/AMS Router for systems, where no TwinCAT Router installation is available. It opens by default an TCP Loopback port (127.0.0.1) Port BF02 (DEFAULT_TCP_PORT) and routes the ADS/AMS communication between AdsServer and AdsClient. If the addressed target server is not on the local system, the AMS Frames are addressed to remote systems, if the route is appropriately assigned on source and target system (on both sides usually via StaticRoutes.xml).
<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the local system.</td>
<td></td>
</tr>
<tr>
<td>NetId</td>
<td>The AmsNetId of the local system.</td>
<td></td>
</tr>
<tr>
<td>RemoteConnections</td>
<td>List of the registered remote Routes.</td>
<td>None/Empty</td>
</tr>
<tr>
<td>TcpPort</td>
<td>The TCP Port to be used for external ADS communication.</td>
<td>0xBF02 (48898)</td>
</tr>
<tr>
<td>LoopbackIP</td>
<td>The LoopbackIP of Device internal communication. This setting should only be changed if ADS Server and Router application should run on different (virtual) devices. On the AdsServer side an appropriate system configuration via RouterEndPoint [1546] is necessary if the loopback ip is changed.</td>
<td>Loopback 127.0.0.1</td>
</tr>
<tr>
<td>LoopbackPort</td>
<td>The Loopback Port of Device internal ADS communication. This setting should only be changed when different TcpPorts are necessary to separate internal and external router communication (e.g. in WSL2 docker containers) where the LoopbackIP 127.0.0.1 isn’t appropriate. On the AdsServer side an appropriate system configuration via RouterEndPoint [1546] is necessary if the loopback port is changed.</td>
<td>0xBF02 (48898)</td>
</tr>
</tbody>
</table>

The router settings can be configured by the .NET Configuration Builder IConfigurationBuilder or simply by calling one of the AmsTcpIpRouter constructor overloads.

**Xml Configuration**

(StaticRoutes.xml) AddStaticRoutesXmlConfiguration(IConfigurationBuilder) [1320]

**Json Configuration**

(appSettings.json) ![Microsoft.Extensions.Configuration.JsonConfigurationExtensions.AddJsonFile(string)]

**Environment variables**


**ConsoleArguments**


**Examples**

The following sample shows how to use the AmsTcpIpRouter class within an own Console application. This console application can also be accessed as binary from the Nuget.org package repository. The ID of the package is ‘Beckhoff.TwinCAT.Ads.AdsRouterConsole’.

**Ads Router WorkerService**

```csharp
class Program
{
/// <summary>
/// Defines the entry point of the application.
/// </summary>
public static void Main(string[] args)
{
    try
    {
        CreateHostBuilder(args).Build().Run();
    }
```
catch (OperationCanceledException /*cex*/)
{
    Console.WriteLine("Router cancelled!");
}
catch (Exception ex)
{
    Console.WriteLine($"Router failed with '{ex.Message}'");
}

/// <summary>
/// Creates the host builder.
/// </summary>
/// <param name="args">The arguments.</param>
/// <returns>IHostBuilder.</returns>
public static IHostBuilder CreateHostBuilder(string[] args) =>
    Host.CreateDefaultBuilder(args)
    .ConfigureServices(hostContext, services) =>
    {
        services.AddHostedService<RouterService>();
    }
    .ConfigureAppConfiguration((hostingContext, config) =>
    {
        // Add further AppConfigurationProvider here.
        //config.Sources.Clear(); // Clear all default config sources
        //config.AddEnvironmentVariables(""); // Use Environment variables
        //config.AddCommandLine(args); // Use Command Line
        //config.AddJsonFile("appSettings.json"); // Use Appsettings
        config.AddStaticRoutesXmlConfiguration(); // Overriding settings with StaticRoutes.Xml
    }
    .ConfigureLogging(logging =>
    {
        //logging.ClearProviders();
        // Adding console logging here.
        //logging.AddConsole();
    });

    /// <summary>
    /// The RouterService instance represents a long running (hosted) service that implements an <see cref="AmsTcpIpRouter"/>.
    /// Implements the <see cref="Microsoft.Extensions.Hosting.BackgroundService" />
    /// </summary>
    /// <remarks>
    /// Long running Background task that runs a <see cref="AmsTcpIpRouter."/>.
    /// The service is stopped via the <see cref="CancellationToken"/> given to the <see cref="RouterService.ExecuteAsync(CancellationToken)"/> method.
    /// </remarks>
    /// <seealso cref="Microsoft.Extensions.Hosting.BackgroundService" />
    public class RouterService : BackgroundService
    {
        /// <summary>
        /// Logger
        /// </summary>
        private readonly ILogger<RouterService> _logger;

        /// <summary>
        /// Configuration
        /// </summary>
        private readonly IConfiguration _configuration;

        /// <summary>
        /// Initializes a new instance of the <see cref="RouterService"/> class.
        /// </summary>
        /// <param name="logger">The logger.</param>
        /// <param name="configuration">The configuration.</param>
        public RouterService(ILogger<RouterService> logger, IConfiguration configuration)
        {
            _logger = logger;
            _configuration = configuration;
            string value = (string)_configuration.GetValue("ASPNETCORE_ENVIRONMENT", "Production");
        }

        /// <summary>
        /// Execute the Router asynchronously as <see cref="BackgroundService"/>.
        /// </summary>
        /// <param name="cancel">The cancellation token.</param>
        protected override async Task ExecuteAsync(CancellationToken cancel)
AmsTcpIpRouter router;

using (_logger.BeginScope("Starting"))
{
StringBuilder appCommon = new StringBuilder();

appCommon.AppendLine($"ApplicationPath: {Environment.GetCommandLineArgs()[0]}");
//_logger.LogInformation(sB.ToString());
StringBuilder config = new StringBuilder();
string value = (string)_configuration.GetValue("ASPNETCORE_ENVIRONMENT", "Production");
config.AppendLine($"ASPNETCORE_ENVIRONMENT: {value}"); Console.WriteLine("Application Directories");
Console.WriteLine("Configuration");
Console.WriteLine(config);
Console.WriteLine("Press Ctrl + C to shutdown!");
router = new AmsTcpIpRouter(_logger, _configuration); router.RouterStatusChanged += Router_RouterStatusChanged;

// Use this overload to instantiate a Router without support of IHost/IConfigurationProvider support and parametrize by code
// AmsTcpIpRouter router = new AmsTcpIpRouter(new AmsNetId("1.2.3.4.5.6"), AmsTcpIpRouter.DEFAULT_TCP_PORT, IPAddress.Loopback, AmsTcpIpRouter.DEFAULT_TCP_PORT, _logger);
// router.AddRoute(...);
}

Task routerTask = router.StartAsync(cancel); // Start the router

#if ADSSERVER
// AdsServer could be started here:
TestAdsServer testServer = new TestAdsServer(_logger);
_logger.LogInformation("Adding Test Server on port '{testServer.ServerAddress}'"); Task systemServiceTask = testServer.ConnectServerAndWaitAsync(cancel);
await Task.WhenAll(routerTask, systemServiceTask);
#else
await routerTask;
#endif

/// <summary>
/// Handles the RouterStatusChanged event of the <see cref="AmsTcpIpRouter"/>
/// </summary>
/// <param name="sender">The source of the event.</param>
/// <param name="e">The <see cref="RouterStatusChangedEventArgs"/></param>
private void Router_RouterStatusChanged(object? sender, RouterStatusChangedEventArgs e)
{
if (e.RouterStatus == RouterStatus.Started)
{
// From here on, the Router is available to receive Data.
}
}

Reference

TwinCAT.Ads.TcpRouter Namespace [1254]
RouteConfig
TwinCAT.Ams.AmsConfiguration [1543]
### 6.6.1.1 AmsTcpIpRouter Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmsTcpIpRouter(IConfiguration)</td>
<td>Initializes a new instance of the AmsTcpIpRouter [1254] class.</td>
</tr>
<tr>
<td>AmsTcpIpRouter(AmsNetId)</td>
<td>Initializes a new instance of the AmsTcpIpRouter [1254] class.</td>
</tr>
<tr>
<td>AmsTcpIpRouter(I_LOGGER, IConfiguration)</td>
<td>Initializes a new instance of the AmsTcpIpRouter [1254] class.</td>
</tr>
<tr>
<td>AmsTcpIpRouter(AmsNetId, Int32, IPAddress, Int32, ILogger)</td>
<td>Initializes a new instance of the AmsTcpIpRouter [1254] class.</td>
</tr>
</tbody>
</table>

#### Reference

AmsTcpIpRouter Class [1254]

TwinCAT.Ads.TcpRouter Namespace [1254]

### 6.6.1.1.1 AmsTcpIpRouter Constructor (IConfiguration)

Initializes a new instance of the AmsTcpIpRouter [1254] class.

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public AmsTcpIpRouter(IConfiguration configuration)
```

#### Parameters

- **configuration**
  Type: IConfiguration

#### Remarks

The AmsTcpIpRouter [1254] will be connected to the Default port DEFAULT_TCP_PORT [1273].

#### Reference

AmsTcpIpRouter Class [1254]
### AmsTcpIpRouter Constructor (AmsNetId)

Initializes a new instance of the `AmsTcpIpRouter` class.

**Namespace:** TwinCAT.Ads.TcpRouter

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AmsTcpIpRouter(
    AmsNetId local
)
```

**Parameters**

- `local`  
  Type: `TwinCAT.Ads.AmsNetId`  
  The local.

**Remarks**

The `AmsTcpIpRouter` will be connected to the Default port `DEFAULT_TCP_PORT`.

**Reference**

- AmsTcpIpRouter Class
- AmsTcpIpRouter Overload
- TwinCAT.Ads.TcpRouter Namespace

### AmsTcpIpRouter Constructor (ILogger, IConfiguration)

Initializes a new instance of the `AmsTcpIpRouter` class.

**Namespace:** TwinCAT.Ads.TcpRouter

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AmsTcpIpRouter(
    ILogger logger,
    IConfiguration configuration
)
```

**Parameters**

- `logger`  
  Type: `ILogger`  
  The logger.

- `configuration`  
  Type: `IConfiguration`  
  The configuration.
Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception</td>
<td>No IPv4 address for server</td>
</tr>
</tbody>
</table>

Remarks

The `AmsTcpIpRouter` will be connected to the Default port `DEFAULT_TCP_PORT`.

Reference

AmsTcpIpRouter Class [1254]

AmsTcpIpRouter Overload [1260]

TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.1.4 AmsTcpIpRouter Constructor (AmsNetId, Int32, IPAddress, Int32, ILogger)

Initializes a new instance of the `AmsTcpIpRouter` class.

Namespace: TwinCAT.Ads.TcpRouter [1254]

Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AmsTcpIpRouter(
    AmsNetId local,
    int externalPort,
    IPAddress loopbackIP,
    int loopbackPort,
    ILogger logger
)
```

Parameters

- **local**
  - Type: `TwinCAT.Ads.AmsNetId` [665]
  - The (enforced) local NET Id.

- **externalPort**
  - Type: `System.Int32`
  - The TCP/IP port the `AmsTcpIpRouter` is using.

- **loopbackIP**
  - Type: `System.Net.IPAddress`
  - The loopback ip.

- **loopbackPort**
  - Type: `System.Int32`
  - The loopback port.

- **logger**
  - Type: `ILogger`
  - The logger.

Remarks

This constructor doesn't read the 'StaticRoutes.xml'. Instead, it is expected to force the local `AmsNetId` [665], the TCP/IP port and add the routes as Dynamic routes.

Reference

AmsTcpIpRouter Class [1254]
### AmsTcpIpRouter Properties

The `AmsTcpIpRouter` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsActive</td>
<td>Gets a value indicating whether the <code>IAmsRouter</code> is active (Running or in Starting / Stopping state).</td>
</tr>
<tr>
<td>IsRunning</td>
<td>Gets a value indicating whether the <code>IAmsRouter</code> is running (Start phase completely finished).</td>
</tr>
<tr>
<td>LocalNetId</td>
<td>Gets the local <code>AmsNetId</code> of this router.</td>
</tr>
<tr>
<td>Loopback</td>
<td>Gets the loopback alias</td>
</tr>
<tr>
<td>NetId</td>
<td>The Local <code>AmsNetId</code> of this router.</td>
</tr>
<tr>
<td>RouterStatus</td>
<td>Gets the router status.</td>
</tr>
</tbody>
</table>

### AmsTcpIpRouter.IsActive Property

Gets a value indicating whether the `IAmsRouter` is active (Running or in Starting / Stopping state).

**Namespace:** `TwinCAT.Ads.TcpRouter`

**Assembly:** `TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public bool IsActive { get; }
```

**Property Value**

Type: `Boolean`

true if this the router is active; otherwise, false.

**Implements**

`IAmsRouter.IsActive`
6.6.1.2.2 AmsTcpIpRouter.IsRunning Property

```csharp
public bool IsRunning { get; }
```

**Property Value**

Type: `Boolean`

true if this the router is running; otherwise, false.

**Reference**

AmsTcpIpRouter Class [1254]

TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.1.2.3 AmsTcpIpRouter.LocalNetId Property

```csharp
public static AmsNetId LocalNetId { get; }
```

**Property Value**

Type: `AmsNetId` [665]

The local net identifier.

**Remarks**

This is initialized during the execution of the AmsTcpIpRouter [1254] constructor.
6.6.1.2.4 AmsTcpIpRouter.Loopback Property

Gets the loopback alias

**Namespace:** TwinCAT.Ads.TcpRouter

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddc4e3e72bc0ea15da1c14

**Syntax**

```csharp
public IPAddress Loopback { get; }
```

**Property Value**

Type: IPAddress

The loopback.

**Reference**

AmsTcpIpRouter Class [1254]

TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.1.2.5 AmsTcpIpRouter.NetId Property

The Local AmsNetId [665] of this router.

**Namespace:** TwinCAT.Ads.TcpRouter

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddc4e3e72bc0ea15da1c14

**Syntax**

```csharp
public AmsNetId NetId { get; }
```

**Property Value**

Type: AmsNetId [665]

The net identifier.

**Implements**

IAmsRouter.NetId [1276]

**Reference**

AmsTcpIpRouter Class [1254]

TwinCAT.Ads.TcpRouter Namespace [1254]
6.6.1.6  AmsTcpIpRouter.RouterStatus Property

Gets the router status.

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public RouterStatus RouterStatus { get; }
```

**Property Value**

**Type:** RouterStatus [1316]

The router status.

**Implements**

IAmsRouter.RouterStatus [1276]

**Reference**

AmsTcpIpRouter Class [1254]

TwinCAT.Ads.TcpRouter Namespace [1254]

AmsTcpIpRouter.RouterStatusChanged [1272]

6.6.1.3  AmsTcpIpRouter Methods

The AmsTcpIpRouter [1254] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddRoute [1267]</td>
<td>Adds a dynamic Route [1281]</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IsRegisteredServer [1267]</td>
<td>Determines whether the Address specifies an locally registered AmsServer</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>RemoveRoute(String) [1268]</td>
<td>Removes a dynamic Route [1281]</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>RemoveRoute(AmsNetId)</td>
<td>Removes a dynamic Route [1269]</td>
</tr>
<tr>
<td>StartAsync [1270]</td>
<td>Starts the AmsTcpIpRouter [1254] asynchronously.</td>
</tr>
<tr>
<td>Stop [1270]</td>
<td>Stops the AmsTcpIpRouter [1254].</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryAddRoute [1271]</td>
<td>Adds a dynamic Route [1281]</td>
</tr>
</tbody>
</table>

Reference

AmsTcpIpRouter Class [1254]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.1.3.1 AmsTcpIpRouter.AddRoute Method

Adds a dynamic Route [1281]

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public void AddRoute(
    Route route
)

Parameters

route Type: TwinCAT.Ads.TcpRouter.Route [1281]
The route.

Implements

IAmsRouter.AddRoute(Route) [1277]

Reference

AmsTcpIpRouter Class [1254]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.1.3.2 AmsTcpIpRouter.IsRegisteredServer Method

Determines whether the Address specifies an locally registered AmsServer

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public bool IsRegisteredServer(
    AmsAddress serverAddress
)
```

Parameters

serverAddress Type: TwinCAT.Ads.AmsAddress

The server address.

Return Value

Type: Boolean
true if [is server registered] [the specified server address]; otherwise, false.

Reference

AmsTcpIpRouter Class [1254]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.1.3.3 AmsTcpIpRouter.RemoveRoute Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RemoveRoute(String) [1268]</td>
<td>Removes a dynamic Route [1281]</td>
</tr>
<tr>
<td>RemoveRoute(AmsNetId) [1269]</td>
<td>Removes a dynamic Route [1281]</td>
</tr>
</tbody>
</table>

Reference

AmsTcpIpRouter Class [1254]
TwinCAT.Ads.TcpRouter Namespace [1254]

AmsTcpIpRouter.RemoveRoute Method (String)

Removes a dynamic Route [1281]

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool RemoveRoute(
    string route
)
```
Parameters

route

Type: System.String

The route.

Return Value

Type: Boolean

true if removed, false otherwise.

Implements

IAmsRouter.RemoveRoute(String) [1278]

Reference

AmsTcpIpRouter Class [1254]

RemoveRoute Overload [1268]

TwinCAT.Ads.TcpRouter Namespace [1254]

AmsTcpIpRouter.RemoveRoute Method (AmsNetId)

Removes a dynamic Route [1281]

Namespace: TwinCAT.Ads.TcpRouter [1254]

Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public bool RemoveRoute(
    AmsNetId netId
)

Parameters

netId

Type: TwinCAT.Ads.AmsNetId [665]

The NetId of the route.

Return Value

Type: Boolean

true if removed, false otherwise.

Implements

IAmsRouter.RemoveRoute(AmsNetId) [1278]

Reference

AmsTcpIpRouter Class [1254]

RemoveRoute Overload [1268]
6.6.1.3.4 AmsTcpIpRouter.StartAsync Method

Starts the AmsTcpIpRouter as asynchronously.

**Namespace:** TwinCAT.Ads.TcpRouter

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

*C#*

```csharp
public Task StartAsync(
    CancellationToken cancel
)
```

**Parameters**

*cancel* Type: System.Threading.CancellationToken

**Return Value**

Type: Task

**Implements**

IAmsRouter.StartAsync(CancellationToken)

**Remarks**

The asynchronous task runs as long until Stop is called.

**Reference**

AmsTcpIpRouter Class

TwinCAT.Ads.TcpRouter Namespace

6.6.1.3.5 AmsTcpIpRouter.Stop Method

Stops the AmsTcpIpRouter.

**Namespace:** TwinCAT.Ads.TcpRouter

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

*C#*

```csharp
public void Stop()
```
6.6.1.3.6 AmsTcpIpRouter.TryAddRoute Method

Adds a dynamic Route

**Namespace:** TwinCAT.Ads.TcpRouter

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#
```csharp
public bool TryAddRoute(
    Route route
)
```

**Parameters**

- `route` Type: `TwinCAT.Ads.TcpRouter.Route`

**Return Value**

Type: `Boolean`
- true if added, false otherwise.

**Implements**

- `IAmsRouter.TryAddRoute(Route)`

**Reference**

- AmsTcpIpRouter Class
- TwinCAT.Ads.TcpRouter Namespace

6.6.1.4 AmsTcpIpRouter Events

The `AmsTcpIpRouter` type exposes the following members.

**Events**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="1272" alt="RouterStatusChanged" /></td>
<td>Occurs when the router status changes.</td>
</tr>
</tbody>
</table>
6.6.1.4.1 AmsTcpIpRouter.RouterStatusChanged Event

Occurs when the router status changes.

**Namespace:** TwinCAT.Ads.TcpRouter

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public event EventHandler<RouterStatusChangedEventArgs> RouterStatusChanged
```

**Value**

Type: `System.EventHandler<RouterStatusChangedEventArgs>`

**Implements**

`IAmsRouter.RouterStatusChanged`
6.6.1.5.1  AmsTcpIpRouter.DEFAULT_TCP_PORT Field

The default TCP port (0xBF02, 48898)

Namespace:  TwinCAT.Ads.TcpRouter [1254]
Assembly:  TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  

```csharp
public const int DEFAULT_TCP_PORT = 48898
```

Field Value

Type: Int32

Reference

AmsTcpIpRouter Class [1254]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.1.5.2  AmsTcpIpRouter.TCP_TIMEOUT Field

The TCP timeout

Namespace:  TwinCAT.Ads.TcpRouter [1254]
Assembly:  TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  

```csharp
public static int TCP_TIMEOUT
```

Field Value

Type: Int32

Reference

AmsTcpIpRouter Class [1254]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.2  IAmsRouter Interface

Interface IAmsRouter

Namespace:  TwinCAT.Ads.TcpRouter [1254]
Assembly:  TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

**C#**

```csharp
public interface IAmsRouter
```

The **IAmsRouter** type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsActive</td>
<td>Gets a value indicating whether the IAmsRouter is active (Running or in Starting / Stopping state).</td>
</tr>
<tr>
<td>IsRunning</td>
<td>Gets a value indicating whether the IAmsRouter is running (Start phase completely finished).</td>
</tr>
<tr>
<td>NetId</td>
<td>The Local AmsNetId of the IAmsRouter</td>
</tr>
<tr>
<td>RouterStatus</td>
<td>Gets the router status.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddRoute</td>
<td>Adds a dynamic Route</td>
</tr>
<tr>
<td>RemoveRoute(String)</td>
<td>Removes a dynamic Route</td>
</tr>
<tr>
<td>RemoveRoute(AmsNetId)</td>
<td>Removes a dynamic Route</td>
</tr>
<tr>
<td>StartAsync</td>
<td>Starts the IAmsRouter asynchronously.</td>
</tr>
<tr>
<td>Stop</td>
<td>Stops the IAmsRouter.</td>
</tr>
<tr>
<td>TryAddRoute</td>
<td>Adds a dynamic Route</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RouterStatusChanged</td>
<td>Occurs when the router status changes.</td>
</tr>
</tbody>
</table>

### Reference

[TwinCAT.Ads.TcpRouter Namespace](#)
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsActive [1275]</td>
<td>Gets a value indicating whether the IAmsRouter [1273] is active (Running or in Starting / Stopping state).</td>
</tr>
<tr>
<td>IsRunning [1275]</td>
<td>Gets a value indicating whether the IAmsRouter [1273] is running (Start phase completely finished).</td>
</tr>
<tr>
<td>NetId [1276]</td>
<td>The Local AmsNetId [665] of the IAmsRouter [1273]</td>
</tr>
<tr>
<td>RouterStatus [1276]</td>
<td>Gets the router status.</td>
</tr>
</tbody>
</table>

Reference

IAmsRouter Interface [1273]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.2.1  IAmsRouter.IsActive Property

Gets a value indicating whether the IAmsRouter [1273] is active (Running or in Starting / Stopping state).

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
bool IsActive { get; }

Property Value

Type: Boolean
true if this the router is active; otherwise, false.

Reference

IAmsRouter Interface [1273]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.2.1.2  IAmsRouter.IsRunning Property

Gets a value indicating whether the IAmsRouter [1273] is running (Start phase completely finished).

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
bool IsRunning { get; }
**6.6.2.1.3 IAmssRouter.NetId Property**

The Local `AmsNetId` of the `IAmsRouter`.

**Namespace:** TwinCAT.Ads.TcpRouter

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
AmsNetId NetId { get; }
```

**Property Value**

Type: `AmsNetId`

The net identifier.

**Reference**

IAmsRouter Interface

TwinCAT.Ads.TcpRouter Namespace

**6.6.2.1.4 IAmssRouter.RouterStatus Property**

Gets the router status.

**Namespace:** TwinCAT.Ads.TcpRouter

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
RouterStatus RouterStatus { get; }
```

**Property Value**

Type: `RouterStatus`

The router status.

**Reference**

IAmsRouter Interface

TwinCAT.Ads.TcpRouter Namespace
6.6.2.2 IAmsRouter Methods

The IAmsRouter [1273] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddRoute [1277]</td>
<td>Adds a dynamic Route [1281]</td>
</tr>
<tr>
<td>RemoveRoute(String) [1278]</td>
<td>Removes a dynamic Route [1281]</td>
</tr>
<tr>
<td>RemoveRoute(AmsNetId) [1278]</td>
<td>Removes a dynamic Route [1281]</td>
</tr>
<tr>
<td>StartAsync [1279]</td>
<td>Starts the IAmsRouter [1273] asynchronously.</td>
</tr>
<tr>
<td>Stop [1280]</td>
<td>Stops the IAmsRouter [1273].</td>
</tr>
<tr>
<td>TryAddRoute [1280]</td>
<td>Adds a dynamic Route [1281]</td>
</tr>
</tbody>
</table>

Reference

IAmsRouter Interface [1273]

TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.2.2.1 IAmsRouter.AddRoute Method

Adds a dynamic Route [1281]

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
void AddRoute(
    Route route
)

Parameters

route Type: TwinCAT.Ads.TcpRouter.Route [1281]
The route.

Reference

IAmsRouter Interface [1273]

TwinCAT.Ads.TcpRouter Namespace [1254]
### 6.6.2.2.2 IAmssRouter.RemoveRoute Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RemoveRoute(String)</td>
<td>Removes a dynamic Route [1281]</td>
</tr>
<tr>
<td>RemoveRoute(AmsNetId)</td>
<td>Removes a dynamic Route [1281]</td>
</tr>
</tbody>
</table>

#### Reference

IAmsRouter Interface [1273]

TwinCAT.Ads.TcpRouter Namespace [1254]

#### IAmssRouter.RemoveRoute Method (String)

Removes a dynamic Route [1281]

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdcc3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
bool RemoveRoute(
    string route
)
```

**Parameters**

- **route**
  - Type: `System.String`
  - The route.

**Return Value**

- Type: `Boolean`
  - `true` if removed, `false` otherwise.

**Reference**

IAmsRouter Interface [1273]

RemoveRoute Overload [1278]

TwinCAT.Ads.TcpRouter Namespace [1254]

#### IAmssRouter.RemoveRoute Method (AmsNetId)

Removes a dynamic Route [1281]
Namespace: TwinCAT.Ads.TcpRouter

Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

bool RemoveRoute(AmsNetId netId)

Parameters

netId Type: TwinCAT.Ads.AmsNetId

The NetId of the route.

Return Value

Type: Boolean

true if removed, false otherwise.

Reference

IAmsRouter Interface

RemoveRoute Overload

TwinCAT.Ads.TcpRouter Namespace

6.6.2.2.3 IAmSRouter.StartAsync Method

Starts the IAmSRouter asynchronously.

Namespace: TwinCAT.Ads.TcpRouter

Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

Task StartAsync(CancellationToken cancel)

Parameters

cancel Type: System.Threading.CancellationToken

Return Value

Type: Task

Task.

Remarks

The asynchronous task runs as long until Stop is called.
6.6.2.2.4   **IAMSRouter.Stop Method**

Stops the *IAMSRouter*. 

**Namespace:** TwinCAT.Ads.TcpRouter 

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
void Stop()
```

6.6.2.2.5   **IAMSRouter.TryAddRoute Method**

Adds a dynamic *Route*. 

**Namespace:** TwinCAT.Ads.TcpRouter 

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
bool TryAddRoute(
    Route route
)
```

**Parameters**

- **route**
  - Type: TwinCAT.Ads.TcpRouter.Route 
  - The route.

**Return Value**

- Type: **Boolean**
  - true if added, false otherwise.
6.6.2.3  IAmRouter Events

The IAmRouter [1273] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RouterStatusChanged [1281]</td>
<td>Occurs when the router status changes.</td>
</tr>
</tbody>
</table>

Reference

IAmRouter Interface [1273]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.2.3.1  IAmRouter.RouterStatusChanged Event

Occurs when the router status changes.

Namespace:  TwinCAT.Ads.TcpRouter [1254]
Assembly:  TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
event EventHandler<RouterStatusChangedEventArgs> RouterStatusChanged
```

Value

Type:  System.EventHandler<RouterStatusChangedEventArgs>[1317].

Reference

IAmRouter Interface [1273]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.3  Route Class

Class Route specifies an AMS/ADS Route object.

Inheritance Hierarchy

System.Object
TwinCAT.Ads.TcpRouter.Route

Namespace:  TwinCAT.Ads.TcpRouter [1254]
Assembly:  TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class Route
```
The Route type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route(String, AmsNetId, IList.IPAddress.)</td>
<td>Initializes a new instance of the Route class.</td>
</tr>
<tr>
<td>Route(String, AmsNetId, String)</td>
<td>Initializes a new instance of the Route class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Gets the address of the Route.</td>
</tr>
<tr>
<td>HostName</td>
<td>Gets the Host name of the route.</td>
</tr>
<tr>
<td>IPAddresses</td>
<td>Gets the resolved IP Addresses belonging to this Route.</td>
</tr>
<tr>
<td>IsHostNameAddressed</td>
<td>Gets a value indicating whether this instance is an HostName Route.</td>
</tr>
<tr>
<td>IsIPAddressed</td>
<td>Gets a value indicating whether the Route address specifies an IPAddress.</td>
</tr>
<tr>
<td>IsResolved</td>
<td>Gets a value indicating whether the IPAddresses are resolved.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Route.</td>
</tr>
<tr>
<td>NetId</td>
<td>Gets the AmsNetId of the route.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to this instance. (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Returns a hash code for this instance. (Overrides Object.GetHashCode.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Overrides Object.ToString.)</td>
</tr>
</tbody>
</table>
Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equality [1291]</td>
<td>Implements the == operator.</td>
</tr>
<tr>
<td>Inequality [1291]</td>
<td>Implements the != operator.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.3.1 Route Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route(String, AmsNetId, IList.IPAddress.) [1283]</td>
<td>Initializes a new instance of the Route [1281] class.</td>
</tr>
<tr>
<td>Route(String, AmsNetId, String) [1284]</td>
<td>Initializes a new instance of the Route [1281] class.</td>
</tr>
</tbody>
</table>

Reference

Route Class [1281]

TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.3.1.1 Route Constructor (String, AmsNetId, IList.IPAddress.)

Initializes a new instance of the Route [1281] class.

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Route(
    string name,
    AmsNetId netId,
    IList<IPAddress> ips
)
```

Parameters

name Type: System.String
The name of the route.
netId

Type: TwinCAT.Ads.AmsNetId [665]
The net identifier.

ips

Type: System.Collections.Generic.IList[System.IPAddress].
The ips.

6.6.3.1.2 Route Constructor (String, AmsNetId, String)

Initializes a new instance of the Route [1281] class.

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public Route(
    string name,
    AmsNetId netId,
    string address
)

Parameters

name Type: System.String
The name.

netId Type: TwinCAT.Ads.AmsNetId [665]
The AmsNetId of the route.

address Type: System.String
The address (HostName or IP)

Reference

Route Class [1281]
Route Overload [1283]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.3.2 Route Properties

The Route [1281] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address [1285]</td>
<td>Gets the address of the Route [1281].</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HostName [1285]</td>
<td>Gets the Host name of the route.</td>
</tr>
<tr>
<td>IPAddresses [1286]</td>
<td>Gets the resolved IP Addresses belonging to this Route [1281].</td>
</tr>
<tr>
<td>IsHostNameAddressed [1286]</td>
<td>Gets a value indicating whether this instance is an HostName Route [1281].</td>
</tr>
<tr>
<td>IsIPAddressed [1287]</td>
<td>Gets a value indicating whether the Route address specifies an IPAddress.</td>
</tr>
<tr>
<td>IsResolved [1287]</td>
<td>Gets a value indicating whether the IPAddresses are resolved.</td>
</tr>
<tr>
<td>Name [1288]</td>
<td>Gets the name of the Route [1281].</td>
</tr>
<tr>
<td>NetId [1288]</td>
<td>Gets the AmsNetId [665] of the route.</td>
</tr>
</tbody>
</table>

### Reference

**Route Class [1281]**

**TwinCAT.Ads.TcpRouter Namespace [1254]**

#### 6.6.3.2.1 Route.Address Property

Gets the address of the Route [1281].

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public string Address { get; }
```

**Property Value**

*Type:* String

The address.

**Remarks**

The Address can be the IPAddress / or the host name of the target system.

**Reference**

**Route Class [1281]**

**TwinCAT.Ads.TcpRouter Namespace [1254]**

#### 6.6.3.2.2 Route.HostName Property

Gets the Host name of the route.
### Route.HostName Property

The name of the host.

**Type:** String

**Reference**

- [Route Class](#)
- [TwinCAT.Ads.TcpRouter Namespace](#)

### Route.IPAddresses Property

Gets the resolved IP Addresses belonging to this Route.

**Namespace:** TwinCAT.Ads.TcpRouter

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd2ca3e72bc0ea15da1c14

**Syntax**

```
C#
public IPAddress[] IPAddresses { get; }
```

**Property Value**

- **Type:** IPAddress

**Reference**

- [Route Class](#)
- [TwinCAT.Ads.TcpRouter Namespace](#)

### Route.IsHostNameAddressed Property

Gets a value indicating whether this instance is an HostName Route.

**Namespace:** TwinCAT.Ads.TcpRouter

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd2ca3e72bc0ea15da1c14

**Syntax**

```
C#
public bool IsHostNameAddressed { get; }
```
Property Value

Type: Boolean
true if this instance is an HostName Addressed route; otherwise, false.

Reference

Route Class [1281]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.3.2.5 Route.IsIPAddressed Property

Gets a value indicating whether the Route address specifies an IPAddress

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool IsIPAddressed { get; }

Property Value

Type: Boolean
true if the Address is an IPAddress; otherwise when its specified as HostAddress false.

Reference

Route Class [1281]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.3.2.6 Route.IsResolved Property

Gets a value indicating whether the IPAddresses are resolved.

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool IsResolved { get; }

Property Value

Type: Boolean
true if the IP Addresses are resolved; otherwise, false.

Reference

Route Class [1281]
TwinCAT.Ads.TcpRouter Namespace [1254]
6.6.3.2.7 Route.Name Property

Gets the name of the Route [1281].

**Namespace:** TwinCAT.Ads.TcpRouter [1254]
**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public string Name { get; }
```

**Property Value**

- **Type:** `String`
The name.

**Reference**

- **Route Class [1281]**
- **TwinCAT.Ads.TcpRouter Namespace [1254]**

6.6.3.2.8 Route.NetId Property

Gets the AmsNetId [665] of the route.

**Namespace:** TwinCAT.Ads.TcpRouter [1254]
**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AmsNetId NetId { get; }
```

**Property Value**

- **Type:** `AmsNetId [665]`
The AmsNetId.

**Remarks**

The `AmsNetId [665]` is the unique identifier used for the route.

**Reference**

- **Route Class [1281]**
- **TwinCAT.Ads.TcpRouter Namespace [1254]**

6.6.3.3 Route Methods

The Route [1281] type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals ![1289]</td>
<td>Determines whether the specified Object is equal to this instance. (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode ![1289]</td>
<td>Returns a hash code for this instance. (Overrides Object.GetHashCode.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString ![1290]</td>
<td>Returns a String that represents this instance. (Overrides Object.ToString.)</td>
</tr>
</tbody>
</table>

Reference

Route Class ![1281]

TwinCAT.Ads.TcpRouter Namespace ![1254]

6.6.3.3.1 Route.Equals Method

Determines whether the specified Object is equal to this instance.

Namespace: TwinCAT.Ads.TcpRouter ![1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public override bool Equals(
    Object obj
)
```

Parameters

obj Type: System.Object
The object to compare with the current object.

Return Value

Type: Boolean
ture if the specified Object is equal to this instance; otherwise, false.

Reference

Route Class ![1281]

TwinCAT.Ads.TcpRouter Namespace ![1254]

6.6.3.3.2 Route.GetHashCode Method

Returns a hash code for this instance.
### Namespace: TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public override int GetHashCode()
```

#### Return Value

Type: `int`

A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

#### Reference

- [Route Class [1281]](TwinCAT.AdsTcpRouter)
- [TwinCAT.Ads.TcpRouter Namespace [1254]](TwinCAT.Ads.TcpRouter)

### 6.6.3.3 Route.ToString Method

Returns a `String` that represents this instance.

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public override string ToString()
```

#### Return Value

Type: `String`

A `String` that represents this instance.

#### Reference

- [Route Class [1281]](TwinCAT.AdsTcpRouter)
- [TwinCAT.Ads.TcpRouter Namespace [1254]](TwinCAT.Ads.TcpRouter)

### 6.6.3.4 Route Operators

The `Route` [1281] type exposes the following members.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>Equality</td>
<td>Implements the <code>==</code> operator.</td>
</tr>
</tbody>
</table>


### Inequality Operator

Implements the \(!=\) operator.

**Reference**

Route Class \[1281\]

TwinCAT.Ads.TcpRouter Namespace \[1254\]

#### 6.6.3.4.1 Route.Equality Operator

Implements the \(==\) operator.

**Namespace:** TwinCAT.Ads.TcpRouter \[1254\]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public static bool operator ==(Route o1, Route o2)
```

**Parameters**

- `o1`: Type: TwinCAT.Ads.TcpRouter.Route \[1281\]
  - The o1.

- `o2`: Type: TwinCAT.Ads.TcpRouter.Route \[1281\]
  - The o2.

**Return Value**

Type: `Boolean`

The result of the operator.

**Reference**

Route Class \[1281\]

TwinCAT.Ads.TcpRouter Namespace \[1254\]

#### 6.6.3.4.2 Route.Inequality Operator

Implements the \(!=\) operator.

**Namespace:** TwinCAT.Ads.TcpRouter \[1254\]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public static bool operator !=(
    Route o1,
    Route o2
)
```

Parameters

- **o1**
  - Type: `TwinCAT.Ads.TcpRouter.Route` |
  - The `o1`.

- **o2**
  - Type: `TwinCAT.Ads.TcpRouter.Route` |
  - The `o2`.

Return Value

- Type: `Boolean`
  - The result of the operator.

Reference

- **Route Class [1281]**
- **TwinCAT.Ads.TcpRouter Namespace [1254]**

### 6.6.4 RouteCollection Class

Collection of routes.

Inheritance Hierarchy

- **System.Object**
- **TwinCAT.Ads.TcpRouter.RouteCollection**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>RouteCollection</code></td>
<td>Initializes a new instance of the RouteCollection class.</td>
</tr>
<tr>
<td><code>RouteCollection(IEnumerable&lt;Route&gt;)</code></td>
<td>Initializes a new instance of the RouteCollection class (copy constructor)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [1295]</td>
<td>Gets the number of elements contained in the ICollection.T.</td>
</tr>
<tr>
<td>IsReadOnly [1296]</td>
<td>Gets a value indicating whether the ICollection.T is read-only.</td>
</tr>
<tr>
<td>Item [1296]</td>
<td>Gets or sets the Route [1281] at the specified index.</td>
</tr>
</tbody>
</table>

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add [1298]</td>
<td>Adds an item to the ICollection.T.</td>
</tr>
<tr>
<td>AddRange [1299]</td>
<td>Adds a range of routes.</td>
</tr>
<tr>
<td>Clear [1299]</td>
<td>Removes all items from the ICollection.T.</td>
</tr>
<tr>
<td>Clone [1299]</td>
<td>Clones this instance.</td>
</tr>
<tr>
<td>Contains [1300]</td>
<td>Determines whether the ICollection.T contains a specific value.</td>
</tr>
<tr>
<td>CopyTo [1300]</td>
<td>Copies the elements of the ICollection.T to an Array, starting at a particular Array index.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator [1301]</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf [1302]</td>
<td>Determines the index of a specific item in the IList.T.</td>
</tr>
<tr>
<td>Insert [1302]</td>
<td>Inserts an item to the IList.T at the specified index.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove(String) [1303]</td>
<td>Removes the specified route from the RouteCollection.</td>
</tr>
<tr>
<td>Remove(AmsNetId) [1304]</td>
<td>Removes the specified AmsNetId [665] from the RouteCollection.</td>
</tr>
<tr>
<td>Remove(Route) [1304]</td>
<td>Removes the first occurrence of a specific object from the ICollection.T.</td>
</tr>
<tr>
<td>RemoveAt [1305]</td>
<td>Removes the IList.T item at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryAdd [1306]</td>
<td>Tries to add the route to the RouteCollection.</td>
</tr>
</tbody>
</table>

## Reference

*TwinCAT.Ads.TcpRouter Namespace [1254]*
6.6.4.1 RouteCollection Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RouteCollection.</td>
<td>Initializes a new instance of the RouteCollection class.</td>
</tr>
<tr>
<td>RouteCollection(IEnumerable.Route.)</td>
<td>Initializes a new instance of the RouteCollection class (copy constructor)</td>
</tr>
</tbody>
</table>

Reference

RouteCollection Class [1292]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.4.1.1 RouteCollection Constructor

Initializes a new instance of the RouteCollection class.

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public RouteCollection()
```

Reference

RouteCollection Class [1292]
RouteCollection Overload [1294]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.4.1.2 RouteCollection Constructor (IEnumerable.Route.)

Initializes a new instance of the RouteCollection class (copy constructor)

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public RouteCollection(
    IEnumerable<Route> routes
)
```
Parameters

routes

Type: System.Collections.Generic.IEnumerable<Route>
The routes.

Reference

RouteCollection Class [1292]
RouteCollection Overload [1294]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.4.2 RouteCollection Properties

The RouteCollection [1292] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [1295]</td>
<td>Gets the number of elements contained in the ICollection.T.</td>
</tr>
<tr>
<td>IsReadOnly [1296]</td>
<td>Gets a value indicating whether the ICollection.T is read-only.</td>
</tr>
<tr>
<td>Item [1296]</td>
<td>Gets or sets the Route [1281] at the specified index.</td>
</tr>
</tbody>
</table>

Reference

RouteCollection Class [1292]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.4.2.1 RouteCollection.Count Property

Gets the number of elements contained in the ICollection.T.

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int Count { get; }
```

Property Value

Type: Int32
The count.

Implements

ICollection<T>.Count
6.6.4.2.2 RouteCollection.IsReadOnly Property

Gets a value indicating whether the ICollection{T} is read-only.

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool IsReadOnly { get; }
```

**Property Value**

Type: Boolean

true if this instance is read only; otherwise, false.

**Implements**

ICollection{T}.IsReadOnly

6.6.4.2.3 RouteCollection.Item Property

Gets or sets the Route[1281] at the specified index.

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Route this[ int index ] { get; set; }
```

**Parameters**

index Type: System.Int32

The index.
Return Value

Type: Route

Implements

IList<T>.Item.Int32.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

RouteCollection Class

TwinCAT.Ads.TcpRouter Namespace

6.6.4.3 RouteCollection Methods

The RouteCollection type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>AddRange</td>
<td>Adds a range of routes.</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>Clone</td>
<td>Clones this instance.</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the ICollection&lt;T&gt; contains a specific value.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection&lt;T&gt; to an Array, starting at a particular Array index.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList&lt;T&gt;.</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList&lt;T&gt; at the specified index.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### RouteCollection.Add Method

Adds an item to the `ICollection<T>`.

**Namespace:** TwinCAT.Ads.TcpRouter

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>public void Add(</td>
<td></td>
</tr>
<tr>
<td>Route item )</td>
<td></td>
</tr>
</tbody>
</table>

**Parameters**

- **item**
  
  Type: TwinCAT.Ads.TcpRouter.Route

**Implements**

- ICollection<T>.Add(T)

**Exceptions**

**Exception** | **Condition**
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>Cannot add item</td>
</tr>
</tbody>
</table>

**Reference**

RouteCollection Class

TwinCAT.Ads.TcpRouter Namespace
6.6.4.3.2 RouteCollection.AddRange Method

Adds a range of routes.

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void AddRange(
    IEnumerable<Route> routes)
```

**Parameters**

routes

Type: System.Collections.Generic.IEnumerable<Route> [1281].
The routes.

**Reference**

RouteCollection Class [1292]

TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.4.3.3 RouteCollection.Clear Method

Removes all items from the ICollection<T>.

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void Clear()
```

**Implements**

ICollection<T>.Clear.

**Reference**

RouteCollection Class [1292]

TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.4.3.4 RouteCollection.Clone Method

Clones this instance.

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
6.6.4.3.5 **RouteCollection.Contains Method**

Determines whether the ICollection{T} contains a specific value.

**Namespace:** TwinCAT.Ads.TcpRouter

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public bool Contains(Route item)
```

**Parameters**

- **item**  
  Type: TwinCAT.Ads.TcpRouter.Route

**Return Value**

Type: Boolean

- if item is found in the ICollection{T}; otherwise, ..

**Implements**

ICollection{T}.Contains(T)

**Reference**

RouteCollection Class

TwinCAT.Ads.TcpRouter Namespace

6.6.4.3.6 **RouteCollection.CopyTo Method**

Copies the elements of the ICollection{T} to an Array, starting at a particular Array index.
Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
```csharp
public void CopyTo(
    Route[] array,
    int arrayIndex
)
```

Parameters

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>array</td>
<td>The one-dimensional Array that is the destination of the elements copied from ICollection&lt;T&gt;. The Array must have zero-based indexing.</td>
</tr>
<tr>
<td>arrayIndex</td>
<td>The zero-based index in array at which copying begins.</td>
</tr>
</tbody>
</table>

Implements

ICollection<T>.CopyTo(T, Int32)

Reference

RouteCollection Class [1292]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.4.3.7 RouteCollection.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
```csharp
public IEnumerator<Route> GetEnumerator()
```

Return Value

Type: IEnumerable<Route> [1281].
An enumerator that can be used to iterate through the collection.

Implements

IEnumerable<T>.GetEnumerator.

Reference

RouteCollection Class [1292]
6.6.4.3.8 RouteCollection.IndexOf Method

Determines the index of a specific item in the `IList<T>`.

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public int IndexOf(
    Route item
)
```

**Parameters**

- `item`: Type: TwinCAT.Ads.TcpRouter.Route [1281]  
  The object to locate in the `IList<T>`.

**Return Value**

- Type: `Int32`  
  The index of item if found in the list; otherwise, -1.

**Implements**

- `IList<T>.IndexOf(T)`

**Reference**

RouteCollection Class [1292]

TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.4.3.9 RouteCollection.Insert Method

Inserts an item to the `IList<T>` at the specified index.

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public void Insert(
    int index,
    Route item
)
```
**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>System.Int32</td>
<td>The zero-based index at which item should be inserted.</td>
</tr>
<tr>
<td>item</td>
<td>TwinCAT.Ads.TcpRouter.Route</td>
<td>The object to insert into the IList&lt;T&gt;.</td>
</tr>
</tbody>
</table>

**Implements**

IList<T>.Insert(Int32, T)

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>Cannot add item</td>
</tr>
</tbody>
</table>

**Reference**

RouteCollection Class [► 1292]

TwinCAT.Ads.TcpRouter Namespace [► 1254]

### 6.6.4.3.10 RouteCollection.Remove Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove(String)</td>
<td>Removes the specified route from the RouteCollection.</td>
</tr>
<tr>
<td>Remove(AmsNetId)</td>
<td>Removes the specified AmsNetId from the RouteCollection.</td>
</tr>
<tr>
<td>Remove(Route)</td>
<td>Removes the first occurrence of a specific object from the ICollection&lt;T&gt;.</td>
</tr>
</tbody>
</table>

**Reference**

RouteCollection Class [► 1292]

TwinCAT.Ads.TcpRouter Namespace [► 1254]

**RouteCollection.Remove Method (String)**

Removes the specified route from the RouteCollection.

**Namespace:** TwinCAT.Ads.TcpRouter [► 1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool Remove(
    string routeName
)
```
**Parameters**

routeName

Type: `System.String`

Name of the route.

**Return Value**

Type: `Boolean`

true if removed, false otherwise.

**Reference**

RouteCollection Class [1292]

Remove Overload [1303]

TwinCAT.Ads.TcpRouter Namespace [1254]

---

**RouteCollection.Remove Method (AmsNetId)**

Removes the specified `AmsNetId` from the RouteCollection [1292].

Namespace: TwinCAT.Ads.TcpRouter [1254]

Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool Remove(
    AmsNetId netId
)
```

**Parameters**

netId

Type: `TwinCAT.Ads.AmsNetId` [665]

The Ams Net ID.

**Return Value**

Type: `Boolean`

true if removed, false otherwise.

**Reference**

RouteCollection Class [1292]

Remove Overload [1303]

TwinCAT.Ads.TcpRouter Namespace [1254]

---

**RouteCollection.Remove Method (Route)**

Removes the first occurrence of a specific object from the ICollection<T>.

Namespace: TwinCAT.Ads.TcpRouter [1254]

Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

```csharp
public bool Remove(
    Route item
)
```

Parameters

- `item` Type: TwinCAT.Ads.TcpRouter.Route
  The object to remove from the ICollection.T.

Return Value

Type: Boolean
- if item was successfully removed from the ICollection.T; otherwise, .. This method also returns .. if item is not found in the original ICollection.T.

Implements

ICollection.T.Remove(T)

Reference

RouteCollection Class
Remove Overload
TwinCAT.Ads.TcpRouter Namespace

6.6.4.3.11 RouteCollection.RemoveAt Method

Removes the IList.T. item at the specified index.

Namespace: TwinCAT.Ads.TcpRouter
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void RemoveAt(
    int index
)
```

Parameters

- `index` Type: System.Int32
  The zero-based index of the item to remove.

Implements

IList.T.RemoveAt(Int32)
6.6.4.3.12 RouteCollection.TryAdd Method

Tries to add the route to the RouteCollection.

Namespace: TwinCAT.Ads.TcpRouter
Assembly: TwinCAT.Ads.TcpRouter.dll
Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool TryAdd(Route route)
```

Parameters

route Type: TwinCAT.Ads.TcpRouter.Route

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

RouteCollection Class
TwinCAT.Ads.TcpRouter Namespace

6.6.5 RouterException Class

An RouterException is thrown on communication errors in the AmsTcpIpRouter class.

Inheritance Hierarchy

System.Object
System.Exception
TwinCAT.Ads.TcpRouter.RouterException
TwinCAT.Ads.TcpRouter.RouterNotInitializedException
TwinCAT.Ads.TcpRouter.RouterNotStartedException

Namespace: TwinCAT.Ads.TcpRouter
Assembly: TwinCAT.Ads.TcpRouter.dll
Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
[SerializableAttribute]
public class RouterException : Exception
```

Reference

RouteCollection Class
TwinCAT.Ads.TcpRouter Namespace
The RouterException type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RouterException</td>
<td>Initializes a new instance of the RouterException class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.TcpRouter Namespace [6.1254]

6.6.5.1 RouterException Constructor

Initializes a new instance of the RouterException [6.1306] class.

Namespace: TwinCAT.Ads.TcpRouter [6.1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

protected RouterException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)

Parameters

serialInfoInfo  Type: System.Runtime.Serialization.SerializationInfo
    The serialization information.
streamingContext  Type: System.Runtime.Serialization.StreamingContext
    The streaming context.

Reference

RouterException Class [6.1306]
TwinCAT.Ads.TcpRouter Namespace [6.1254]

6.6.5.2 RouterException Properties

The RouterException [6.1306] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

**Reference**

RouterException Class [1306]

TwinCAT.Ads.TcpRouter Namespace [1254]

### 6.6.5.3 RouterException Methods

The RouterException [1306] type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

**Reference**

RouterException Class [1306]

TwinCAT.Ads.TcpRouter Namespace [1254]

### 6.6.5.3.1 RouterException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

Parameters

- **info**
  - Type: `System.Runtime.Serialization.SerializationInfo`
  - The `SerializationInfo` that holds the serialized object data about the exception being thrown.

- **context**
  - Type: `System.Runtime.Serialization.StreamingContext`
  - The `StreamingContext` that contains contextual information about the source or destination.

Implements

- `ISerializable.GetObjectData(SerializationInfo, StreamingContext)`
- `Exception.GetObjectData(SerializationInfo, StreamingContext)`

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>info</td>
</tr>
</tbody>
</table>

Reference

- `RouterException Class [➔ 1306]`
- `TwinCAT.Ads.TcpRouter Namespace [➔ 1254]`

### 6.6.5.4 RouterException Events

The `RouterException [➔ 1306]` type exposes the following members.

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="_serialize.png" alt="_serialize" /> SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

Reference

- `RouterException Class [➔ 1306]`
- `TwinCAT.Ads.TcpRouter Namespace [➔ 1254]`

### 6.6.6 RouterNotInitializedException Class

Class `RouterNotInitializedException`. Implements the `RouterException [➔ 1306]`
Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.Ads.TcpRouter.RouterException [1306]
    TwinCAT.Ads.TcpRouter.RouterNotInitializedException

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class RouterNotInitializedException : RouterException

The RouterNotInitializedException type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from RouterException [1306])</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

#### Reference

- TwinCAT.Ads.TcpRouter Namespace [1254]
- TwinCAT.Ads.TcpRouter.RouterException [1306]

### 6.6.6.1 RouterNotInitializedException Properties

The `RouterNotInitializedException` [1310] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

#### Reference

- RouterNotInitializedException Class [1310]
- TwinCAT.Ads.TcpRouter Namespace [1254]

### 6.6.6.2 RouterNotInitializedException Methods

The `RouterNotInitializedException` [1310] type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from RouterException [1306].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

RouterNotInitializedException Class [1310]

TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.6.3 RouterNotInitializedException Events

The RouterNotInitializedException [1310] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

RouterNotInitializedException Class [1310]

TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.7 RouterNotStartedException Class

Class RouterNotStartedException. Implements the RouterException [1306]

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.Ads.TcpRouter.RouterException [1306]
      TwinCAT.Ads.TcpRouter.RouterNotStartedException
The RouterNotStartedException type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>).</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>).</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from <code>Exception</code>).</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>).</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the <code>Exception</code> that is the root cause of one or more subsequent exceptions. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the <code>SerializationInfo</code> with information about the exception. (Inherited from <code>RouterException</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>
Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.TcpRouter Namespace [1254]

TwinCAT.Ads.TcpRouter.RouterException [1306]

6.6.7.1 RouterNotStartedException Properties

The RouterNotStartedException [1313] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

RouterNotStartedException Class [1313]

TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.7.2 RouterNotStartedException Methods

The RouterNotStartedException [1313] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from RouterException.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

RouterNotStartedException Class [1313]

TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.7.3 RouterNotStartedException Events

The RouterNotStartedException [1313] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

RouterNotStartedException Class [1313]

TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.8 RouterStatus Enumeration

Enum RouterStatus

Namespace: TwinCAT.Ads.TcpRouter [1254]

Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public enum RouterStatus
```
Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>None / Uninitialized</td>
</tr>
<tr>
<td>Initializing</td>
<td>1</td>
<td>Initializing</td>
</tr>
<tr>
<td>Starting</td>
<td>2</td>
<td>Router Starting</td>
</tr>
<tr>
<td>Started</td>
<td>3</td>
<td>Router Started / Executing</td>
</tr>
<tr>
<td>Stopping</td>
<td>4</td>
<td>Router Stopping.</td>
</tr>
<tr>
<td>Stopped</td>
<td>5</td>
<td>Router stopped.</td>
</tr>
</tbody>
</table>

Remarks
Indicates the status of the router.

Reference
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.9 RouterStatusChangedEventArgs Class
Event Arguments sent when the RouterStatusChanged [1281]. Implements the EventArgs.

Inheritance Hierarchy
System.Object
  System.EventArgs
  TwinCAT.Ads.TcpRouter.RouterStatusChangedEventArgs

Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#

public class RouterStatusChangedEventArgs : EventArgs

The RouterStatusChangedEventArgs type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RouterStatusChangedEventArgs</td>
<td>Initializes a new instance of the RouterStatusChangedEventArgs class.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### TwinCAT.Ads NAMESPACES

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

#### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RouterStatus</td>
<td>The router status</td>
</tr>
</tbody>
</table>

#### Reference

- **TwinCAT.Ads.TcpRouter Namespace** [1254]
- **System.EventArgs**
- **IAmsRouter.RouterStatusChangedEventArgs** [1281]

### 6.6.9.1  RouterStatusChangedEventArgs Constructor

Initializes a new instance of the `RouterStatusChangedEventArgs` class.

**Namespace:** TwinCAT.Ads.TcpRouter [1254]
**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public RouterStatusChangedEventArgs(
    RouterStatus status)
```

**Parameters**

- `status`: Type: TwinCAT.Ads.TcpRouter.RouterStatus [1316]
The status.

**Reference**

- **RouterStatusChangedEventArgs Class** [1317]
- **TwinCAT.Ads.TcpRouter Namespace** [1254]

### 6.6.9.2  RouterStatusChangedEventArgs Methods

The `RouterStatusChangedEventArgs` type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Reference**

RouterStatusChangedEventArgs Class [1317]

TwinCAT.Ads.TcpRouter Namespace [1254]

### 6.6.9.3 RouterStatusChangedEventArgs Fields

The `RouterStatusChangedEventArgs` type exposes the following members.

#### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RouterStatus</td>
<td>The router status</td>
</tr>
</tbody>
</table>

**Reference**

RouterStatusChangedEventArgs Class [1317]

TwinCAT.Ads.TcpRouter Namespace [1254]

### 6.6.9.3.1 RouterStatusChangedEventArgs.RouterStatus Field

The router status

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public readonly RouterStatus RouterStatus
```

**Field Value**

Type: `RouterStatus` [1316]

**Reference**

RouterStatusChangedEventArgs Class [1317]

TwinCAT.Ads.TcpRouter Namespace [1254]
6.6.10  **StaticRoutesXmlConfigurationBuilderExtension Class**

Extension class adding StaticRoutes.xml file reading to the IConfigurationBuilder.

**Inheritance Hierarchy**

- System.Object
- TwinCAT.Ads.TcpRouter.StaticRoutesXmlConfigurationBuilderExtension

**Namespace:**  TwinCAT.Ads.TcpRouter

**Assembly:**  TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static class StaticRoutesXmlConfigurationBuilderExtension
```

The StaticRoutesXmlConfigurationBuilderExtension type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="AddStaticRoutesXmlConfiguration" /></td>
<td>Adds the static routes XML configuration.</td>
</tr>
</tbody>
</table>

**Reference**

- TwinCAT.Ads.TcpRouter Namespace [1254]

### 6.6.10.1  **StaticRoutesXmlConfigurationBuilderExtension Methods**

The StaticRoutesXmlConfigurationBuilderExtension [1320] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="AddStaticRoutesXmlConfiguration" /></td>
<td>Adds the static routes XML configuration.</td>
</tr>
</tbody>
</table>

**Reference**

- StaticRoutesXmlConfigurationBuilderExtension Class [1320]
- TwinCAT.Ads.TcpRouter Namespace [1254]

### 6.6.10.1.1  **StaticRoutesXmlConfigurationBuilderExtension.AddStaticRoutesXmlConfiguration Method**

Adds the static routes XML configuration.
**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static IConfigurationBuilder AddStaticRoutesXmlConfiguration(
    this IConfigurationBuilder builder
)
```

**Parameters**

- **builder**  
  Type: `IConfigurationBuilder`  
  The builder.

**Return Value**

Type: `IConfigurationBuilder`  
`IConfigurationBuilder`.

**Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type `IConfigurationBuilder`. When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods (Visual Basic)] or [Extension Methods (C# Programming Guide)].

**Reference**

- [StaticRoutesXmlConfigurationBuilderExtension Class] [1320]
- [TwinCAT.Ads.TcpRouter Namespace] [1254]

### 6.6.11 StaticRoutesXmlConfigurationProvider Class

Class `StaticRoutesXmlConfigurationProvider`. Implements the `ConfigurationProvider`.

**Inheritance Hierarchy**

- `System.Object`
  - `ConfigurationProvider`
  - TwinCAT.Ads.TcpRouter.StaticRoutesXmlConfigurationProvider

**Namespace:** TwinCAT.Ads.TcpRouter [1254]

**Assembly:** TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public class StaticRoutesXmlConfigurationProvider : ConfigurationProvider
```

The `StaticRoutesXmlConfigurationProvider` type exposes the following members.
Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StaticRoutesXmlConfigurationProvider</td>
<td>Initializes a new instance of the StaticRoutesXmlConfigurationProvider class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>The configuration key value pairs for this provider. (Inherited from ConfigurationProvider.)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
| GetChildKeys       | Returns the list of keys that this provider has. (Overrides ConfigurationProvider.GetChildKeys(IEnumerable.String., String).
| GetHashCode        | Serves as the default hash function. (Inherited from Object.) |
| GetReloadToken     | Returns a IChangeToken that can be used to listen when this provider is reloaded. (Inherited from ConfigurationProvider.) |
| GetType            | Gets the Type of the current instance. (Inherited from Object.) |
| Load               | Loads (or reloads) the data for this provider. (Overrides ConfigurationProvider.Load.) |
| MemberwiseClone    | Creates a shallow copy of the current Object. (Inherited from Object.) |
| OnReload           | Triggers the reload change token and creates a new one. (Inherited from ConfigurationProvider.) |
| Set                | Sets a value for a given key. (Inherited from ConfigurationProvider.) |
| ToString           | Generates a string representing this provider name and relevant details. (Inherited from ConfigurationProvider.) |
| TryGet             | Attempts to find a value with the given key, returns true if one is found, false otherwise. (Overides ConfigurationProvider.TryGet(String, String.). |

Reference

TwinCAT.Ads.TcpRouter Namespace [1254]

ConfigurationProvider

6.6.11.1 StaticRoutesXmlConfigurationProvider Constructor

Initializes a new instance of the StaticRoutesXmlConfigurationProvider class.

Namespace: TwinCAT.Ads.TcpRouter [1254]

Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
public static RoutesXmlConfigurationProvider()
```

Reference

`StaticRoutesXmlConfigurationProvider Class [1321]`

`TwinCAT.Ads.TcpRouter Namespace [1254]`

6.6.11.2 StaticRoutesXmlConfigurationProvider Properties

The `StaticRoutesXmlConfigurationProvider [1321]` type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>The configuration key value pairs for this provider. (Inherited from ConfigurationProvider.)</td>
</tr>
</tbody>
</table>

Reference

`StaticRoutesXmlConfigurationProvider Class [1321]`

`TwinCAT.Ads.TcpRouter Namespace [1254]`

6.6.11.3 StaticRoutesXmlConfigurationProvider Methods

The `StaticRoutesXmlConfigurationProvider [1321]` type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetChildKeys</td>
<td>Returns the list of keys that this provider has. (Overrides ConfigurationProvider.GetChildKeys(IEnumerable&lt;string, string&gt;.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetReloadToken</td>
<td>Returns a IChangeToken that can be used to listen when this provider is reloaded. (Inherited from ConfigurationProvider.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Load</td>
<td>Loads (or reloads) the data for this provider. (Overrides ConfigurationProvider.Load.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>OnReload</td>
<td>Triggers the reload change token and creates a new one. (Inherited from ConfigurationProvider.)</td>
</tr>
<tr>
<td>Set</td>
<td>Sets a value for a given key. (Inherited from ConfigurationProvider.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ToString</td>
<td>Generates a string representing this provider name and relevant details. (Inherited from ConfigurationProvider.)</td>
</tr>
<tr>
<td>TryGet</td>
<td>Attempts to find a value with the given key, returns true if one is found, false otherwise. (overrides ConfigurationProvider.TryGet(String, String.).)</td>
</tr>
</tbody>
</table>

Reference

StaticRoutesXmlConfigurationProvider Class [1321]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.11.3.1 StaticRoutesXmlConfigurationProvider.GetChildKeys Method

Returns the list of keys that this provider has.

Namespace: TwinCAT.Ads.TcpRouter [1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public override IEnumerable<string> GetChildKeys(
    IEnumerable<string> earlierKeys,
    string parentPath
)

Parameters

- earlierKeys Type: System.Collections.Generic.IEnumerable<String>. The earlier keys that other providers contain.
- parentPath Type: System.String. The path for the parent IConfiguration.

Return Value

Type: IEnumerable<String>. The list of keys for this provider.

Implements

IConfigurationProvider.GetChildKeys(IEnumerable<String>, String)

Reference

StaticRoutesXmlConfigurationProvider Class [1321]
TwinCAT.Ads.TcpRouter Namespace [1254]

6.6.11.3.2 StaticRoutesXmlConfigurationProvider.Load Method

Loads (or reloads) the data for this provider.
Syntax

C#

```csharp
public override void Load()
```

Implements

`IConfigurationProvider.Load`.

Reference

`StaticRoutesXmlConfigurationProvider Class [1321]`

`TwinCAT.Ads.TcpRouter Namespace [1254]`

### 6.6.11.3.3 StaticRoutesXmlConfigurationProvider.TryGet Method

Attempts to find a value with the given key, returns true if one is found, false otherwise.

Namespace: `TwinCAT.Ads.TcpRouter [1254]`
Assembly: `TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`

Syntax

C#

```csharp
public override bool TryGet(
    string key,
    out string value
)
```

Parameters

- **key**
  - Type: `System.String`
  - The key to lookup.

- **value**
  - Type: `System.String`
  - The value found at key if one is found.

Return Value

- **Type**: `Boolean`
- True if key has a value, false otherwise.

Implements

`IConfigurationProvider.TryGet(String, String.)`

Reference

`StaticRoutesXmlConfigurationProvider Class [1321]`

`TwinCAT.Ads.TcpRouter Namespace [1254]`
6.6.12 StaticRoutesXmlConfigurationSource Class

StaticRoutes Configuration Sources Implements the IConfigurationSource

Inheritance Hierarchy

System.Object
    TwinCAT.Ads.TcpRouter.StaticRoutesXmlConfigurationSource

Namespace: TwinCAT.Ads.TcpRouter
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class StaticRoutesXmlConfigurationSource : IConfigurationSource

The StaticRoutesXmlConfigurationSource type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StaticRoutesXmlConfigurationSource</td>
<td></td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build</td>
<td>Builds the IConfigurationProvider for this source.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.TcpRouter Namespace [1254]
IConfigurationSource

6.6.12.1 StaticRoutesXmlConfigurationSource Constructor

Namespace: TwinCAT.Ads.TcpRouter
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

public StaticRoutesXmlConfigurationSource()

Reference

StaticRoutesXmlConfigurationSource Class [↑ 1326]
TwinCAT.Ads.TcpRouter Namespace [↑ 1254]

6.6.12.2 StaticRoutesXmlConfigurationSource Methods

The StaticRoutesXmlConfigurationSource [↑ 1326] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build [↑ 1327]</td>
<td>Builds the IConfigurationProvider for this source.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

StaticRoutesXmlConfigurationSource Class [↑ 1326]
TwinCAT.Ads.TcpRouter Namespace [↑ 1254]

6.6.12.2.1 StaticRoutesXmlConfigurationSource.Build Method

Builds the IConfigurationProvider for this source.

Namespace: TwinCAT.Ads.TcpRouter [↑ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public IConfigurationProvider Build(
  IConfigurationBuilder builder
)
Parameters

builder Type: IConfigurationBuilder
The IConfigurationBuilder.

Return Value

Type: IConfigurationProvider
An IConfigurationProvider

Implements

IConfigurationSource.Build(IConfigurationBuilder)

Reference

StaticRoutesXmlConfigurationSource Class [► 1326]
TwinCAT.Ads.TcpRouter Namespace [► 1254]

6.7 TwinCAT.Ads.TypeSystem Namespace

Root namespace for the ADS type system.

Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AliasType</td>
<td>Alias DataType</td>
</tr>
<tr>
<td>ArrayType</td>
<td>Represents an Array DataType</td>
</tr>
<tr>
<td>BitMappingType</td>
<td>Helper Data Type to implement Bit mapping types.</td>
</tr>
<tr>
<td>DataType</td>
<td>DataType class</td>
</tr>
<tr>
<td>EnumTypeT</td>
<td>Enum DataType</td>
</tr>
<tr>
<td>Field</td>
<td>Represents a field of an Struct/Alias/Union</td>
</tr>
<tr>
<td>Instance</td>
<td>Instance implementation</td>
</tr>
<tr>
<td>Member</td>
<td>Represents a member of an StructType</td>
</tr>
<tr>
<td>PCCHType</td>
<td>Class PCCHType. This class cannot be inherited. Implements the PointerType</td>
</tr>
<tr>
<td>PointerType</td>
<td>Represents a pointer type.</td>
</tr>
<tr>
<td>PrimitiveType</td>
<td>Class PrimitiveType.</td>
</tr>
<tr>
<td>PVoidType</td>
<td>Class PVoidType. This class cannot be inherited. Implements the PointerType</td>
</tr>
<tr>
<td>ReferenceType</td>
<td>Represents a reference type</td>
</tr>
<tr>
<td>RpcMethod</td>
<td>RPC Method Description</td>
</tr>
</tbody>
</table>
### Class Description

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RpcStructType[1447]</td>
<td>StructType which is callable by RPC Methods.</td>
</tr>
<tr>
<td>StringType[1452]</td>
<td>String DataType</td>
</tr>
<tr>
<td>StructType[1457]</td>
<td>Represents a struct type</td>
</tr>
<tr>
<td>SubRangeType.T.[1463]</td>
<td>Represents a SubRangeType</td>
</tr>
<tr>
<td>Symbol[1469]</td>
<td>Symbol class</td>
</tr>
<tr>
<td>SymbolIterator[1517]</td>
<td>Iterator class for enumerations of Symbols[2176].</td>
</tr>
<tr>
<td>SymbolLoaderFactory[1523]</td>
<td>The class SymbolLoaderFactory[1523] is used to create a new instance of the AdsSymbolLoader initialized to the parametrized mode (SymbolBrowser V2, new Version)</td>
</tr>
<tr>
<td>UnionType[1533]</td>
<td>Represents a union type</td>
</tr>
<tr>
<td>WStringType[1537]</td>
<td>Represents an Unicode string (Wide string)</td>
</tr>
</tbody>
</table>

### Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAdsSymbol[1379]</td>
<td>Interface IAdsSymbol</td>
</tr>
<tr>
<td>IAdsSymbolLoader[1383]</td>
<td>Symbol Loader interface</td>
</tr>
<tr>
<td>IContextMaskProvider[1387]</td>
<td>Interface IContextMaskProvider</td>
</tr>
</tbody>
</table>

### 6.7.1  AliasType Class

**Alias DataType**

**Inheritance Hierarchy**

System.Object

<table>
<thead>
<tr>
<th>System.Object</th>
<th>TwinCAT.Ads.TypeSystem.DataType[1349]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TwinCAT.Ads.TypeSystem.AliasType</td>
</tr>
</tbody>
</table>

**Namespace:** TwinCAT.Ads.TypeSystem[1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public sealed class AliasType : DataType,
    IAliasType, IDataType, IBitSize
```
The AliasType type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the <a href="#">IDataType</a> (Inherited from <a href="#">DataType</a>.)</td>
</tr>
<tr>
<td>BaseType</td>
<td>Gets the Base Type</td>
</tr>
<tr>
<td>BaseTypeName</td>
<td>Gets the BaseType name</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <a href="#">DataType</a> in bits. (Inherited from <a href="#">DataType</a>.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from <a href="#">DataType</a>.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from <a href="#">DataType</a>.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from <a href="#">DataType</a>.)</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the <a href="#">IDataType</a> (Namespace + Name) (Inherited from <a href="#">DataType</a>.)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the DataType (Inherited from <a href="#">DataType</a>.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this <a href="#">IDataType</a> is a bit mapping Type (Inherited from <a href="#">DataType</a>.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <a href="#">DataType</a>.)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this <a href="#">IDataType</a> is a container type (Overrides <a href="#">DataType.IsContainer</a>.)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this <a href="#">DataType</a> is a pointer type (Inherited from <a href="#">DataType</a>.)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this <a href="#">DataType</a> is primitive (Overrides <a href="#">DataType.IsPrimitive</a>.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this <a href="#">DataType</a> is a reference type (Inherited from <a href="#">DataType</a>.)</td>
</tr>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached. (Overrides <a href="#">DataType.ManagedType</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from <a href="#">DataType</a>.)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the <a href="#">IDataType</a> exists. (Inherited from <a href="#">DataType</a>.)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the Size of the <a href="#">DataType</a> in Bytes or bits. (Inherited from <a href="#">DataType</a>.)</td>
</tr>
</tbody>
</table>

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <a href="#.">Object</a></td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <a href="#.">Object</a>)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
### AliasType Properties

The AliasType type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType (Inherited from DataType.)</td>
</tr>
<tr>
<td>BaseType</td>
<td>Gets the Base Type</td>
</tr>
<tr>
<td>BaseTypeName</td>
<td>Gets the BaseType name</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the DataType in bits. (Inherited from DataType.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from DataType.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from DataType.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from DataType.)</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the IDataType (Namespace + Name) (Inherited from DataType.)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the Data Type (Inherited from DataType.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this IDataType is a bit mapping Type (Inherited from DataType.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType.)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this IDataType is a container type (Inherited from DataType.IsContainer.)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this IDataType is a pointer type (Inherited from DataType.)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IDataType is primitive (Inherited from DataType.IsPrimitive.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this IDataType is a reference type (Inherited from DataType.)</td>
</tr>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached. (Overides DataType.ManagedType.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from DataType.)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the IDataType exists. (Inherited from DataType.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Size [1361]</td>
<td>Gets the Size of the <strong>DataType [1349]</strong> in Bytes or bits. (Inherited from <strong>DataType [1349]</strong>.)</td>
</tr>
</tbody>
</table>

**Reference**

AliasType Class [1329]

TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.1.1.1  AliasType.BaseType Property

Gets the Base Type

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public IDataType BaseType { get; }
```

**Property Value**

Type: **IDataType [1986]**

**Implements**

IAliasType.BaseType [1956]

**Reference**

AliasType Class [1329]

TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.1.1.2  AliasType.BaseTypeName Property

Gets the BaseType name

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public string BaseTypeName { get; }
```

**Property Value**

Type: **String**
Implement

IAliasType.BaseTypeName [► 1957]

Reference

AliasType Class [► 1329]

TwinCAT.Ads.TypeSystem Namespace [► 1328]

6.7.1.1.3 AliasType.IsContainer Property

Gets a value indicating whether this IDataType [► 1986] is a container type

Namespace: TwinCAT.Ads.TypeSystem [► 1328]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public override bool IsContainer { get; }
```

Property Value

Type: Boolean

true if this instance is container type; otherwise, false.

Implements

IDataType.IsContainer [► 1990]

Remarks

Container Types are all types that contain SubElements like

- Array [► 1649]
- Pointer [► 1649]
- Union [► 1649]
- Struct [► 1649]
- Function [► 1649]
- FunctionBlock [► 1649]
- Program [► 1649]

And the Alias [► 1649] types, if they have a container type as base type.

Reference

AliasType Class [► 1329]

TwinCAT.Ads.TypeSystem Namespace [► 1328]

IDataType.Category [► 1988]
6.7.1.4 AliasType.IsPrimitive Property

Gets a value indicating whether this IDataType is primitive.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public override bool IsPrimitive { get; }
```

**Property Value**

Type: Boolean
true if this instance is primitive; otherwise, false.

**Implements**

IDataType.IsPrimitive

**Reference**

AliasType Class

TwinCAT.Ads.TypeSystem Namespace

6.7.1.5 AliasType.ManagedType Property

Gets the corresponding .NET Type if attached.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public override Type ManagedType { get; }
```

**Property Value**

Type: Type
Dot net type.

**Reference**

AliasType Class

TwinCAT.Ads.TypeSystem Namespace

6.7.1.2 AliasType Methods

The AliasType type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents this instance. (Overrrides <code>DataType.ToString</code>.</td>
</tr>
</tbody>
</table>

Reference

AliasType Class [1329]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.1.2.1 AliasType.ToString Method

Returns a `String` that represents this instance.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  

```csharp
public override string ToString()
```

Return Value

Type: `String`  
A `String` that represents this instance.

Reference

AliasType Class [1329]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.2 ArrayType Class

Represents an Array `DataType`

DataType class

Inheritance Hierarchy

System.Object  
  TwinCAT.Ads.TypeSystem.DataType [1349]  
  TwinCAT.Ads.TypeSystem.ArrayType

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#  
public class ArrayType : DataType, IArrayType,  
IDataType, IBitSize

The ArrayType type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes[1352]</td>
<td>Gets the attributes of the [1986] (Inherited from [1349].)</td>
</tr>
<tr>
<td>BitSize[1353]</td>
<td>Gets the size of the [1349] in bits. (Inherited from [1349].)</td>
</tr>
<tr>
<td>ByteSize[1353]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from [1349].)</td>
</tr>
<tr>
<td>Category[1354]</td>
<td>Gets the Data Type category (Inherited from [1349].)</td>
</tr>
<tr>
<td>Comment[1354]</td>
<td>Gets the comment. (Inherited from [1349].)</td>
</tr>
<tr>
<td>DimensionCount[1339]</td>
<td>Gets the dimension count.</td>
</tr>
<tr>
<td>Dimensions[1339]</td>
<td>Gets the dimensions as read only collection.</td>
</tr>
<tr>
<td>ElementCount[1339]</td>
<td>Gets the element count.</td>
</tr>
<tr>
<td>ElementSize[1340]</td>
<td>Gets the byte-size of a single element of the array</td>
</tr>
<tr>
<td>ElementType[1340]</td>
<td>Gets the type of the contained elements.</td>
</tr>
<tr>
<td>ElementTypeName[1341]</td>
<td>Gets the name of the element type.</td>
</tr>
<tr>
<td>FullName[1355]</td>
<td>Gets the full name of the [1986] (Namespace + Name) (Inherited from [1349].)</td>
</tr>
<tr>
<td>Id[1355]</td>
<td>Gets the ID of the DataType (Inherited from [1349].)</td>
</tr>
<tr>
<td>IsBitType[1356]</td>
<td>Gets a value indicating whether this [1986] is a bit mapping Type (Inherited from [1349].)</td>
</tr>
<tr>
<td>IsByteAligned[1356]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from [1349].)</td>
</tr>
<tr>
<td>IsContainer[1357]</td>
<td>Gets a value indicating whether this [1986] is a container type (Inherited from [1349].)</td>
</tr>
<tr>
<td>IsJagged[1341]</td>
<td>Gets a value indicating whether this instance is jagged.</td>
</tr>
<tr>
<td>IsOversampled[1342]</td>
<td>Gets a value indicating whether this array instance describes an oversampling type.</td>
</tr>
<tr>
<td>IsPointer[1357]</td>
<td>Gets a value indicating whether this [1986] is a pointer type (Inherited from [1349].)</td>
</tr>
<tr>
<td>IsPrimitive[1342]</td>
<td>Gets a value indicating whether this [1986] is primitive (Overrides [1349].)</td>
</tr>
<tr>
<td>IsReference[1359]</td>
<td>Gets a value indicating whether this [1986] is a reference type (Inherited from [1349].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JaggedLevel [1343]</td>
<td>Gets the jagged level (Non-Jagged Array have level 1)</td>
</tr>
<tr>
<td>ManagedType [1343]</td>
<td>Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType [1359] )</td>
</tr>
<tr>
<td>Name [1360]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from DataType [1349] )</td>
</tr>
<tr>
<td>Namespace [1360]</td>
<td>Gets the namespace string within the IDataType [1986] exists. (Inherited from DataType [1349] )</td>
</tr>
<tr>
<td>Size [1361]</td>
<td>Gets the Size of the DataType [1349] in Bytes or bits. (Inherited from DataType [1349] )</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object )</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object )</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object )</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object )</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object )</td>
</tr>
<tr>
<td>SetDimensions [1344]</td>
<td>Sets the dimensions.</td>
</tr>
<tr>
<td>SetElementType [1345]</td>
<td>Sets the type of the element.</td>
</tr>
<tr>
<td>ToString [1363]</td>
<td>Returns a String that represents this instance. (Inherited from DataType [1349] )</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.2.1 ArrayType Properties

The ArrayType [1335] type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1352]</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from DataType [1349] )</td>
</tr>
<tr>
<td>BitSize [1353]</td>
<td>Gets the size of the DataType [1349] in bits. (Inherited from DataType [1349] )</td>
</tr>
<tr>
<td>ByteSize [1353]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [1349] )</td>
</tr>
<tr>
<td>Category [1354]</td>
<td>Gets the Data Type category (Inherited from DataType [1349] )</td>
</tr>
<tr>
<td>Comment [1354]</td>
<td>Gets the comment. (Inherited from DataType [1349] )</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DimensionCount</td>
<td>Gets the dimension count.</td>
</tr>
<tr>
<td>Dimensions</td>
<td>Gets the dimensions as read only collection.</td>
</tr>
<tr>
<td>ElementCount</td>
<td>Gets the element count.</td>
</tr>
<tr>
<td>ElementSize</td>
<td>Gets the byte-size of a single element of the array</td>
</tr>
<tr>
<td>ElementType</td>
<td>Gets the type of the contained elements.</td>
</tr>
<tr>
<td>ElementTypeName</td>
<td>Gets the name of the element type.</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the $\text{IDataType}$ (Namespace + Name) (Inherited from $\text{DataType}$.)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the $\text{DataType}$ (Inherited from $\text{DataType}$.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this $\text{IDataType}$ is a bit mapping $\text{Type}$ (Inherited from $\text{DataType}$.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from $\text{DataType}$.)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this $\text{IDataType}$ is a container type (Inherited from $\text{DataType}$.)</td>
</tr>
<tr>
<td>IsJagged</td>
<td>Gets a value indicating whether this instance is jagged.</td>
</tr>
<tr>
<td>IsOversampled</td>
<td>Gets a value indicating whether this array instance describes an oversampling type.</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this $\text{IDataType}$ is a pointer type (Inherited from $\text{DataType}$.)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this $\text{IDataType}$ is primitive (Overrides $\text{DataType}.IsPrimitive$.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this $\text{IDataType}$ is a reference type (Inherited from $\text{DataType}$.)</td>
</tr>
<tr>
<td>JaggedLevel</td>
<td>Gets the jagged level (Non-Jagged Array have level 1)</td>
</tr>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached. (Overrides $\text{DataType.ManagedType}$.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the $\text{DataType}$ (without namespace) (Inherited from $\text{DataType}$.)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the $\text{IDataType}$ exists. (Inherited from $\text{DataType}$.)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the Size of the $\text{DataType}$ in Bytes or bits. (Inherited from $\text{DataType}$.)</td>
</tr>
</tbody>
</table>

**Reference**

- ArrayType Class [1335]
- TwinCAT.Ads.TypeSystem Namespace [1328]
6.7.2.1.1 ArrayType.DimensionCount Property

Gets the dimension count.

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public int DimensionCount { get; }

Property Value

Type: Int32
The dimension count.

Reference

ArrayType Class
TwinCAT.Ads.TypeSystem Namespace

6.7.2.1.2 ArrayType.Dimensions Property

Gets the dimensions as read only collection.

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public IDimensionCollection Dimensions { get; }

Property Value

Type: IDimensionCollection
The dimensions.

Implements

IArrrayType.Dimensions

Reference

ArrayType Class
TwinCAT.Ads.TypeSystem Namespace

6.7.2.1.3 ArrayType.ElementCount Property

Gets the element count.
Namespace: TwinCAT.Ads.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int ElementCount { get; }
```

Property Value

Type: Int32
The element count.

Reference

ArrayType Class
TwinCAT.Ads.TypeSystem Namespace

6.7.2.1.4 ArrayType.ElementSize Property

Gets the byte-size of a single element of the array

Namespace: TwinCAT.Ads.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int ElementSize { get; }
```

Property Value

Type: Int32
The size of the element.

Reference

ArrayType Class
TwinCAT.Ads.TypeSystem Namespace

6.7.2.1.5 ArrayType.ElementType Property

Gets the type of the contained elements.

Namespace: TwinCAT.Ads.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public IDataType ElementType { get; }
```
Property Value

Type: IDataType
The type of the element.

Implements

IArrayType.ElementType

Reference

ArrayType Class
TwinCAT.Ads.TypeSystem Namespace

6.7.2.1.6 ArrayType.ElementTypeName Property

Gets the name of the element type.

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public string ElementTypeName { get; }

Property Value

Type: String
The name of the element type.

Implements

IArrayType.ElementTypeName

Reference

ArrayType Class
TwinCAT.Ads.TypeSystem Namespace

6.7.2.1.7 ArrayType.IsJagged Property

Gets a value indicating whether this instance is jagged.

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool IsJagged { get; }
TwinCAT.Ads Namespaces

Property Value
Type: Boolean
true if this instance is jagged; otherwise, false.

Implements
IArrayType.IsJagged [1975]

Reference
ArrayType Class [1335]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.2.1.8   ArrayType.IsOversampled Property

Gets a value indicating whether this array instance describes an oversampling type.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
public bool IsOversampled { get; }

Property Value
Type: Boolean
true if this instance is oversampling; otherwise, false.

Reference
ArrayType Class [1335]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.2.1.9   ArrayType.IsPrimitive Property

Gets a value indicating whether this [IDataType [1986]] is primitive

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
public override bool IsPrimitive { get; }

Property Value
Type: Boolean
true if this instance is primitive; otherwise, false.
6.7.2.1.10 ArrayType.JaggedLevel Property

Gets the jagged level (Non-Jagged Array have level 1)

**Namespace:** TwinCAT.Ads.TypeSystem

**Reference**

ArrayType Class

TwinCAT.Ads.TypeSystem Namespace

**Syntax**

**C#**

```csharp
public int JaggedLevel { get; }
```

**Property Value**

Type: Int32

The jagged level.

**Implements**

IArrayType.JaggedLevel

**Reference**

ArrayType Class

TwinCAT.Ads.TypeSystem Namespace

6.7.2.1.11 ArrayType.ManagedType Property

Gets the corresponding .NET Type if attached.

**Namespace:** TwinCAT.Ads.TypeSystem

**Reference**

ArrayType Class

TwinCAT.Ads.TypeSystem Namespace

**Syntax**

**C#**

```csharp
public override Type ManagedType { get; }
```
Property Value

Type: Type
Dot net type.

Reference

ArrayType Class [1335]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.2.2 ArrayType Methods

The ArrayType [1335] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetDimensions [1344]</td>
<td>Sets the dimensions.</td>
</tr>
<tr>
<td>SetElementType [1345]</td>
<td>Sets the type of the element.</td>
</tr>
<tr>
<td>ToString [1363]</td>
<td>Returns a String that represents this instance. (Inherited from DataType</td>
</tr>
<tr>
<td></td>
<td>[1349].)</td>
</tr>
</tbody>
</table>

Reference

ArrayType Class [1335]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.2.2.1 ArrayType.SetDimensions Method

Sets the dimensions.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected void SetDimensions(
  DimensionCollection dims
)

Reference

ArrayType Class [1335]
TwinCAT.Ads.TypeSystem Namespace [1328]
Parameters
dims Type: TwinCAT.TypeSystem.DimensionCollection [1671]
The dims.

Reference
ArrayType Class [1335]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.2.2 ArrayType.SetElementType Method
Sets the type of the element.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected void SetElementType(
    DataType elementType
)

Parameters
elementType Type: TwinCAT.Ads.TypeSystem.DataType [1349]
Type of the element.

Reference
ArrayType Class [1335]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.3 BitMappingType Class
Helper Data Type to implement Bit mapping types.

Inheritance Hierarchy
System.Object
    TwinCAT.Ads.TypeSystem.DataType [1349]
    TwinCAT.Ads.TypeSystem.BitMappingType
Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public sealed class BitMappingType : DataType

The BitMappingType type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitMappingType</td>
<td>Initializes a new instance of the BitMappingType class.</td>
</tr>
</tbody>
</table>

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the <code>IDataType</code> (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>DataType</code> in bits. (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the <code>IDataType</code> (Namespace + Name) (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the <code>DataType</code> (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a bit mapping Type (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a container type (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a pointer type (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this <code>IDataType</code> is primitive (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a reference type (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the <code>IDataType</code> exists. (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the Size of the <code>DataType</code> in Bytes or bits. (Inherited from <code>DataType</code>.)</td>
</tr>
</tbody>
</table>

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DataType)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.Ads.TypeSystem Namespace [1328]

#### 6.7.3.1 BitMappingType Constructor

Initializes a new instance of the BitMappingType [1345] class.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public BitMappingType(
    string name,
    int bitSize,
    Type dotnetType
)
```

**Parameters**

- **name**  
  Type: System.String  
  The name.
- **bitSize**  
  Type: System.Int32  
  The size of the type in bits.
- **dotnetType**  
  Type: System.Type  
  Type of the dotnet.

**Reference**

BitMappingType Class [1345]  
TwinCAT.Ads.TypeSystem Namespace [1328]

#### 6.7.3.2 BitMappingType Properties

The BitMappingType [1345] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the DataType [1349] in bits. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from DataType [1349].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comment [1354]</td>
<td>Gets the comment. (Inherited from <code>DataType [1349]</code>.)</td>
</tr>
<tr>
<td>FullName [1355]</td>
<td>Gets the full name of the <code>IDataType [1986]</code> (Namespace + Name) (Inherited from <code>DataType [1349]</code>.)</td>
</tr>
<tr>
<td>Id [1355]</td>
<td>Gets the ID of the <code>DataType</code> (Inherited from <code>DataType [1349]</code>.)</td>
</tr>
<tr>
<td>IsBitType [1356]</td>
<td>Gets a value indicating whether this <code>IDataType [1986]</code> is a bit mapping Type (Inherited from <code>DataType [1349]</code>.)</td>
</tr>
<tr>
<td>IsByteAligned [1356]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <code>DataType [1349]</code>.)</td>
</tr>
<tr>
<td>IsContainer [1357]</td>
<td>Gets a value indicating whether this <code>IDataType [1986]</code> is a container type (Inherited from <code>DataType [1349]</code>.)</td>
</tr>
<tr>
<td>IsPointer [1357]</td>
<td>Gets a value indicating whether this <code>IDataType [1986]</code> is a pointer type (Inherited from <code>DataType [1349]</code>.)</td>
</tr>
<tr>
<td>IsPrimitive [1358]</td>
<td>Gets a value indicating whether this <code>IDataType [1986]</code> is primitive (Inherited from <code>DataType [1349]</code>.)</td>
</tr>
<tr>
<td>IsReference [1359]</td>
<td>Gets a value indicating whether this <code>IDataType [1986]</code> is a reference type (Inherited from <code>DataType [1349]</code>.)</td>
</tr>
<tr>
<td>ManagedType [1359]</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from <code>DataType [1349]</code>.)</td>
</tr>
<tr>
<td>Name [1360]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from <code>DataType [1349]</code>.)</td>
</tr>
<tr>
<td>Namespace [1360]</td>
<td>Gets the namespace string within the <code>IDataType [1986]</code> exists. (Inherited from <code>DataType [1349]</code>.)</td>
</tr>
<tr>
<td>Size [1361]</td>
<td>Gets the Size of the <code>DataType [1349]</code> in Bytes or bits. (Inherited from <code>DataType [1349]</code>.)</td>
</tr>
</tbody>
</table>

### Reference

**BitMappingType Class [1345]**

**TwinCAT.Ads.TypeSystem Namespace [1328]**

#### 6.7.3.3 BitMappingType Methods

The `BitMappingType [1345]` type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents this instance. (Inherited from <code>DataType [1349]</code>.)</td>
</tr>
</tbody>
</table>

### Reference

**BitMappingType Class [1345]**
6.7.4  **DataType Class**

DataType class

**Inheritance Hierarchy**

System.Object  
TwinCAT.Ads.TypeSystem.DataType  

More... [1350]

**Namespace:** TwinCAT.Ads.TypeSystem [1328]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public class DataType : IDataType, IBitSize
```

The `DataType` type exposes the following members.

**Constructors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DataType</code></td>
<td>Initializes a new instance of the <code>DataType</code> class (copy Constructor)</td>
</tr>
</tbody>
</table>

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Attributes</code></td>
<td>Gets the attributes of the <code>IDataType</code></td>
</tr>
<tr>
<td><code>BitSize</code></td>
<td>Gets the size of the <code>DataType</code> in bits.</td>
</tr>
<tr>
<td><code>ByteSize</code></td>
<td>Gets the (aligned) size of of the <code>Type/Instance</code> in Bytes</td>
</tr>
<tr>
<td><code>Category</code></td>
<td>Gets the <code>DataType</code> category</td>
</tr>
<tr>
<td><code>Comment</code></td>
<td>Gets the comment.</td>
</tr>
<tr>
<td><code>FullName</code></td>
<td>Gets the full name of the <code>IDataType</code></td>
</tr>
<tr>
<td><code>Id</code></td>
<td>Gets the ID of the <code>DataType</code></td>
</tr>
<tr>
<td><code>IsBitType</code></td>
<td>Gets a value indicating whether this <code>IDataType</code> is a bit mapping Type</td>
</tr>
<tr>
<td><code>IsByteAligned</code></td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td><code>IsContainer</code></td>
<td>Gets a value indicating whether this <code>IDataType</code> is a container type</td>
</tr>
<tr>
<td><code>IsPointer</code></td>
<td>Gets a value indicating whether this <code>IDataType</code> is a pointer type</td>
</tr>
<tr>
<td><code>IsPrimitive</code></td>
<td>Gets a value indicating whether this <code>IDataType</code> is primitive</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this <a href="#">DataType</a> is a reference type.</td>
</tr>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace).</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the <a href="#">DataType</a> exists.</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the Size of the Data Type in Bytes or bits.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>IsPointerType</td>
<td>Determines whether the specified category is a pointer type.</td>
</tr>
<tr>
<td>IsReferenceType</td>
<td>Determines whether the specified category is a reference type.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance.</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads.TypeSystem Namespace [1328]

### Inheritance Hierarchy

System.Object

- TwinCAT.Ads.TypeSystem.DataType
  - TwinCAT.Ads.TypeSystem.AliasType [1329]
  - TwinCAT.Ads.TypeSystem.ArrayType [1335]
  - TwinCAT.Ads.TypeSystem.BitMappingType [1345]
  - TwinCAT.Ads.TypeSystem.EnumType.T. [1363]
  - TwinCAT.Ads.TypeSystem.PointerType [1414]
  - TwinCAT.Ads.TypeSystem.PrimitiveType [1418]
  - TwinCAT.Ads.TypeSystem ReferenceType [1425]
  - TwinCAT.Ads.TypeSystem.StringType [1452]
  - TwinCAT.Ads.TypeSystem.StructType [1457]
  - TwinCAT.Ads.TypeSystem.SubRangeType.T. [1463]
  - TwinCAT.Ads.TypeSystem.UnionType [1533]
  - TwinCAT.Ads.TypeSystem.WStringType [1537]
6.7.4.1  **DataType Constructor**

Initializes a new instance of the `DataType` class (copy Constructor)

**Namespace:**  TwinCAT.Ads.TypeSystem

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd2ca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
protected DataType(
    DataType copy
)
```

**Parameters**

`copy`  
Type: `TwinCAT.Ads.TypeSystem.DataType`  
The copy.

**Reference**

DataType Class [1349]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.4.2  **DataType Properties**

The `DataType` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Attributes [1352] | Gets the attributes of the `IDataType`[
| BitSize [1353]  | Gets the size of the `DataType` in bits.                                   |
| ByteSize [1353] | Gets the (aligned) size of of the Type/Instance in Bytes                   |
| Category [1354] | Gets the Data Type category                                                 |
| Comment [1354]  | Gets the comment.                                                           |
| FullName [1355] | Gets the full name of the `IDataType` (Namespace + Name)                    |
| Id [1355]       | Gets the ID of the Data Type                                                |
| IsBitType [1356] | Gets a value indicating whether this `IDataType` is a bit mapping Type       |
| IsByteAligned [1356] | Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) |
| IsContainer [1357] | Gets a value indicating whether this `IDataType` is a container type        |
| IsPointer [1357] | Gets a value indicating whether this `IDataType` is a pointer type          |
| IsPrimitive [1358] | Gets a value indicating whether this `IDataType` is primitive               |
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this [IDataType] is a reference type</td>
</tr>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the [IDataType] exists.</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the Size of the [DataType] in Bytes or bits.</td>
</tr>
</tbody>
</table>

#### Reference

- **DataType Class**
- **TwinCAT.Ads.TypeSystem Namespace**

#### 6.7.4.2.1 Data Type.Attributes Property

Gets the attributes of the [IDataType]

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public ITypeAttributeCollection Attributes { get; }
```

**Property Value**

Type: [ITypeAttributeCollection]

The attributes.

**Implements**

[IDataType.Attributes]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

- **DataType Class**
- **TwinCAT.Ads.TypeSystem Namespace**
6.7.4.2.2  DataType.BitSize Property

Gets the size of the `DataType` in bits.

**Namespace:**  TwinCAT.Ads.TypeSystem
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```c#
public int BitSize { get; }
```

**Property Value**

Type: `Int32`
The size of the bit.

**Implements**

`IBitSize.BitSize`

**Reference**

`DataType Class`
`TwinCAT.Ads.TypeSystem Namespace`

6.7.4.2.3  DataType.ByteSize Property

Gets the (aligned) size of the Type/Instance in Bytes

**Namespace:**  TwinCAT.Ads.TypeSystem
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```c#
public int ByteSize { get; }
```

**Property Value**

Type: `Int32`
The size of the byte.

**Implements**

`IBitSize.ByteSize`

**Reference**

`DataType Class`
`TwinCAT.Ads.TypeSystem Namespace`
6.7.4.2.4  **DataType.Category Property**

Gets the Data Type category

**Namespace:**  TwinCAT.Ads.TypeSystem

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public DataTypeCategory Category { get; protected set; }
```

**Property Value**

Type:  **DataTypeCategory**

The category.

**Implements**

IDataType.Category

**Reference**

DataType Class

TwinCAT.Ads.TypeSystem Namespace

6.7.4.2.5  **DataType.Comment Property**

Gets the comment.

**Namespace:**  TwinCAT.Ads.TypeSystem

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public string Comment { get; }
```

**Property Value**

Type:  **String**

The comment.

**Implements**

IDataType.Comment

**Reference**

DataType Class

TwinCAT.Ads.TypeSystem Namespace
6.7.4.2.6  **DataType.FullName Property**

Gets the full name of the [IDataType](#1986) (Namespace + Name)

**Namespace:** TwinCAT.Ads.TypeSystem [1328]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```
public string FullName { get; }
```

**Property Value**

Type: **String**
The full name.

**Implements**

[IDataType.FullName](#1986)

**Reference**

DataType Class [1349]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.4.2.7  **DataType.Id Property**

Gets the ID of the DataType

**Namespace:** TwinCAT.Ads.TypeSystem [1328]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```
public int Id { get; }
```

**Property Value**

Type: **Int32**
The id.

**Implements**

[IDataType.Id](#1989)

**Reference**

DataType Class [1349]

TwinCAT.Ads.TypeSystem Namespace [1328]
**6.7.4.2.8 **DataType.IsBitType Property

Gets a value indicating whether this IDataType is a bit mapping Type

**Namespace**: TwinCAT.Ads.TypeSystem

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool IsType { get; }
```

**Property Value**

Type: Boolean
true if this instance is bit mapping subtype; otherwise, false.

**Implements**

IBitSize.IsBitType

**Reference**

DataType Class
TwinCAT.Ads.TypeSystem Namespace

---

**6.7.4.2.9 **DataType.IsByteAligned Property

Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)

**Namespace**: TwinCAT.Ads.TypeSystem

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool IsByteAligned { get; }
```

**Property Value**

Type: Boolean
true if this instance is byte aligned; otherwise, false.

**Implements**

IBitSize.IsByteAligned

**Reference**

DataType Class
TwinCAT.Ads.TypeSystem Namespace
6.7.4.2.10  **DataType.IsContainer Property**

Gets a value indicating whether this `IDataType` [p. 1986] is a container type

**Namespace:**  TwinCAT.Ads.TypeSystem [p. 1328]  
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public virtual bool IsContainer { get; }
```

**Property Value**

Type:  `Boolean`
true if this instance is container type; otherwise, false.

**Implements**

`IDataType.IsContainer` [p. 1990]

**Remarks**

Container Types are all types that contain SubElements like

- `Array` [p. 1649]
- `Pointer` [p. 1649]
- `Union` [p. 1649]
- `Struct` [p. 1649]
- `Function` [p. 1649]
- `FunctionBlock` [p. 1649]
- `Program` [p. 1649]

and the `Alias` [p. 1649] and `Reference` [p. 1649] types, if they have a container type as base type.

**Reference**

`DataType Class` [p. 1349]  
TwinCAT.Ads.TypeSystem Namespace [p. 1328]  
`IDataType.Category` [p. 1988]

---

6.7.4.2.11  **DataType.IsPointer Property**

Gets a value indicating whether this `IDataType` [p. 1986] is a pointer type

**Namespace:**  TwinCAT.Ads.TypeSystem [p. 1328]  
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public virtual bool IsPointer { get; }
```

Property Value

Type: Boolean
true if this instance is pointer type; otherwise, false.

Implements

IDataType.IsPointer

Remarks

Pointer types can be dereferenced with the '^' operator.

Reference

DataType Class
TwinCAT.Ads.TypeSystem Namespace
IDataType.Category

6.7.4.2.12   DataType.IsPrimitive Property

Gets a value indicating whether this IDataType is primitive

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public virtual bool IsPrimitive { get; }
```

Property Value

Type: Boolean
true if this instance is primitive; otherwise, false.

Implements

IDataType.IsPrimitive

Reference

DataType Class
TwinCAT.Ads.TypeSystem Namespace
6.7.4.2.13  **DataType.IsReference Property**

Gets a value indicating whether this [IDataType](#1986) is a reference type

**Namespace:**  TwinCAT.Ads.TypeSystem [1328]

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version:

5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public virtual bool IsReference { get; }
```

**Property Value**

Type:  Boolean

true if this instance is container type; otherwise, false.

**Implements**

IDataType.IsReference [1991]

**Remarks**

Reference types can be dereferenced.

**Reference**

DataType Class [1349]

TwinCAT.Ads.TypeSystem Namespace [1328]

IDataType.Category [1988]

---

6.7.4.2.14  **DataType.ManagedType Property**

Gets the corresponding .NET Type if attached.

**Namespace:**  TwinCAT.Ads.TypeSystem [1328]

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version:

5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public virtual Type ManagedType { get; protected set; }
```

**Property Value**

Type:  Type

Dot net type.

**Reference**

DataType Class [1349]

TwinCAT.Ads.TypeSystem Namespace [1328]
6.7.4.2.15  **DataType.Name Property**

Gets the name of the Data Type (without namespace)

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public string Name { get; protected set; }
```

**Property Value**

Type: **String**

The name.

**Implements**

[IDataType.Name] [1992]

**Reference**

DataType Class [1349]  
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.4.2.16  **DataType.Namespace Property**

Gets the namespace string within the [IDataType [1986] exists.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public string Namespace { get; }
```

**Property Value**

Type: **String**

The namespace.

**Implements**

[IDataType.Namespace] [1992]

**Reference**

DataType Class [1349]  
TwinCAT.Ads.TypeSystem Namespace [1328]
6.7.4.2.17 **DataType.Size Property**

Gets the Size of the DataType in Bytes or bits.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int Size { get; protected set; }
```

**Property Value**

Type: Int32

The size.

**Implements**

IBitSize.Size

**Reference**

DataType Class

TwinCAT.Ads.TypeSystem Namespace

### 6.7.4.3 **DataType Methods**

The **DataType** type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IsPointerType</td>
<td>Determines whether the specified category is a pointer type.</td>
</tr>
<tr>
<td></td>
<td>[1362]</td>
</tr>
<tr>
<td>IsReferenceType</td>
<td>Determines whether the specified category is a reference type.</td>
</tr>
<tr>
<td></td>
<td>[1362]</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Overrides Object.ToString.)</td>
</tr>
</tbody>
</table>
6.7.4.3.1  DataType.IsPointerType Method

Determines whether the specified category is a pointer type.

Namespace:  TwinCAT.Ads.TypeSystem
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static bool IsPointerType(
    DataTypeCategory cat
)

Parameters

cat  Type: TwinCAT.TypeSystem.DataTypeCategory

Return Value

Type:  Boolean
true if [is pointer type] [the specified cat]; otherwise, false.

6.7.4.3.2  DataType.IsReferenceType Method

Determines whether the specified category is a reference type.

Namespace:  TwinCAT.Ads.TypeSystem
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static bool IsReferenceType(
    DataTypeCategory cat
)

Parameters

cat  Type: TwinCAT.TypeSystem.DataTypeCategory

The data type category.
Return Value

Type: Boolean
true if [is reference type] [the specified cat]; otherwise, false.

Reference

DataType Class [1349]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.4.3.3  DataType.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public override string ToString()

Return Value

Type: String
A String that represents this instance.

Reference

DataType Class [1349]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.5  EnumType.T. Class

Enum DataType [1349].

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [1349]
  TwinCAT.Ads.TypeSystem.EnumType.T.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public sealed class EnumType<T> : DataType,
  IEnumType<T>, IAliasType, IDataType, IBitSize, IEnumType
where T : IConvertible
Type Parameters

T

The EnumType.T. type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1352]</td>
<td>Gets the attributes of the IDatatype [1986] (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>BaseType [1366]</td>
<td>Gets the Base Type</td>
</tr>
<tr>
<td>BaseType Name [1367]</td>
<td>Gets the BaseType name</td>
</tr>
<tr>
<td>BitSize [1353]</td>
<td>Gets the size of the DataType [1349] in bits. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ByteSize [1353]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Category [1354]</td>
<td>Gets the Data Type category (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Comment [1354]</td>
<td>Gets the comment. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>EnumValues [1367]</td>
<td>Enumeration specification (if enum)</td>
</tr>
<tr>
<td>FullName [1355]</td>
<td>Gets the full name of the IDatatype [1986] (Namespace + Name) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Id [1355]</td>
<td>Gets the ID of the Data Type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsBitType [1356]</td>
<td>Gets a value indicating whether this IDatatype [1986] is a bit mapping Type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsByteAligned [1356]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsContainer [1357]</td>
<td>Gets a value indicating whether this IDatatype [1986] is a container type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsPointer [1357]</td>
<td>Gets a value indicating whether this IDatatype [1986] is a pointer type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsPrimitive [1358]</td>
<td>Gets a value indicating whether this IDatatype [1986] is primitive (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsReference [1359]</td>
<td>Gets a value indicating whether this IDatatype [1986] is a reference type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ManagedType [1359]</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Name [1360]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Namespace [1360]</td>
<td>Gets the namespace string within the IDatatype [1986] exists. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Size [1361]</td>
<td>Gets the Size of the DataType [1349] in Bytes or bits. (Inherited from DataType [1349].)</td>
</tr>
</tbody>
</table>
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains[1368]</td>
<td>Determines whether the enum values contains the specified name</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetNames[1369]</td>
<td>Gets the filed names of the IEnumType.T. [2021]</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetValues[1370]</td>
<td>Gets the values of the IEnumType.T. [2021]</td>
</tr>
<tr>
<td>Parse[1370]</td>
<td>Parses a name of the IEnumType.T. [2021] and returns the value (as base type)</td>
</tr>
<tr>
<td>ToString[1363]</td>
<td>Returns a String that represents this instance. (Inherited from DataType [1349])</td>
</tr>
<tr>
<td>ToString(IConvertible)[1371]</td>
<td>Returns a String that represents this instance.</td>
</tr>
<tr>
<td>ToString(T)[1372]</td>
<td>Returns a String that represents this instance.</td>
</tr>
<tr>
<td>TryParse(String, IEnumValue)[1373]</td>
<td>Tries to parse the Enum Value</td>
</tr>
<tr>
<td>TryParse(String, T)[1374]</td>
<td>Tries to parse the Enum Value</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.5.1 EnumType.T. Properties

The EnumType.T. [1363] generic type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes[1352]</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>BaseType[1366]</td>
<td>Gets the Base Type</td>
</tr>
<tr>
<td>BaseTypeName[1367]</td>
<td>Gets the BaseType name</td>
</tr>
<tr>
<td>BitSize[1353]</td>
<td>Gets the size of the DataType [1349] in bits. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ByteSize[1353]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Category[1354]</td>
<td>Gets the Data Type category (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Comment[1354]</td>
<td>Gets the comment. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EnumValues [1367]</td>
<td>Enumeration specification (if enum)</td>
</tr>
<tr>
<td>FullName [1355]</td>
<td>Gets the full name of the [IDataType [1986] (Namespace + Name) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Id [1355]</td>
<td>Gets the ID of the DataType (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsBitType [1356]</td>
<td>Gets a value indicating whether this [IDataType [1986] is a bit mapping Type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsByteAligned [1356]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsContainer [1357]</td>
<td>Gets a value indicating whether this [IDataType [1986] is a container type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsPointer [1357]</td>
<td>Gets a value indicating whether this [IDataType [1986] is a pointer type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsPrimitive [1358]</td>
<td>Gets a value indicating whether this [IDataType [1986] is primitive (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsReference [1359]</td>
<td>Gets a value indicating whether this [IDataType [1986] is a reference type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ManagedType [1359]</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Name [1360]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Namespace [1360]</td>
<td>Gets the namespace string within the [IDataType [1986] exists. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Size [1361]</td>
<td>Gets the Size of the [IDataType [1349] in Bytes or bits. (Inherited from DataType [1349].)</td>
</tr>
</tbody>
</table>

Reference

EnumType.T. Class [1363]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.5.1.1 EnumType.T..BaseType Property

Gets the Base Type

**Namespace**: TwinCAT.Ads.TypeSystem [1328]

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public IDataType BaseType { get; }
```

**Property Value**

Type: [IDataType [1986]

The type of the base.
Implements
IAliasType.BaseType

Reference
EnumType.T.Class
TwinCAT.Ads.TypeSystem Namespace

6.7.5.1.2 EnumType.T..BaseTypeName Property

Gets the BaseType name

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public string BaseTypeName { get; }

Property Value

Type: String
The name of the base type.

Implements
IAliasType.BaseTypeName

Reference
EnumType.T.Class
TwinCAT.Ads.TypeSystem Namespace

6.7.5.1.3 EnumType.T..EnumValues Property

Enumeration specification (if enum)

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public IEnumValueCollection EnumValues { get; }

Property Value

Type: IEnumValueCollection
The enum specification.
**TwinCAT.Ads Namespaces**

**Implements**

IEnumType.T..EnumValues [2024]

**Reference**

EnumType.T. Class [1363]

TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.5.2 EnumType.T. Methods

The EnumType.T. [1363] generic type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains [1368]</td>
<td>Determines whether the enum values contains the specified name</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetNames [1369]</td>
<td>Gets the filed names of the IEnumType.T. [2021]</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetValues [1370]</td>
<td>Gets the values of the IEnumType.T. [2021]</td>
</tr>
<tr>
<td>Parse [1370]</td>
<td>Parses a name of the IEnumType.T. [2021] and returns the value (as base type)</td>
</tr>
<tr>
<td>ToString [1363]</td>
<td>Returns a String that represents this instance. (Inherited from DataType [1349])</td>
</tr>
<tr>
<td>ToString(IConvertible) [1371]</td>
<td>Returns a String that represents this instance.</td>
</tr>
<tr>
<td>ToString(T) [1372]</td>
<td>Returns a String that represents this instance.</td>
</tr>
<tr>
<td>TryParse(String, IEnumValue) [1373]</td>
<td>Tries to parse the Enum Value</td>
</tr>
<tr>
<td>TryParse(String, T) [1374]</td>
<td>Tries to parse the Enum Value</td>
</tr>
</tbody>
</table>

**Reference**

EnumType.T. Class [1363]

TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.5.2.1 EnumType.T..Contains Method

Determines whether the enum values contains the specified name

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdcac3e72bc0ea15da1c14
Syntax

C#

```csharp
public bool Contains(
    string name
)
```

Parameters

- **name**
  - Type: `System.String`
  - The name.

Return Value

- Type: `Boolean`
  - true if contains the value, otherwise, false.

Implements

- `IEnumerable<T>.Contains(String)`
- `IEnumerable.Contains(String)`

Reference

- `EnumType.T. Class` [1363]
- `TwinCAT.Ads.TypeSystem Namespace` [1328]

---

**6.7.5.2.2 EnumType.T.Contains Method**

Gets the filed names of the `IEnumerable<T>`.

Namespace: `TwinCAT.Ads.TypeSystem` [1328]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public string[] GetNames()
```

Return Value

- Type: `String[]`
  - System.String[].

Implements

- `IEnumerable<T>.GetNames` [2025]
- `IEnumerable.GetNames` [2018]

Reference

- `EnumType.T. Class` [1363]
- `TwinCAT.Ads.TypeSystem Namespace` [1328]
6.7.5.2.3 EnumType.T..GetValues Method

Gets the values of the EnumType.T. 

Namespace: TwinCAT.Ads.TypeSystem 
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72b0ea15da1c14

Syntax

C#
public T[] GetValues();

Return Value

Type: T[].
T[].

Implements

IEnumType.T..GetValues.

Reference

EnumType.T. Class
TwinCAT.Ads.TypeSystem Namespace

6.7.5.2.4 EnumType.T..Parse Method

Parses a name of the EnumType.T. 
and returns the value (as base type)

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72b0ea15da1c14

Syntax

C#
public T Parse(
    string strValue
    );

Parameters

strValue Type: System.String
Enum Value as string.

Return Value

Type: T[].
T[.]

Implements

IEnumType.T..Parse(String)

2021
1328
1363
2026
1363
1328
2026
2026
6.7.5.2.5 EnumType.T..ToString Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents this instance. (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>ToString(IConvertible)</td>
<td>Returns a <code>String</code> that represents this instance.</td>
</tr>
<tr>
<td>ToString(T)</td>
<td>Returns a <code>String</code> that represents this instance.</td>
</tr>
</tbody>
</table>

**Enumeration**

EnumType.T..ToString Method (IConvertible)

Returns a `String` that represents this instance.

**NameSpace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public string ToString(IConvertible val)
```

**Parameters**

- `val` Type: System.IConvertible
  The value.

**Return Value**

Type: `String`  
A `String` that represents this instance.

**Implements**

- EnumType.ToString(IConvertible)
**EnumType.T..ToString Method (T)**

Returns a **String** that represents this instance.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public string ToString(T val)
```

**Parameters**

- **val**
  - **Type:** **T**
  - The value.

**Return Value**

- **Type:** **String**
  - A **String** that represents this instance.

**Implements**

- IEnumType.T..ToString(T)  

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>val</td>
</tr>
</tbody>
</table>

**Reference**

- EnumType.T..Class
- ToString Overload
- TwinCAT.Ads.TypeSystem Namespace
EnumType.T..TryParse Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryParse(String, IEnumValue.)</td>
<td>Tries to parse the Enum Value</td>
</tr>
<tr>
<td>TryParse(String, T.)</td>
<td>Tries to parse the Enum Value</td>
</tr>
</tbody>
</table>

Reference

EnumType.T. Class [1363]

TwinCAT.Ads.TypeSystem Namespace [1328]

EnumType.T..TryParse Method (String, IEnumValue.)

Tries to parse the Enum Value

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool TryParse(
    string strValue,
    out IEnumValue value
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strValue</td>
<td>System.String</td>
<td>Enum value (in string representation).</td>
</tr>
<tr>
<td>value</td>
<td>TwinCAT.TypeSystem.IEnumValue [2028]</td>
<td>The value.</td>
</tr>
</tbody>
</table>

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IEnumType.TryParse(String, IEnumValue.) [2021]

Reference

EnumType.T. Class [1363]
TryParse Overload [1373]
TwinCAT.Ads.TypeSystem Namespace [1328]
EnumType.T..TryParse Method (String, T.)

Tries to parse the Enum Value

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```
public bool TryParse(
    string strValue,
    out T value
)
```

**Parameters**

- **strValue**
  - Type: System.String
  - Enum value (in string representation).

- **value**
  - Type: T [1363]
  - The value.

**Return Value**

Type: Boolean
true if XXXX, false otherwise.

**Implements**

IEnumeratorType.T..TryParse(String, T.) [2027]

**Reference**

EnumType.T..Class [1363]

TryParse Overload [1373]

TwinCAT.Ads.TypeSystem Namespace [1382]

### 6.7.6 Field Class

Represents a field of an Struct/Alias/Union

**Inheritance Hierarchy**

- System.Object
  - TwinCAT.Ads.TypeSystem.Instance [1388]
  - TwinCAT.Ads.TypeSystem.Field
    - TwinCAT.Ads.TypeSystem.Member [1406]

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public class Field : Instance, IField,
IAttributedInstance, IInstance, IBitSize
```

The Field type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from Instance)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of this Instance in bits. (Inherited from Instance)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from Instance)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the DataTypeCategory of the Instance. (Inherited from Instance)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from Instance)</td>
</tr>
<tr>
<td>ContextMask</td>
<td>Gets the context mask of this instance. (Inherited from Instance)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType of the Instance. (Inherited from Instance)</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this instance has a value. (Inherited from Instance)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from Instance)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from Instance)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from Instance)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from Instance)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Indicates that this instance is persistent. (Inherited from Instance)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this instance is reference. (Inherited from Instance)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from Instance)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this instance is reference. (Inherited from Instance)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this Instance is static. (Inherited from Instance)</td>
</tr>
<tr>
<td>IsTcComInterfacePointer</td>
<td>Indicates that this instance is a TcComInterfacePointer. (Inherited from Instance)</td>
</tr>
<tr>
<td>IsTypeGuid</td>
<td>Indicates that this instance has set TypeGuid flag. (Inherited from Instance)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace name. (Inherited from Instance)</td>
</tr>
</tbody>
</table>
### Field Properties

The `Field` type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from <code>Instance</code>)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of this <code>Instance</code> in bits. (Inherited from <code>Instance</code>)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from <code>Instance</code>)</td>
</tr>
</tbody>
</table>

---

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>OnGetSize</td>
<td>Handler function getting the size of the <code>Instance</code> (Inherited from <code>Instance</code>)</td>
</tr>
<tr>
<td>OnSetInstanceName</td>
<td>Sets a new InstanceName InstancePath (Inherited from <code>Instance</code>)</td>
</tr>
<tr>
<td>SetAttributes</td>
<td>Sets the type attributes (Inherited from <code>Instance</code>)</td>
</tr>
<tr>
<td>SetContextMask</td>
<td>Sets the context mask. (Inherited from <code>Instance</code>)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents this instance. (Inherited from <code>Instance</code>)</td>
</tr>
</tbody>
</table>

---

### Reference

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.6.1 Field Properties

The `Field` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Gets the the DataTypeCategory [1649] of the Instance. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>ContextMask</td>
<td>Gets the context mask of this instance. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType [1986] of the Instance [2052]. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this instance has a value. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the size of the object is Byte aligned (BitSize % 8 == 0) (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Indicates that this instance is persistent. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this instance is reference. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this instance is reference. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this Instance [2052] is static. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsTcComInterfacePointer</td>
<td>Indicates that this instance is a TcComInterfacePointer. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsTypeGuid</td>
<td>Indicates that this instance has set TypeGuid flag. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace name. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>ParentType</td>
<td>Gets the Parent of this IField [2040].</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the IDataType [1986] in bytes or Bits dependant on IsBitType [1396] (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052]. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding of this Field [1374]</td>
</tr>
</tbody>
</table>

**Reference**

Field Class [1374]

TwinCAT.Ads.TypeSystem Namespace [1328]
### 6.7.6.1.1 Field.ParentType Property

Gets the Parent of this `Field`.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd0c3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public IDataType ParentType { get; }
```

**Property Value**

Type: `IDataType`

The type of the parent (Alias, Union, Struct)

**Implements**

`IField.ParentType`

**Reference**

Field Class

TwinCAT.Ads.TypeSystem Namespace

### 6.7.6.1.2 Field.ValueEncoding Property

Gets the value encoding of this `Field`.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd0c3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public virtual Encoding ValueEncoding { get; }
```

**Property Value**

Type: `Encoding`

The value encoding.

**Implements**

`IAtributedInstance.ValueEncoding`

**Reference**

Field Class

TwinCAT.Ads.TypeSystem Namespace
6.7.6.2 Field Methods

The Field [1374] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>OnGetSize</td>
<td>Handler function getting the size of the Instance (Inherited from Instance.)</td>
</tr>
<tr>
<td>OnSetInstanceName</td>
<td>Sets a new InstanceName InstancePath (Inherited from Instance.)</td>
</tr>
<tr>
<td>SetAttributes</td>
<td>Sets the type attributes (Inherited from Instance.)</td>
</tr>
<tr>
<td>SetContextMask</td>
<td>Sets the context mask. (Inherited from Instance.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from Instance.)</td>
</tr>
</tbody>
</table>

Reference

Field Class [1374]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.7 IAdsSymbol Interface

Interface IAdsSymbol

Namespace: TwinCAT.Ads.TypeSystem [1328]

Assembly: TwinCAT.Ads.Abistractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IAdsSymbol : ISymbol,
    IAttributedInstance, IInstance, IBitSize, IProcessImageAddress, IContextMaskProvider
```

The IAdsSymbol type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the Instance [2052] (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>ContextMask [1388]</td>
<td>Gets the ContextMask of the symbol, indicating the task the variable belongs to. If ContextMask is not zero use CyclicInContext [639] or OnChangeInContext [639] to add notifications. (Inherited from IContextMaskProvider [1387].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>DataTypeld [1382]</td>
<td>DataType identifier of the Symbol AdsDataTypeld [574]</td>
</tr>
<tr>
<td>ImageBaseAddress [1383]</td>
<td>Gets the AmsAddress [648] of the Process Image</td>
</tr>
<tr>
<td>IndexGroup [2093]</td>
<td>Gets the index group of the Symbol (Inherited from IProcessImageAddress [2092].)</td>
</tr>
<tr>
<td>IndexOffset [2093]</td>
<td>Gets the index offset of the Symbol (Inherited from IProcessImageAddress [2092].)</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods .) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this IInstance [2052] is static. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsVirtual [2094]</td>
<td>Gets a value indicating whether this instance is virtual. (Inherited from IProcessImageAddress [2092].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Parent [▶ 2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)</td>
</tr>
<tr>
<td>Size [▶ 1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsDataType [▶ 1986] (Inherited from IBitSize [▶ 1982].)</td>
</tr>
<tr>
<td>SubSymbols [▶ 2182]</td>
<td>Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)</td>
</tr>
<tr>
<td>TypeName [▶ 2056]</td>
<td>Gets the name of the DataType [▶ 1986] that is used for this Instance [▶ 2052]. (Inherited from IInstance [▶ 2052].)</td>
</tr>
<tr>
<td>ValueEncoding [▶ 1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

TwinCAT.TypeSystem.ISymbol [▶ 2176]

TwinCAT.TypeSystem.IProcessImageAddress [▶ 2092]

### 6.7.7.1 IAdsSymbol Properties

The IAdsSymbol [▶ 1379] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [▶ 1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)</td>
</tr>
<tr>
<td>BitSize [▶ 1984]</td>
<td>Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)</td>
</tr>
<tr>
<td>ByteSize [▶ 1984]</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)</td>
</tr>
<tr>
<td>Category [▶ 2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)</td>
</tr>
<tr>
<td>Comment [▶ 2053]</td>
<td>Gets the comment of the Instance [▶ 2052] (Inherited from Instance [▶ 2052].)</td>
</tr>
<tr>
<td>ContextMask [▶ 1388]</td>
<td>Gets the ContextMask of the symbol, indicating the task the variable belongs to. If ContextMask is not zero use CyclicInContext [▶ 639] or OnChangeInContext [▶ 639] to add notifications. (Inherited from IContextMaskProvider [▶ 1387].)</td>
</tr>
<tr>
<td>DataType [▶ 2054]</td>
<td>Gets the IDataType [▶ 1986] of the Instance [▶ 2052]. (Inherited from Instance [▶ 2052].)</td>
</tr>
<tr>
<td>DatTypeld [▶ 1382]</td>
<td>DataTyple identifier of the Symbol AdsDataTyple [▶ 574]</td>
</tr>
<tr>
<td>ImageBaseAddress [▶ 1383]</td>
<td>Gets the AmsAddress [▶ 648] of the Process Image</td>
</tr>
<tr>
<td>IndexGroup [▶ 2093]</td>
<td>Gets the index group of the Symbol (Inherited from IProcessImageAddress [▶ 2092].)</td>
</tr>
<tr>
<td>IndexOffset [▶ 2093]</td>
<td>Gets the index offset of the Symbol (Inherited from IProcessImageAddress [▶ 2092].)</td>
</tr>
<tr>
<td>InstanceName [▶ 2054]</td>
<td>Gets the name of the instance (without periods .) (Inherited from Instance [▶ 2052].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Instance)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full</td>
</tr>
<tr>
<td></td>
<td>DataType but instead of some sort of bit mapping (Inherited from IBitSize</td>
</tr>
<tr>
<td></td>
<td>(1982).)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from IBitSize)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from ISymbol)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this ISymbol is persistent. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>ISymbol)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the Instance represents a Pointer type (Pointer TO) (Inherited</td>
</tr>
<tr>
<td></td>
<td>from ISymbol)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from ISymbol)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>ISymbol)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the Instance represents a Reference type (REFERENCE TO)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Instance)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this Instance is static. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Instance)</td>
</tr>
<tr>
<td>IsVirtual</td>
<td>Gets a value indicating whether this instance is virtual. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>IProcessImageAddress)</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol (Inherited from ISymbol)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType</td>
</tr>
<tr>
<td></td>
<td>(Inherited from IBitSize)</td>
</tr>
<tr>
<td>SubSymbols</td>
<td>Gets the SubSymbols of the ISymbol (Inherited from ISymbol)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType that is used for this Instance (Inherited</td>
</tr>
<tr>
<td></td>
<td>from Instance)</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance)</td>
</tr>
</tbody>
</table>

Reference

IAdsSymbol Interface [1379]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.7.1.1 IAdsSymbol.DataTypeId Property

DataType identifier of the Symbol AdsDataTypeId [574]

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
AdsDataTypeId DataTypeId { get; }
```

Property Value

Type: `AdsDataTypeId` [574]
Data type of the symbol.

Reference

IAdsSymbol Interface [1379]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.7.1.2 IAdsSymbol.ImageBaseAddress Property

Gets the `AmsAddress` [648] of the Process Image

**Namespace**: TwinCAT.Ads.TypeSystem [1328]
**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
AmsAddress ImageBaseAddress { get; }
```

Property Value

Type: `AmsAddress` [648]
The address.

Reference

IAdsSymbol Interface [1379]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.8 IAdsSymbolLoader Interface

Symbol Loader interface

**Namespace**: TwinCAT.Ads.TypeSystem [1328]
**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IAdsSymbolLoader : ISymbolLoader, ISymbolProvider, ISymbolServer
```

The IAdsSymbolLoader type exposes the following members.
## TwinCAT.Ads Namespaces

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BuildInTypes [2202]</td>
<td>Gets the build in types. (Inherited from ISymbolLoader [2200].)</td>
</tr>
<tr>
<td>DataTypes [2206]</td>
<td>Gets the data types (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>DefaultNotificationSettings [1385]</td>
<td>Gets/Sets the default notification settings for this SymbolLoader</td>
</tr>
<tr>
<td>DefaultValueEncoding [2206]</td>
<td>Gets the default value encoding. (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>ImageBaseAddress [1386]</td>
<td>Gets the image base address.</td>
</tr>
<tr>
<td>PlatformPointerSize [1386]</td>
<td>Gets the (byte) size of Pointers on the attached platform system.</td>
</tr>
<tr>
<td>RootNamespaceName [2204]</td>
<td>Gets the name of the root namespace (Inherited from ISymbolProvider [2203].)</td>
</tr>
<tr>
<td>Settings [2202]</td>
<td>Gets or sets the access Method (Inherited from ISymbolLoader [2200].)</td>
</tr>
<tr>
<td>Symbols [2207]</td>
<td>Gets the symbols. (Inherited from ISymbolServer [2205].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDataTypesAsync [2208]</td>
<td>Gets the data types asynchronously. (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>GetSymbolsAsync [2208]</td>
<td>Gets the symbols asynchronously (Inherited from ISymbolServer [2205].)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads.TypeSystem Namespace [1328]

#### 6.7.8.1 IAdsSymbolLoader Properties

The IAdsSymbolLoader [1383] type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BuildInTypes [2202]</td>
<td>Gets the build in types. (Inherited from ISymbolLoader [2200].)</td>
</tr>
<tr>
<td>DataTypes [2206]</td>
<td>Gets the data types (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>DefaultNotificationSettings [1385]</td>
<td>Gets/Sets the default notification settings for this SymbolLoader</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DefaultValueEncoding</td>
<td>Gets the default value encoding. (Inherited from ISymbolServer.)</td>
</tr>
<tr>
<td>ImageBaseAddress</td>
<td>Gets the image base address.</td>
</tr>
<tr>
<td>PlatformPointerSize</td>
<td>Gets the (byte) size of Pointers on the attached platform system.</td>
</tr>
<tr>
<td>RootNamespaceName</td>
<td>Gets the name of the root namespace (Inherited from ISymbolProvider.)</td>
</tr>
<tr>
<td>Settings</td>
<td>Gets or sets the access Method (Inherited from ISymbolLoader.)</td>
</tr>
<tr>
<td>Symbols</td>
<td>Gets the symbols. (Inherited from ISymbolServer.)</td>
</tr>
</tbody>
</table>

**Reference**

IAdsSymbolLoader Interface [1383]
TwinCAT.Ads.TypeSystem Namespace [1328]

**6.7.8.1.1 IAdsSymbolLoader.DefaultNotificationSettings Property**

Gets/Sets the default notification settings for this SymbolLoader

**Namespace:** TwinCAT.Ads.TypeSystem [1328]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
INotificationSettings DefaultNotificationSettings { get; set; }
```

**Property Value**

Type: INotificationSettings [972]
The default notification settings.

**Remarks**

The Default notification Settings can be set on the SymbolLoader and is used as default on the different Symbols. On the symbol itself the NotificationSettings [979] can be overridden.

**Examples**

Setting the DefaultNotificationSettings on the IAdsSymbolLoader [1383] object:

**Set DefaultNotificationSettings**

```csharp
// Create AdsClient object
using (AdsClient client = new AdsClient())
{
    // No automatic Synchronization (necessary for Console applications without message loop)
    // client.Synchronize = false;
    // Connect to client
    client.Connect(address);
```
// Usage of 'dynamic' type/symbol loader
SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolLoadMode.DynamicTree, ValueAccessMode.IndexGroupOffsetPreferred);
IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);

// Set the DefaultNotification Properties
dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransMode.ClientOnChange, 200, 200);

// Determine the symbols
dynamic dynamicSymbols = ((IDynamicSymbolLoader)dynLoader).SymbolsDynamic;

// Task 1 Symbol (build in symbol)
dynamic task1Symbol = dynamicSymbols._TaskInfo[1];

// CycleCount Symbol
dynamic cycleCountSymbol = task1Symbol.CycleCount;

// Override Notification Setting for Cycle Count Symbol
cycleCountSymbol.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 250, 0);

// Register Dynamic Value Changed event.
cycleCountSymbol.ValueChanged += new EventHandler<ValueChangedEventArgs>(cycleCount_ValueChanged);

Reference

IAdsSymbolLoader Interface [1383]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.8.1.2 IAdsSymbolLoader.ImageBaseAddress Property

Gets the image base address.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

AmsAddress ImageBaseAddress { get; }

Property Value

Type: AmsAddress [648]
The image base address.

Reference

IAdsSymbolLoader Interface [1383]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.8.1.3 IAdsSymbolLoader.PlatformPointerSize Property

Gets the (byte) size of Pointers on the attached platform system.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
int PlatformPointerSize { get; }
```

Property Value

Type: Int32
The size of the platform pointer.

Reference

IAdsSymbolLoader Interface [1383]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.8.2 IAdsSymbolLoader Methods

The IAdsSymbolLoader [1383] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDataTypesAsync</td>
<td>Gets the data types asynchronously. (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>GetSymbolsAsync</td>
<td>Gets the symbols asynchronously (Inherited from ISymbolServer [2205].)</td>
</tr>
</tbody>
</table>

Reference

IAdsSymbolLoader Interface [1383]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.9 IContextMaskProvider Interface

Interface IContextMaskProvider

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IContextMaskProvider
```

The IContextMaskProvider type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContextMask</td>
<td>Gets the ContextMask of the symbol, indicating the task the variable belongs to. If ContextMask is not zero use CyclicInContext [639] or OnChangelnContext [639] to add notifications.</td>
</tr>
</tbody>
</table>
6.7.9.1  **IContextMaskProvider Properties**

The **IContextMaskProvider** type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContextMask</td>
<td>Gets the ContextMask of the symbol, indicating the task the variable belongs to. If ContextMask is not zero use <strong>CyclicInContext</strong> or <strong>OnChangeInContext</strong> to add notifications.</td>
</tr>
</tbody>
</table>

### Reference

**IContextMaskProvider Interface**

**TwinCAT.Ads.TypeSystem Namespace**

### 6.7.9.1.1 **IContextMaskProvider.ContextMask Property**

Gets the ContextMask of the symbol, indicating the task the variable belongs to. If ContextMask is not zero use **CyclicInContext** or **OnChangeInContext** to add notifications.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
byte ContextMask { get; }
```

### Property Value

Type: **Byte**

### Reference

**IContextMaskProvider Interface**

**TwinCAT.Ads.TypeSystem Namespace**

### 6.7.10 **Instance Class**

Instance implementation

### Inheritance Hierarchy

**System.Object**

- TwinCAT.Ads.TypeSystem.Instance
  - TwinCAT.Ads.TypeSystem.Field
  - TwinCAT.Ads.TypeSystem.Symbol
**Namespace:** TwinCAT.Ads.TypeSystem [1328]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public class Instance : IInstance, IBitSize
```

The `Instance` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1391]</td>
<td>Gets the Type Attributes.</td>
</tr>
<tr>
<td>BitSize [1392]</td>
<td>Gets the size of this Instance in bits.</td>
</tr>
<tr>
<td>ByteSize [1392]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes</td>
</tr>
<tr>
<td>Category [1393]</td>
<td>Gets the the <code>DataTypeCategory [1649]</code> of the Instance.</td>
</tr>
<tr>
<td>Comment [1393]</td>
<td>Gets the comment.</td>
</tr>
<tr>
<td>ContextMask [1394]</td>
<td>Gets the context mask of this instance.</td>
</tr>
<tr>
<td>DataType [1394]</td>
<td>Gets the <code>IDataType [1986]</code> of the <code>IInstance [2052]</code>.</td>
</tr>
<tr>
<td>HasValue [1395]</td>
<td>Gets a value indicating whether this instance has a value.</td>
</tr>
<tr>
<td>InstanceName [1395]</td>
<td>Gets the name of the instance (without periods (.))</td>
</tr>
<tr>
<td>InstancePath [1396]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.))</td>
</tr>
<tr>
<td>IsBitType [1396]</td>
<td>Gets a value indicating whether this instance is not basing on a full</td>
</tr>
<tr>
<td></td>
<td><code>DataType but instead of some sort of bit mapping</code>.</td>
</tr>
<tr>
<td>IsByteAligned [1397]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td>IsPersistent [1397]</td>
<td>Indicates that this instance is persistent.</td>
</tr>
<tr>
<td>IsPointer [1398]</td>
<td>Gets a value indicating whether this instance is reference.</td>
</tr>
<tr>
<td>IsReadOnly [1398]</td>
<td>Indicates that this instance is read only.</td>
</tr>
<tr>
<td>IsReference [1399]</td>
<td>Gets a value indicating whether this instance is reference.</td>
</tr>
<tr>
<td>IsStatic [1399]</td>
<td>Gets a value indicating whether this <code>IInstance [2052]</code> is static.</td>
</tr>
<tr>
<td>IsTcComInterfacePointer [1400]</td>
<td>Indicates that this instance is a TcComInterfacePointer.</td>
</tr>
<tr>
<td>IsTypeGuid [1400]</td>
<td>Indicates that this instance has set TypeGuid flag.</td>
</tr>
<tr>
<td>Namespace [1401]</td>
<td>Gets the namespace name.</td>
</tr>
<tr>
<td>Size [1401]</td>
<td>Gets the size of the <code>IDataType [1986]</code> in bytes or Bits dependant on</td>
</tr>
<tr>
<td></td>
<td><code>IsBitType [1396]</code></td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType that is used for this Instance.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlignTypeName</td>
<td>Aligns the type name</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>OnGetSize</td>
<td>Handler function getting the size of the Instance</td>
</tr>
<tr>
<td>OnSetInstanceName</td>
<td>Sets a new InstanceName InstancePath</td>
</tr>
<tr>
<td>SetAttributes</td>
<td>Sets the type attributes</td>
</tr>
<tr>
<td>SetContextMask</td>
<td>Sets the context mask.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Overrides Object.ToString.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads.TypeSystem Namespace

### 6.7.10.1 Instance Properties

The Instance type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes.</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of this Instance in bits.</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the the DataTypeCategory of the Instance.</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment.</td>
</tr>
<tr>
<td>ContextMask</td>
<td>Gets the context mask of this instance.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>[1394]</strong></td>
<td>Gets the <code>IDataType</code> of the <code>IInstance</code>.</td>
</tr>
<tr>
<td><strong>[1395]</strong></td>
<td>Gets a value indicating whether this instance has a value.</td>
</tr>
<tr>
<td><strong>[1396]</strong></td>
<td>Gets the name of the instance (without periods (.)).</td>
</tr>
<tr>
<td><strong>[1396]</strong></td>
<td>Gets the relative / absolute access path to the instance (with periods (.)).</td>
</tr>
<tr>
<td><strong>[1396]</strong></td>
<td>Gets a value indicating whether this instance is not basing on a full</td>
</tr>
<tr>
<td></td>
<td><code>DataType</code> but instead of some sort of bit mapping.</td>
</tr>
<tr>
<td><strong>[1396]</strong></td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0).</td>
</tr>
<tr>
<td><strong>[1397]</strong></td>
<td>Indicates that this instance is persistent.</td>
</tr>
<tr>
<td><strong>[1398]</strong></td>
<td>Gets a value indicating whether this instance is reference.</td>
</tr>
<tr>
<td><strong>[1398]</strong></td>
<td>Indicates that this instance is read only.</td>
</tr>
<tr>
<td><strong>[1399]</strong></td>
<td>Gets a value indicating whether this instance is reference.</td>
</tr>
<tr>
<td><strong>[1399]</strong></td>
<td>Gets a value indicating whether this <code>IInstance</code> is static.</td>
</tr>
<tr>
<td><strong>[1400]</strong></td>
<td>Indicates that this instance is a <code>TcComInterfacePointer</code>.</td>
</tr>
<tr>
<td><strong>[1400]</strong></td>
<td>Indicates that this instance has set TypeGuid flag.</td>
</tr>
<tr>
<td><strong>[1401]</strong></td>
<td>Gets the namespace name.</td>
</tr>
<tr>
<td><strong>[1401]</strong></td>
<td>Gets the size of the <code>IDataType</code> in bytes or Bits dependant on <code>IsBitType</code>.</td>
</tr>
<tr>
<td><strong>[1402]</strong></td>
<td>Gets the name of the <code>DataType</code> that is used for this <code>IInstance</code>.</td>
</tr>
</tbody>
</table>

**Reference**

**Instance Class** [1388]

**TwinCAT.Ads.TypeSystem Namespace** [1328]

### 6.7.10.1.1 Instance.Attributes Property

Gets the Type Attributes.

**Namespace**: TwinCAT.Ads.TypeSystem [1328]

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: S.0.294+Branch.releases-S.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public ITypeAttributeCollection Attributes { get; }
```
6.7.10.1.2 Instance.BitSize Property

Gets the size of this Instance in bits.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public virtual int BitSize { get; }
```

**Property Value**

Type: Int32
The size of the bit.

**Implements**

IBitSize.BitSize

**Reference**

Instance Class
TwinCAT.Ads.TypeSystem Namespace

6.7.10.1.3 Instance.ByteSize Property

Gets the (aligned) size of of the Type/Instance in Bytes.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public int ByteSize { get; }
```

**Property Value**

Type: Int32
The size of the byte.
**Implements**

IBitSize.ByteSize [1984]

**Reference**

Instance Class [1388]

TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.10.1.4 Instance.Category Property

Gets the the **DataTypeCategory** [1649] of the Instance.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public DataTypeCategory Category { get; protected set; }
```

**Property Value**

**Type:** DataTypeCategory [1649]

The category.

**Remarks**

Corresponds to the **Category** [1988]

**Reference**

Instance Class [1388]

TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.10.1.5 Instance.Comment Property

Gets the comment.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public string Comment { get; }
```

**Property Value**

**Type:** String

The comment.
Implements

IInstance.Comment [2053]

Reference

Instance Class [1388]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.1.6 Instance.ContextMask Property

Gets the context mask of this instance.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public byte ContextMask { get; }

Property Value

Type: Byte

Remarks

The Size of the internal data is 4-Bit

Reference

Instance Class [1388]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.1.7 Instance.DataType Property

Gets the IDataType [1986] of the IInstance [2052].

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public IDataType DataType { get; protected set; }

Property Value

Type: IDataType [1986]
The type of the data.
6.7.10.1.8 Instance.HasValue Property

Gets a value indicating whether this instance has a value.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# 

```csharp
public virtual bool HasValue { get; }
```

**Property Value**

Type: Boolean

true if this instance has value; otherwise, false.

**Remarks**

Reference

Instance Class [1388]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.1.9 Instance.InstanceName Property

Gets the name of the instance (without periods (.)

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# 

```csharp
public string InstanceName { get; protected set; }
```

**Property Value**

Type: String

The name of the instance.
6.7.10.1.10 Instance.InstancePath Property

Gets the relative / absolute access path to the instance (with periods (.)

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public virtual string InstancePath { get; }
```

**Property Value**

Type: **String**
The instance path.

**Implements**

IInstance.InstancePath [2055]

**Remarks**

If this path is relative or absolute depends on the context. IMember [2065] are using relative paths, ISymbol [2176]s are using absolute ones.

**Reference**

Instance Class [1388]

TwinCAT.Ads.TypeSystem Namespace [1328]
Syntax

C#

```csharp
public bool IsBitType { get; }
```

Property Value

Type: Boolean
true if this instance is bit mapping; otherwise, false.

Implements

IBitSize.IsBitType [1984]

Reference

Instance Class [1388]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.1.12 Instance.IsByteAligned Property

Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool IsByteAligned { get; }
```

Property Value

Type: Boolean
true if this instance is byte aligned; otherwise, false.

Implements

IBitSize.IsByteAligned [1985]

Reference

Instance Class [1388]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.1.13 Instance.IsPersistent Property

Indicates that this instance is persistent.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
## 6.7.10.1.14 Instance.IsPointer Property

Gets a value indicating whether this instance is reference.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public bool IsPointer { get; }
```

### Property Value

Type: Boolean

true if this instance is reference; otherwise, false.

**Implements**

IInstance.IsPointer

### Reference

- **Instance Class** [1388]
- **TwinCAT.Ads.TypeSystem Namespace** [1328]

## 6.7.10.1.15 Instance.IsReadOnly Property

Indicates that this instance is read only.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public bool IsReadOnly { get; }
```

### Property Value

Type: Boolean

true if this instance is read only; otherwise, false.

**Reference**

- **Instance Class** [1388]
- **TwinCAT.Ads.TypeSystem Namespace** [1328]
Property Value

Type: Boolean

Remarks

Actually, this Flag is restricted to TcCOM-Objects readonly Parameters. Within the PLC this is used for the ApplicationName and ProjectName of PLC instances. Write-Access on these Modules will create an DeviceAccessDenied [575] error.

Reference

Instance Class [1388]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.1.16  Instance.IsReference Property

Gets a value indicating whether this instance is reference.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool IsReference { get; }

Property Value

Type: Boolean
true if this instance is reference; otherwise, false.

Implements

IInstance.IsReference [2056]

Reference

Instance Class [1388]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.1.17  Instance.IsStatic Property

Gets a value indicating whether this IInstance [2052] is static.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool IsStatic { get; protected set; }
Property Value

Type: Boolean
true if this instance is static; otherwise, false.

Implements

IInstance.IsStatic [2056]

Reference

Instance Class [1388]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.1.18 Instance.IsTcComInterfacePointer Property

Indicates that this instance is a TcComInterfacePointer.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public bool IsTcComInterfacePointer { get; }

Property Value

Type: Boolean

Reference

Instance Class [1388]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.1.19 Instance.IsTypeGuid Property

Indicates that this instance has set TypeGuid flag.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public bool IsTypeGuid { get; }

Property Value

Type: Boolean
6.7.10.1.20 Instance.Namespace Property

Gets the namespace name.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# public string Namespace { get; } 

**Property Value**

Type: String

The namespace.

Reference

Instance Class [1388]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.1.21 Instance.Size Property

Gets the size of the `IDataType` in bytes or Bits dependant on `IsBitType`.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# public int Size { get; protected set; } 

**Property Value**

Type: Int32

The size of the bit.

**Implements**

IBitSize.Size [1985]

Reference

Instance Class [1388]

TwinCAT.Ads.TypeSystem Namespace [1328]
6.7.10.1.22 Instance.TypeName Property

Gets the name of the DataType [1986] that is used for this IInstance [2052].

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public string TypeName { get; protected set; }

Property Value

Type: String
The name of the type.

Implements

IInstance.TypeName [2056]

Reference

Instance Class [1388]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.2 Instance Methods

The Instance [1388] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlignTypeName</td>
<td>Aligns the type name</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>OnGetSize</td>
<td>Handler function getting the size of the Instance.</td>
</tr>
<tr>
<td>OnSetInstanceName</td>
<td>Sets a new InstanceName InstancePath</td>
</tr>
<tr>
<td>SetAttributes</td>
<td>Sets the type attributes</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SetContextMask</td>
<td>[1405] Sets the context mask.</td>
</tr>
<tr>
<td>ToString</td>
<td>[1405] Returns a String that represents this instance. (Overrrides Object.ToString.)</td>
</tr>
</tbody>
</table>

Reference

Instance Class [1388]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.2.1 Instance.AlignTypeName Method

Aligns the type name

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
protected static string AlignTypeName(
    string typeName
)
```

**Parameters**

- **typeName**
  Type: System.String
  Name of the type.

**Return Value**

Type: String
System.String.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>Type name not valid!</td>
</tr>
</tbody>
</table>

Reference

Instance Class [1388]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.2.2 Instance.OnGetSize Method

Handler function getting the size of the Instance [1388]

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#
protected virtual int OnGetSize()

Return Value
Type: Int32
System.Int32.

Reference
Instance Class [1388]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.2.3 Instance.OnSetInstanceName Method
Sets a new InstanceName InstancePath

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected virtual void OnSetInstanceName(
    string instanceName
)

Parameters
instanceName Type: System.String
Instance name.

Reference
Instance Class [1388]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.10.2.4 Instance.SetAttributes Method
Sets the type attributes

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected void SetAttributes(
    TypeAttributeCollection coll
)
Parameters

coll  
Type: TwinCAT.TypeSystem.TypeAttributeCollection [2418]
The attributes.

Reference

Instance Class [1388]
TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.10.2.5 Instance.SetContextMask Method

Sets the context mask.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
protected void SetContextMask(
    byte contextMask
)
```

**Parameters**

- **contextMask**  
  Type: System.Byte  
  The context mask.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td>contextMask</td>
</tr>
</tbody>
</table>

Reference

Instance Class [1388]
TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.10.2.6 Instance.ToString Method

Returns a `String` that represents this instance.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public override string ToString()
```
Return Value

Type: String
A String that represents this instance.

Reference

Instance Class [1388]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.11 Member Class

Represents a member of an StructType [1457]

Inheritance Hierarchy

System.Object
   TwinCAT.Ads.TypeSystem.Instance [1388]
      TwinCAT.Ads.TypeSystem.Field [1374]
         TwinCAT.Ads.TypeSystem.Member

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public sealed class Member : Field,
   IMember, IField, IAttributedInstance, IInstance, IBitSize

The Member type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1391]</td>
<td>Gets the Type Attributes. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>BitOffset [1409]</td>
<td>Gets the bit offset.</td>
</tr>
<tr>
<td>BitSize [1392]</td>
<td>Gets the size of this Instance [1388] in bits. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>ByteOffset [1409]</td>
<td>Gets the byte offset.</td>
</tr>
<tr>
<td>ByteSize [1392]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>Category [1393]</td>
<td>Gets the the DataTypeCategory [1649] of the Instance. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>Comment [1393]</td>
<td>Gets the comment. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>ContextMask [1394]</td>
<td>Gets the context mask of this instance. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>DataType [1394]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>HasValue [1395]</td>
<td>Gets a value indicating whether this instance has a value. (Inherited from Instance [1388].)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance.</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads.TypeSystem Namespace [1328]
### 6.7.11.1 Member Properties

The Member [1406] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1391]</td>
<td>Gets the Type Attributes. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>BitOffset [1409]</td>
<td>Gets the bit offset.</td>
</tr>
<tr>
<td>BitSize [1392]</td>
<td>Gets the size of this Instance [1388] in bits. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>ByteOffset [1409]</td>
<td>Gets the byte offset.</td>
</tr>
<tr>
<td>ByteSize [1392]</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>Category [1393]</td>
<td>Gets the the DataTypeCategory [1649] of the Instance. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>Comment [1393]</td>
<td>Gets the comment. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>ContextMask [1394]</td>
<td>Gets the context mask of this instance. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>DataType [1394]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>HasValue [1395]</td>
<td>Gets a value indicating whether this instance has a value. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>InstanceName [1395]</td>
<td>Gets the name of the instance (without periods (.) (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>InstancePath [1396]</td>
<td>Gets the relative / absolute access path to the instance (with periods ( . ) (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsBitType [1396]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsByteAligned [1397]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsPersistent [1397]</td>
<td>Indicates that this instance is persistent. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsPointer [1398]</td>
<td>Gets a value indicating whether this instance is reference. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsReadOnly [1398]</td>
<td>Indicates that this instance is read only. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsReference [1399]</td>
<td>Gets a value indicating whether this instance is reference. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsStatic [1399]</td>
<td>Gets a value indicating whether this IInstance [2052] is static. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsTcComInterfacePointer [1400]</td>
<td>Indicates that this instance is a TcComInterfacePointer. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsTypeGuid [1400]</td>
<td>Indicates that this instance has set TypeGuid flag. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>Namespace [1401]</td>
<td>Gets the namespace name. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Offset [1410]</td>
<td>Gets the offset of the Member [1406] within the parent StructType [1457] in bits or bytes dependent on IsBitType [1396].</td>
</tr>
<tr>
<td>ParentType [1378]</td>
<td>Gets the Parent of this IField [2040]. (Inherited from Field [1374].)</td>
</tr>
<tr>
<td>Size [1401]</td>
<td>Gets the size of the IDataType [1986] in bytes or Bits dependant on IsBitType [1396] (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>TypeName [1402]</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>ValueEncoding [1378]</td>
<td>Gets the value encoding of this Field [1374] (Inherited from Field [1374].)</td>
</tr>
</tbody>
</table>

**Reference**

Member Class [1406]

TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.11.1.1 Member.BitOffset Property

Gets the bit offset.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int BitOffset { get; }
```

**Property Value**

Type: Int32

The bit offset.

**Implements**

IMember.BitOffset [2067]

**Reference**

Member Class [1406]

TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.11.1.2 Member.ByteOffset Property

Gets the byte offset.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
### TwinCAT.Ads Namespaces

#### Syntax

**C#**

```csharp
public int ByteOffset { get; }
```

**Property Value**

Type: `Int32`
The byte offset.

**Implements**

`IMember.ByteOffset` [2067]

**Reference**

- **Member Class** [1406]
- **TwinCAT.Ads.TypeSystem Namespace** [1328]

#### 6.7.11.1.3 Member.Offset Property

Gets the offset of the `Member` [1406] within the parent `StructType` [1457] in bits or bytes dependent on `IsBitType` [1396]

**Namespace:** TwinCAT.Ads.TypeSystem [1328]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public int Offset { get; }
```

**Property Value**

Type: `Int32`
The offset.

**Implements**

`IMember.Offset` [2068]

**Reference**

- **Member Class** [1406]
- **TwinCAT.Ads.TypeSystem Namespace** [1328]

#### 6.7.11.2 Member Methods

The `Member` [1406] type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents this instance. (Inherited from <code>Instance</code>)</td>
</tr>
</tbody>
</table>

Reference

Member Class [ ในไทย]

`TwinCAT.Ads.TypeSystem Namespace` [ ในไทย]

### 6.7.12 PCCHType Class

Class PCCHType. This class cannot be inherited. Implements the `PointerType` [ ในไทย]

Inheritance Hierarchy

- `System.Object`
  - `TwinCAT.Ads.TypeSystem.DataType` [ ในไทย]
    - `TwinCAT.Ads.TypeSystem.PointerType` [ ในไทย]
      - `TwinCAT.Ads.TypeSystem.PCCHType`

Namespace: `TwinCAT.Ads.TypeSystem` [ ในไทย]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

**C#**

```csharp
public sealed class PCCHType : PointerType
```

The PCCHType type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the <code>IDataType</code> [ ในไทย] (Inherited from <code>DataType</code> [ ในไทย])</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>DataType</code> [ ในไทย] in bits. (Inherited from <code>DataType</code> [ ในไทย])</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from <code>DataType</code> [ ในไทย])</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from <code>DataType</code> [ ในไทย])</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from <code>DataType</code> [ ในไทย])</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the <code>IDataType</code> [ จากภาษาไทย] (Namespace + Name)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>DataType</code> [ ในไทย])</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the DataType (Inherited from <code>DataType</code> [ ในไทย])</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsBitType [1356]</td>
<td>Gets a value indicating whether this [DataType [1986] is a bit mapping Type (Inherited from DataType [1349]).)</td>
</tr>
<tr>
<td>IsByteAligned [1356]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsContainer [1357]</td>
<td>Gets a value indicating whether this [DataType [1986] is a container type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsPointer [1357]</td>
<td>Gets a value indicating whether this [DataType [1986] is a pointer type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsPrimitive [1358]</td>
<td>Gets a value indicating whether this [DataType [1986] is primitive (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsReference [1359]</td>
<td>Gets a value indicating whether this [DataType [1986] is a reference type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ManagedType [1417]</td>
<td>Gets the corresponding .NET Type if attached. (Inherited fromPointerType [1414].)</td>
</tr>
<tr>
<td>Name [1360]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Namespace [1360]</td>
<td>Gets the namespace string within the [DataType [1986] exists. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ReferencedType [1417]</td>
<td>Gets the the referenced type. (Inherited from PointerType [1414].)</td>
</tr>
<tr>
<td>ReferenceTypeName [1417]</td>
<td>Gets the name of the referenced datatype (Inherited from PointerType [1414].)</td>
</tr>
<tr>
<td>Size [1361]</td>
<td>Gets the Size of the DataType [1349] in Bytes or bits. (Inherited from DataType [1349].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString [1363]</td>
<td>Returns a String that represents this instance. (Inherited from DataType [1349].)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads.TypeSystem Namespace [1328]

TwinCAT.Ads.TypeSystem.PointerType [1414]

#### 6.7.12.1 PCCHType Properties

The PCCHType [1411] type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1352]</td>
<td>Gets the attributes of the [IDataType] [1986] (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>BitSize [1353]</td>
<td>Gets the size of the [DataType] [1349] in bits. (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>ByteSize [1353]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>Category [1354]</td>
<td>Gets the Data Type category (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>Comment [1354]</td>
<td>Gets the comment. (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>FullName [1355]</td>
<td>Gets the full name of the [IDataType] [1986] (Namespace + Name) (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>Id [1355]</td>
<td>Gets the ID of the DataType (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>IsBitType [1356]</td>
<td>Gets a value indicating whether this [IDataType] [1986] is a bit mapping Type (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>IsByteAligned [1356]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>IsContainer [1357]</td>
<td>Gets a value indicating whether this [IDataType] [1986] is a container type (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>IsPointer [1357]</td>
<td>Gets a value indicating whether this [IDataType] [1986] is a pointer type (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>IsPrimitive [1358]</td>
<td>Gets a value indicating whether this [IDataType] [1986] is primitive (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>IsReference [1359]</td>
<td>Gets a value indicating whether this [IDataType] [1986] is a reference type (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>ManagedType [1417]</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from [PointerType] [1414].)</td>
</tr>
<tr>
<td>Name [1360]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>Namespace [1360]</td>
<td>Gets the namespace string within the [IDataType] [1986] exists. (Inherited from [DataType] [1349].)</td>
</tr>
<tr>
<td>ReferencedType [1417]</td>
<td>Gets the the referenced type. (Inherited from [PointerType] [1414].)</td>
</tr>
<tr>
<td>ReferenceTypeName [1417]</td>
<td>Gets the name of the referenced datatype (Inherited from [PointerType] [1414].)</td>
</tr>
<tr>
<td>Size [1361]</td>
<td>Gets the Size of the [DataType] [1349] in Bytes or bits. (Inherited from [DataType] [1349].)</td>
</tr>
</tbody>
</table>

## Reference

- [PCCHType Class [1411]]
- [TwinCAT.Ads.TypeSystem Namespace [1328]]

### 6.7.12.2 PCCHType Methods

The PCCHType [1411] type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance.</td>
</tr>
</tbody>
</table>

### Reference

- PCCHType Class [1411]
- TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.13 PointerType Class

Represents a pointer type.

#### Inheritance Hierarchy

```
System.Object
    TwinCAT.Ads.TypeSystem.DataType [1349]
    TwinCAT.Ads.TypeSystem.PointerType
    TwinCAT.Ads.TypeSystem.PCCHType [1411]
    TwinCAT.Ads.TypeSystem.PVoidType [1422]
```

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public class PointerType : DataType,
    IPointerType, IDataType, IBitSize
```

The PointerType type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the DataType [1349] in bits. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name)</td>
</tr>
</tbody>
</table>
```

---

**Version:** 1.1  **TC1000**
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id [1355]</td>
<td>Gets the ID of the DataType (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsBitType [1356]</td>
<td>Gets a value indicating whether this [DataType [1349] is a bit mapping Type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsByteAligned [1356]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsContainer [1357]</td>
<td>Gets a value indicating whether this [DataType [1349] is a container type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsPointer [1357]</td>
<td>Gets a value indicating whether this [DataType [1349] is a pointer type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsPrimitive [1358]</td>
<td>Gets a value indicating whether this [DataType [1349] is primitive (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsReference [1359]</td>
<td>Gets a value indicating whether this [DataType [1349] is a reference type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ManagedType [1417]</td>
<td>Gets the corresponding .NET Type if attached. (Overrides DataType MANagedType [1359].)</td>
</tr>
<tr>
<td>Name [1360]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Namespace [1360]</td>
<td>Gets the namespace string within the [DataType [1349] exists. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ReferencedType [1417]</td>
<td>Gets the referenced type.</td>
</tr>
<tr>
<td>ReferenceTypeName [1417]</td>
<td>Gets the name of the referenced datatype</td>
</tr>
<tr>
<td>Size [1361]</td>
<td>Gets the Size of the DataType [1349] in Bytes or bits. (Inherited from DataType [1349].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString [1363]</td>
<td>Returns a String that represents this instance. (Inherited from DataType [1349].)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.TypeSystem Namespace [1328]
### 6.7.13.1 PointerType Properties

The PointerType [1414] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1352]</td>
<td>Gets the attributes of the IDatatype [1986] (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>BitSize [1353]</td>
<td>Gets the size of the Datatype [1349] in bits. (Inherited from Datatype [1349].)</td>
</tr>
<tr>
<td>ByteSize [1353]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from Datatype [1349].)</td>
</tr>
<tr>
<td>Category [1354]</td>
<td>Gets the Data Type category (Inherited from Datatype [1349].)</td>
</tr>
<tr>
<td>Comment [1354]</td>
<td>Gets the comment. (Inherited from Datatype [1349].)</td>
</tr>
<tr>
<td>FullName [1355]</td>
<td>Gets the full name of the IDatatype [1986] (Namespace + Name) (Inherited from Datatype [1349].)</td>
</tr>
<tr>
<td>Id [1355]</td>
<td>Gets the ID of the Datatype (Inherited from Datatype [1349].)</td>
</tr>
<tr>
<td>IsBitType [1356]</td>
<td>Gets a value indicating whether this IDatatype [1986] is a bit mapping Type (Inherited from Datatype [1349].)</td>
</tr>
<tr>
<td>IsByteAligned [1356]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from Datatype [1349].)</td>
</tr>
<tr>
<td>IsContainer [1357]</td>
<td>Gets a value indicating whether this IDatatype [1986] is a container type (Inherited from Datatype [1349].)</td>
</tr>
<tr>
<td>IsPointer [1357]</td>
<td>Gets a value indicating whether this IDatatype [1986] is a pointer type (Inherited from Datatype [1349].)</td>
</tr>
<tr>
<td>IsPrimitive [1358]</td>
<td>Gets a value indicating whether this IDatatype [1986] is primitive (Inherited from Datatype [1349].)</td>
</tr>
<tr>
<td>IsReference [1359]</td>
<td>Gets a value indicating whether this IDatatype [1986] is a reference type (Inherited from Datatype [1349].)</td>
</tr>
<tr>
<td>ManagedType [1417]</td>
<td>Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType [1359].)</td>
</tr>
<tr>
<td>Name [1360]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from Datatype [1349].)</td>
</tr>
<tr>
<td>Namespace [1360]</td>
<td>Gets the namespace string within the IDatatype [1986] exists. (Inherited from Datatype [1349].)</td>
</tr>
<tr>
<td>ReferencedType [1417]</td>
<td>Gets the the referenced type.</td>
</tr>
<tr>
<td>ReferenceTypeName [1417]</td>
<td>Gets the name of the referenced datatype</td>
</tr>
<tr>
<td>Size [1361]</td>
<td>Gets the Size of the Datatype [1349] in Bytes or bits. (Inherited from Datatype [1349].)</td>
</tr>
</tbody>
</table>

#### Reference

PointerType Class [1414]

TwinCAT.Ads.TypeSystem Namespace [1328]
6.7.13.1.1 **PointerType.ManagedType Property**

Gets the corresponding .NET Type if attached.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public override Type ManagedType { get; }
```

**Property Value**

Type: `Type`
Dot net type.

**Reference**

**PointerType Class**

**TwinCAT.Ads.TypeSystem Namespace**

---

6.7.13.1.2 **PointerType.ReferencedType Property**

Gets the the referenced type.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public IDataType ReferencedType { get; }
```

**Property Value**

Type: `IDataType`
The type of the referenced.

**Implements**

[PointerType.ReferencedType](#)

**Reference**

**PointerType Class**

**TwinCAT.Ads.TypeSystem Namespace**

---

6.7.13.1.3 **PointerType.ReferencedTypeName Property**

Gets the name of the referenced datatype
6.7.13.2 PointerType Methods

The `PointerType` type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td><code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents this instance. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

Reference

PointerType Class [1414]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.14 PrimitiveType Class

Class PrimitiveType.
Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [1349]
  TwinCAT.Ads.TypeSystem.PrimitiveType

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd72bc0ea15da1c14

Syntax

C#

```csharp
public class PrimitiveType : DataType,
    IPrimitiveType, IDataType, IBitSize
```

The PrimitiveType type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the <a href="#1986">IDataType</a> (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <a href="#1349">DataType</a> in bits. (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the <a href="#1986">IDataType</a> (Namespace + Name) (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the DataType (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this <a href="#1986">IDataType</a> is a bit mapping Type (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this <a href="#1986">IDataType</a> is a container type (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this <a href="#1986">IDataType</a> is a pointer type (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this <a href="#1986">IDataType</a> is primitive (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this <a href="#1986">IDataType</a> is a reference type (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the <a href="#1986">IDataType</a> exists. (Inherited from <a href="#1349">DataType</a>.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrimitiveFlags</td>
<td>Indicates types of different PrimitiveTypes with flags.</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the Size of the DataType in Bytes or bits. (Inherited from DataType.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DataType.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads.TypeSystem Namespace [1328]
TwinCAT.Ads.TypeSystem.DataType [1349]

### 6.7.14.1 PrimitiveType Properties

The PrimitiveType type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType (Inherited from DataType.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the DataType in bits. (Inherited from DataType.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from DataType.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from DataType.)</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the IDataType (Namespace + Name) (Inherited from DataType.)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the DataType (Inherited from DataType.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this IDataType is a bit mapping Type (Inherited from DataType.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsContainer ✨1357</td>
<td>Gets a value indicating whether this [DataType ✨1986] is a container type (Inherited from [DataType ✨1349].)</td>
</tr>
<tr>
<td>IsPointer ✨1357</td>
<td>Gets a value indicating whether this [DataType ✨1986] is a pointer type (Inherited from [DataType ✨1349].)</td>
</tr>
<tr>
<td>IsPrimitive ✨1358</td>
<td>Gets a value indicating whether this [DataType ✨1986] is primitive (Inherited from [DataType ✨1349].)</td>
</tr>
<tr>
<td>IsReference ✨1359</td>
<td>Gets a value indicating whether this [DataType ✨1986] is a reference type (Inherited from [DataType ✨1349].)</td>
</tr>
<tr>
<td>ManagedType ✨1359</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from [DataType ✨1349].)</td>
</tr>
<tr>
<td>Name ✨1360</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from [DataType ✨1349].)</td>
</tr>
<tr>
<td>Namespace ✨1360</td>
<td>Gets the namespace string within the [DataType ✨1986] exists. (Inherited from [DataType ✨1349].)</td>
</tr>
<tr>
<td>PrimitiveFlags ✨1421</td>
<td>Indicates types of different PrimitiveTypes with flags.</td>
</tr>
<tr>
<td>Size ✨1361</td>
<td>Gets the Size of the [DataType ✨1349] in Bytes or bits. (Inherited from [DataType ✨1349].)</td>
</tr>
</tbody>
</table>

**Reference**

PrimitiveType Class ✨1418

TwinCAT.Ads.TypeSystem Namespace ✨1328

**6.7.14.1.1 PrimitiveType.PrimitiveFlags Property**

Indicates types of different PrimitiveTypes with flags.

**Namespace:** TwinCAT.Ads.TypeSystem ✨1328  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
public PrimitiveTypeFlags PrimitiveFlags { get; }
```

**Property Value**

Type: PrimitiveTypeFlags ✨2289  
The primitive flags.

**Implements**

IPrimitiveType.PrimitiveFlags ✨2091

**Reference**

PrimitiveType Class ✨1418

TwinCAT.Ads.TypeSystem Namespace ✨1328
6.7.14.2 PrimitiveType Methods

The PrimitiveType type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DataType.)</td>
</tr>
</tbody>
</table>

**Reference**

PrimitiveType Class

TwinCAT.Ads.TypeSystem Namespace

6.7.15 PVoidType Class

Class PVoidType. This class cannot be inherited. Implements the PointerType.

**Inheritance Hierarchy**

- System.Object
  - TwinCAT.Ads.TypeSystem.DataType
  - TwinCAT.Ads.TypeSystem.PointerType
  - TwinCAT.Ads.TypeSystem.PVoidType

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public sealed class PVoidType : PointerType
```

The PVoidType type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType (Inherited from DataType.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the DataType in bits. (Inherited from DataType.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ByteSize [1.1353]</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from</td>
</tr>
<tr>
<td></td>
<td>DataType [1.1349].)</td>
</tr>
<tr>
<td>Category [1.1354]</td>
<td>Gets the Data Type category (Inherited from DataType [1.1349].)</td>
</tr>
<tr>
<td>Comment [1.1354]</td>
<td>Gets the comment. (Inherited from DataType [1.1349].)</td>
</tr>
<tr>
<td>FullName [1.1355]</td>
<td>Gets the full name of the IDatatype [1.1986] (Namespace + Name)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DataType [1.1349].)</td>
</tr>
<tr>
<td>Id [1.1355]</td>
<td>Gets the ID of the DataType (Inherited from DataType [1.1349].)</td>
</tr>
<tr>
<td>IsBitType [1.1356]</td>
<td>Gets a value indicating whether this IDatatype [1.1986] is a bit mapping</td>
</tr>
<tr>
<td></td>
<td>Type (Inherited from DataType [1.1349].)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DataType [1.1349].)</td>
</tr>
<tr>
<td>IsContainer [1.1357]</td>
<td>Gets a value indicating whether this IDatatype [1.1986] is a container type</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DataType [1.1349].)</td>
</tr>
<tr>
<td>IsPointer [1.1357]</td>
<td>Gets a value indicating whether this IDatatype [1.1986] is a pointer type</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DataType [1.1349].)</td>
</tr>
<tr>
<td>IsPrimitive [1.1358]</td>
<td>Gets a value indicating whether this IDatatype [1.1986] is primitive</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DataType [1.1349].)</td>
</tr>
<tr>
<td>IsReference [1.1359]</td>
<td>Gets a value indicating whether this IDatatype [1.1986] is a reference type</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DataType [1.1349].)</td>
</tr>
<tr>
<td>ManagedType [1.1417]</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from PointerType</td>
</tr>
<tr>
<td></td>
<td>[1.1414].)</td>
</tr>
<tr>
<td>Name [1.1360]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from</td>
</tr>
<tr>
<td></td>
<td>DataType [1.1349].)</td>
</tr>
<tr>
<td>Namespace [1.1360]</td>
<td>Gets the namespace string within the IDatatype [1.1986] exists. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from DataType [1.1349].)</td>
</tr>
<tr>
<td>ReferencedType</td>
<td>Gets the the referenced type. (Inherited from PointerType [1.1414].)</td>
</tr>
<tr>
<td>[1.1417]</td>
<td></td>
</tr>
<tr>
<td>ReferenceTypeName</td>
<td>Gets the name of the referenced datatype (Inherited from PointerType</td>
</tr>
<tr>
<td></td>
<td>[1.1414].)</td>
</tr>
<tr>
<td>Size [1.1361]</td>
<td>Gets the Size of the DataType [1.1349] in Bytes or bits. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>DataType [1.1349].)</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString [1.1363]</td>
<td>Returns a String that represents this instance. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>DataType [1.1349].)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.Ads.TypeSystem Namespace [1.1328]
6.7.15.1 **PVoidType Properties**

The `PVoidType` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the <code>IDataType</code> (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>DataType</code> in bits. (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the <code>IDataType</code> (Namespace + Name) (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the DataType (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a bit mapping Type (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a container type (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a pointer type (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this <code>IDataType</code> is primitive (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a reference type (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from <code>PointerType</code>).</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the <code>IDataType</code> exists. (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>ReferencedType</td>
<td>Gets the the referenced type. (Inherited from <code>PointerType</code>).</td>
</tr>
<tr>
<td>ReferenceTypeName</td>
<td>Gets the name of the referenced datatype (Inherited from <code>PointerType</code>).</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the Size of the <code>DataType</code> in Bytes or bits. (Inherited from <code>DataType</code>).</td>
</tr>
</tbody>
</table>
Reference

PVoidType Class [1422]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.15.2 PVoidType Methods

The PVoidType [1422] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance.</td>
</tr>
</tbody>
</table>

Reference

PVoidType Class [1422]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.16 ReferenceType Class

Represents a reference type

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [1349]
  TwinCAT.Ads.TypeSystem.ReferenceType

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public sealed class ReferenceType : DataType,
    IReferenceType, IDataType, IBitSize
```

The ReferenceType type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the DataType [1349] in bits. (Inherited from DataType [1349].)</td>
</tr>
</tbody>
</table>
# TwinCAT.Ads Namespaces

## Name | Description
--- | ---
ByteSize | Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType.)
Category | Gets the Data Type category (Inherited from DataType.)
Comment | Gets the comment. (Inherited from DataType.)
FullName | Gets the full name of the IDatatype (Namespace + Name) (Inherited from DataType.)
Id | Gets the ID of the Data Type (Inherited from DataType.)
IsBitType | Gets a value indicating whether this IDatatype is a bit mapping Type (Inherited from DataType.)
IsByteAligned | Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType.)
IsContainer | Gets a value indicating whether this IDatatype is a container type (Overrides DataType.IsContainer.)
IsPointer | Gets a value indicating whether this IDatatype is a pointer type (Inherited from DataType.)
IsPrimitive | Gets a value indicating whether this IDatatype is primitive (Overrides DataType.IsPrimitive.)
IsReference | Gets a value indicating whether this IDatatype is a reference type (Inherited from DataType.)
ManagedType | Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType.)
Name | Gets the name of the Data Type (without namespace) (Inherited from DataType.)
Namespace | Gets the namespace string within the IDatatype (Inherited from DataType.)
ReferencedType | Gets the the referenced type.
ReferencedTypeName | Gets the name of the referenced type.
ResolvedByteSize | Gets the size of the resolved byte.
ResolvedCategory | Gets the resolved category.
ResolvedType | Gets the type of the resolved.
Size | Gets the Size of the DataType in Bytes or bits. (Inherited from DataType.)

## Methods

| Name | Description |
--- | --- |
Equals | Determines whether the specified object is equal to the current object. (Inherited from Object.)
GetHashCode | Serves as the default hash function. (Inherited from Object.)
GetType | Gets the Type of the current instance. (Inherited from Object.)
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DataType)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.16.1 ReferenceType Properties

The ReferenceType [1425] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the DataType [1349] in bits. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the Data Type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this IDataType [1986] is a bit mapping Type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this IDataType [1986] is a container type (Overrides DataType.IsContainer [1357].)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this IDataType [1986] is a pointer type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IDataType [1986] is primitive (Overrides DataType.IsPrimitive [1358].)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this IDataType [1986] is a reference type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType [1359].)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the IDataType [1986] exists. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ReferencedType</td>
<td>Gets the the referenced type.</td>
</tr>
</tbody>
</table>
### ReferenceType.IsContainer Property

Gets a value indicating whether this `IDataType` is a container type.

#### Syntax

**C#**

```csharp
public override bool IsContainer { get; }
```

#### Property Value

Type: Boolean

true if this instance is container type; otherwise, false.

#### Remarks

Container Types are all types that contain SubElements like

- `Array`
- `Pointer`
- `Union`
- `Struct`
- `Function`
- `FunctionBlock`
- `Program`
and the Alias [1649] and Reference [1649] types, if they have a container type as base type.

Reference

ReferenceType Class [1425]
TwinCAT.Ads.TypeSystem Namespace [1328]
IDataType.Category [1988]

6.7.16.1.2 ReferenceType.IsPrimitive Property

Gets a value indicating whether this IDataType [1986] is primitive

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public override bool IsPrimitive { get; }

Property Value

Type: Boolean
true if this instance is primitive; otherwise, false.

Implements

IDataType.IsPrimitive [1991]
IDataType.IsPrimitive [1991]

Reference

ReferenceType Class [1425]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.16.1.3 ReferenceType.ManagedType Property

Gets the corresponding .NET Type if attached.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public override Type ManagedType { get; }

Property Value

Type: Type
Dot net type.
6.7.16.1.4 ReferenceType.ReferencedType Property

Gets the referenced type.

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public IDataType ReferencedType { get; }

Property Value

Type: IDataType
The type of the referenced.

Implements

IReferenceType.ReferencedType

6.7.16.1.5 ReferenceType.ReferencedTypeName Property

Gets the name of the referenced type.

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public string ReferencedTypeName { get; }

Property Value

Type: String
The name of the referenced type.

Implements

IReferenceType.ReferencedTypeName
6.7.16.1.6  ReferenceType.ResolvedByteSize Property

Gets the size of the resolved byte.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public int ResolvedByteSize { get; }
```

**Property Value**

- **Type:** Int32
  - The size of the resolved byte.

**Implements**

IReferenceType.ResolvedByteSize

6.7.16.1.7  ReferenceType.ResolvedCategory Property

Gets the resolved category.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public DataTypeCategory ResolvedCategory { get; }
```

**Property Value**

- **Type:** DataTypeCategory
  - The resolved category.

**Implements**

IReferenceType.ResolvedCategory
6.7.16.1.8  ReferenceType.ResolvedType Property

Gets the type of the resolved.

**Namespace:**  TwinCAT.Ads.TypeSystem [1328]

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public IDataType ResolvedType { get; }
```

**Property Value**

Type:  `IDataType` [1986]

The type of the resolved.

**Implements**

`IReferenceType.ResolvedType` [2103]

### ReferenceType Methods

The `ReferenceType` [1425] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a <code>String</code> that represents this instance. (Inherited from <code>DataType</code> [1349].)</td>
</tr>
</tbody>
</table>

**Reference**

**ReferenceType Class [1425]**

**TwinCAT_Ads_TypeSystem Namespace [1328]**
6.7.17   **RpcMethod Class**

**RPC Method Description**

**Inheritance Hierarchy**

*System.Object*
  *TwinCAT.Ads.TypeSystem.RpcMethod*

**Namespace:** TwinCAT.Ads.TypeSystem [► 1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public class RpcMethod : IRpcMethod
```

The `RpcMethod` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comment [► 1434]</td>
<td>Gets the Method comment.</td>
</tr>
<tr>
<td>InParameters [► 1435]</td>
<td>Gets the In-Parameters of the IRpcMethod [► 2123]</td>
</tr>
<tr>
<td>IsVoid [► 1435]</td>
<td>Gets a value indicating whether this IRpcMethod [► 2123] has no return parameter</td>
</tr>
<tr>
<td>Name [► 1436]</td>
<td>Gets the name of the method</td>
</tr>
<tr>
<td>OutParameters [► 1436]</td>
<td>Gets the Out-Parameters of the IRpcMethod [► 2123]</td>
</tr>
<tr>
<td>Parameters [► 1437]</td>
<td>Gets all parameters (In, Out and ref parameters) of the _ [► 2123]</td>
</tr>
<tr>
<td>ReturnAlignSize [► 1437]</td>
<td>Gets the size of the biggest element in bytes for Alignment</td>
</tr>
<tr>
<td>ReturnType [► 1438]</td>
<td>Gets the return type.</td>
</tr>
<tr>
<td>ReturnTypeSize [► 1438]</td>
<td>Gets the Byte size of the return type.</td>
</tr>
<tr>
<td>VTableIndex [► 1439]</td>
<td>Gets the V-table index of the method.</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
6.7.17.1  RpcMethod Properties

The RpcMethod type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comment</td>
<td>Gets the Method comment.</td>
</tr>
<tr>
<td>InParameters</td>
<td>Gets the In-Parameters of the IRpcMethod.</td>
</tr>
<tr>
<td>IsVoid</td>
<td>Gets a value indicating whether this IRpcMethod has no return parameter.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the method.</td>
</tr>
<tr>
<td>OutParameters</td>
<td>Gets the Out-Parameters of the IRpcMethod.</td>
</tr>
<tr>
<td>Parameters</td>
<td>Gets all parameters (In, Out and ref parameters) of the IRpcMethod.</td>
</tr>
<tr>
<td>ReturnAlignSize</td>
<td>Gets the size of the biggest element in bytes for Alignment</td>
</tr>
<tr>
<td>ReturnType</td>
<td>Gets the return type.</td>
</tr>
<tr>
<td>ReturnTypeSize</td>
<td>Gets the Byte size of the return type.</td>
</tr>
<tr>
<td>VTableIndex</td>
<td>Gets the V-table index of the method.</td>
</tr>
</tbody>
</table>

Reference

RpcMethod Class [1433]

TwinCAT.Ads.TypeSystem Namespace [1328]
Property Value

Type: String
The comment.

Implements

IRpcMethod.Comment [2124]

Reference

RpcMethod Class [1433]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.17.1.2 RpcMethod.InParameters Property

Gets the In-Parameters of the IRpcMethod [2123]

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public IRpcMethodParameterCollection InParameters { get; }

Property Value

Type: IRpcMethodParameterCollection [2137]
The In- and Ref-Parameters

Implements

IRpcMethod.InParameters [2124]

Reference

RpcMethod Class [1433]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.17.1.3 RpcMethod.IsVoid Property

Gets a value indicating whether this IRpcMethod [2123] has no return parameter

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool IsVoid { get; }
Property Value

Type: Boolean
true if this instance is void; otherwise, false.

Implements

IRpcMethod.IsVoid [2125]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

RpcMethod Class [1433]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.17.1.4 RpcMethod.Name Property

Gets the name of the method

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public string Name { get; }
```

Property Value

Type: String
The name.

Implements

IRpcMethod.Name [2125]

Reference

RpcMethod Class [1433]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.17.1.5 RpcMethod.OutParameters Property

Gets the Out-Parameters of the IRpcMethod [2123]
**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

### Syntax

**C#**
```
public IRpcMethodParameterCollection OutParameters { get; }
```

### Property Value

**Type:** IRpcMethodParameterCollection [2137]

The In- and Ref-Parameters

**Implements**

IRpcMethod.OutParameters [2126]

### Reference

RpcMethod Class [1433]

TwinCAT.Ads.TypeSystem Namespace [1328]

---

### 6.7.17.1.6 RpcMethod.Parameters Property

Gets all parameters (In, Out and ref parameters) of the .

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**
```
public IRpcMethodParameterCollection Parameters { get; }
```

**Property Value**

**Type:** IRpcMethodParameterCollection [2137]

The parameters.

**Implements**

IRpcMethod.Parameters [2126]

### Reference

RpcMethod Class [1433]

TwinCAT.Ads.TypeSystem Namespace [1328]

---

### 6.7.17.1.7 RpcMethod.ReturnAlignSize Property

Gets the size of the biggest element in bytes for Alignment
**Namespace:** TwinCAT.Ads.TypeSystem

**Syntax**

C#  
```csharp
public int ReturnAlignSize { get; }
```

**Property Value**

**Type:** Int32  
The size of the return align.

**Reference**

RpcMethod Class [1433]

TwinCAT.Ads.TypeSystem Namespace [1328]

---

### 6.7.17.1.8 RpcMethod.ReturnType Property

Gets the return type.

**Namespace:** TwinCAT.Ads.TypeSystem  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public string ReturnType { get; }
```

**Property Value**

**Type:** String  
Return type.

**Implements**

IRpcMethod.ReturnType [2126]

**Reference**

RpcMethod Class [1433]

TwinCAT.Ads.TypeSystem Namespace [1328]

---

### 6.7.17.1.9 RpcMethod.ReturnTypeSize Property

Gets the Byte size of the return type.

**Namespace:** TwinCAT.Ads.TypeSystem  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public int ReturnTypeSize { get; }
```

**Property Value**

Type: `Int32`
The size of the return type.

**Implements**

`IRpcMethod.ReturnTypeSize`[

---

**Reference**

`RpcMethod Class`[

`TwinCAT.Ads.TypeSystem Namespace`[

---

6.7.17.1.10 **RpcMethod.VTableIndex Property**

Gets the V-table index of the method.

**Namespace:** TwinCAT.Ads.TypeSystem[

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public int VTableIndex { get; }
```

**Property Value**

Type: `Int32`
The index of the v table.

**Reference**

`RpcMethod Class`[

`TwinCAT.Ads.TypeSystem Namespace`[

---

6.7.17.2 **RpcMethod Methods**

The `RpcMethod`[

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ equality icon ]</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Overrides Object.ToString.)</td>
</tr>
</tbody>
</table>

### Reference

 RpcMethod Class [1433]
 TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.17.2.1  RpcMethod.ToString Method

Returns a String that represents this instance.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public override string ToString()
```

**Return Value**

Type: String

A String that represents this instance.

### Reference

 RpcMethod Class [1433]
 TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.18  RpcMethodParameter Class

Class RpcMethodParameter.

**Inheritance Hierarchy**

System.Object
   TwinCAT.Ads.TypeSystem.RpcMethodParameter

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public class RpcMethodParameter : IRpcMethodParameter
```

The RpcMethodParameter type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlignSize</td>
<td>Gets the size of biggest element for alignment</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the Parameter Comment.</td>
</tr>
<tr>
<td>HasLengthIsParameter</td>
<td>Gets a value indicating whether this instance has a related LengthIs Parameter.</td>
</tr>
<tr>
<td>LengthIsParameterIndex</td>
<td>Gets the index of the LengthIs parameter (within the MethodParameter List)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the Parameter Name</td>
</tr>
<tr>
<td>ParameterFlags</td>
<td>Gets the parameter flags.</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the RpcMethodParameter</td>
</tr>
<tr>
<td>&gt;TypeGuid</td>
<td>Gets the Unique identifier of the parameters data type.</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the Data type of the Parameter</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.Ads.TypeSystem Namespace ![1328](image)

6.7.18.1 RpcMethodParameter Properties

The RpcMethodParameter ![1440](image) type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlignSize [1442]</td>
<td>Gets the size of biggest element for alignment</td>
</tr>
<tr>
<td>Comment [1443]</td>
<td>Gets the Parameter Comment.</td>
</tr>
<tr>
<td>HasLengthIsParameter</td>
<td>Gets a value indicating whether this instance has a related LengthIs Parameter.</td>
</tr>
<tr>
<td>LengthIsParameterIndex [1443]</td>
<td>Gets the index of the LengthIs parameter (within the MethodParameter List)</td>
</tr>
<tr>
<td>Name [1444]</td>
<td>Gets the Parameter Name</td>
</tr>
<tr>
<td>ParameterFlags [1445]</td>
<td>Gets the parameter flags.</td>
</tr>
<tr>
<td>Size [1445]</td>
<td>Gets the size of the RpcMethodParameter [1440]</td>
</tr>
<tr>
<td>TypeGuid [1446]</td>
<td>Gets the Unique identifier of the parameters data type.</td>
</tr>
<tr>
<td>TypeName [1446]</td>
<td>Gets the Data type of the Parameter</td>
</tr>
</tbody>
</table>

### Reference

 RpcMethodParameter Class [1440]

TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.18.1.1 RpcMethodParameter.AlignSize Property

Gets the size of biggest element for alignment

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int AlignSize { get; }
```

**Property Value**

Type: Int32

The size of the align.

**Reference**

RpcMethodParameter Class [1440]

TwinCAT.Ads.TypeSystem Namespace [1328]
6.7.18.1.2  RpcMethodParameter.Comment Property

Gets the Parameter Comment.

**Namespace:**  TwinCAT.Ads.TypeSystem

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public string Comment { get; }
```

**Property Value**

**Type:**  String

The comment.

**Reference**

RpcMethodParameter Class

TwinCAT.Ads.TypeSystem Namespace

6.7.18.1.3  RpcMethodParameter.HasLengthIsParameter Property

Gets a value indicating whether this instance has a related LengthIs Parameter.

**Namespace:**  TwinCAT.Ads.TypeSystem

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool HasLengthIsParameter { get; }
```

**Property Value**

**Type:**  Boolean

true if this instance has a LengthIs parameter; otherwise, false.

**Implements**

IRpcMethodParameter.HasLengthIsParameter

**Reference**

RpcMethodParameter Class

TwinCAT.Ads.TypeSystem Namespace

6.7.18.1.4  RpcMethodParameter.LengthIsParameterIndex Property

Gets the index of the LengthIs parameter (within the MethodParameter List)
**TwinCAT.Ads Namespaces**

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int LengthIsParameterIndex { get; }
```

**Property Value**

Type: `Int32`
The index of the length is parameter.

**Implements**

IRpcMethodParameter.LengthIsParameterIndex

**Remarks**

This field references to the Parameter that defines the length for this generic one. Equally to the marshalling attributes of COM (sizeof, length) this enables to transport parameter of type (PVOID)

**Reference**

RpcMethodParameter Class
TwinCAT.Ads.TypeSystem Namespace

---

6.7.18.1.5 **RpcMethodParameter.Name Property**

Gets the Parameter Name

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public string Name { get; }
```

**Property Value**

Type: `String`
The name.

**Implements**

IRpcMethodParameter.Name

**Reference**

RpcMethodParameter Class
TwinCAT.Ads.TypeSystem Namespace
6.7.18.1.6   RpcMethodParameter.ParameterFlags Property

Gets the parameter flags.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public MethodParamFlags ParameterFlags { get; }
```

**Property Value**

Type: MethodParamFlags

The parameter flags.

**Implements**

IRpcMethodParameter.ParameterFlags

**Reference**

RpcMethodParameter Class

TwinCAT.Ads.TypeSystem Namespace

6.7.18.1.7   RpcMethodParameter.Size Property

Gets the size of the RpcMethodParameter.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public int Size { get; }
```

**Property Value**

Type: Int32

The size.

**Implements**

IRpcMethodParameter.Size

**Reference**

RpcMethodParameter Class

TwinCAT.Ads.TypeSystem Namespace
6.7.18.1.8 RpcMethodParameter.TypeGuid Property

Gets the Unique identifier of the parameters data type.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Guid TypeGuid { get; }
```

**Property Value**

Type: `Guid`
The type unique identifier.

**Reference**

RpcMethodParameter Class

TwinCAT.Ads.TypeSystem Namespace

6.7.18.1.9 RpcMethodParameter.TypeName Property

Gets the Data type of the Parameter

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public string TypeName { get; }
```

**Property Value**

Type: `String`
The type.

**Implements**

IRpcMethodParameter.TypeName

**Reference**

RpcMethodParameter Class

TwinCAT.Ads.TypeSystem Namespace

6.7.18.2 RpcMethodParameter Methods

The RpcMethodParameter type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Reference

- **RpcMethodParameter Class** [► 1440]
- **TwinCAT.Ads.TypeSystem Namespace** [► 1328]

## 6.7.19 RpcStructType Class

StructType which is callable by RPC Methods.

### Inheritance Hierarchy

- System.Object
  - TwinCAT.Ads.TypeSystem.DataType [► 1349]
    - TwinCAT.Ads.TypeSystem.StructType [► 1457]
      - TwinCAT.Ads.TypeSystem.RpcStructType

### Syntax

**C#**

```csharp
public sealed class RpcStructType : StructType, IRpcCallableType
```

The RpcStructType type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllMembers</td>
<td>Gets all members (down the derivation hierarchy) (Inherited from StructType [► 1457].)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType [► 1986] (Inherited from DataType [► 1349].)</td>
</tr>
<tr>
<td>BaseType</td>
<td>Gets the structs Base Type (Null if not derived). (Inherited from StructType [► 1457].)</td>
</tr>
<tr>
<td>BaseTypeName</td>
<td>Gets the Name of the Base class (if derived) (Inherited from StructType [► 1457].)</td>
</tr>
</tbody>
</table>
#### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitSize [1353]</td>
<td>Gets the size of the DataType [1349] in bits. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ByteSize [1353]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Category [1354]</td>
<td>Gets the Data Type category (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Comment [1354]</td>
<td>Gets the comment. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>FullName [1355]</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>HasRpcMethods [1450]</td>
<td>Gets a value indicating whether this instance has RPC Methods. (Overrrides IRpcCallableType.HasRpcMethods [1461].)</td>
</tr>
<tr>
<td>Id [1355]</td>
<td>Gets the ID of the DataType (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsBitType [1356]</td>
<td>Gets a value indicating whether this IDataType [1986] is a bit mapping Type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsByteAligned [1356]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsContainer [1357]</td>
<td>Gets a value indicating whether this IDataType [1986] is a container type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsDerived [1462]</td>
<td>Gets a value indicating whether this instance is derived. (Inherited from StructType [1457].)</td>
</tr>
<tr>
<td>IsPointer [1357]</td>
<td>Gets a value indicating whether this IDataType [1986] is a pointer type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsPrimitive [1358]</td>
<td>Gets a value indicating whether this IDataType [1986] is primitive (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsReference [1359]</td>
<td>Gets a value indicating whether this IDataType [1986] is a reference type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ManagedType [1359]</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Members [1462]</td>
<td>Gets a read only collection of the Members [2065] of the IStructType [2162]. (Inherited from StructType [1457].)</td>
</tr>
<tr>
<td>Name [1360]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Namespace [1360]</td>
<td>Gets the namespace string within the IDataType [1986] exists. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>RpcMethods [1451]</td>
<td>Gets the Method descriptions for the IRpcCallableType [2121]</td>
</tr>
<tr>
<td>Size [1361]</td>
<td>Gets the Size of the DataType [1349] in Bytes or bits. (Inherited from DataType [1349].)</td>
</tr>
</tbody>
</table>

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ToString</code> [1363]</td>
<td>Returns a <code>String</code> that represents this instance. (Inherited from <code>DataType</code> [1349].)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.19.1 RpcStructType Properties

The `RpcStructType` [1447] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>AllMembers</code> [1460]</td>
<td>Gets all members (down the derivation hierarchy) (Inherited from <code>StructType</code> [1457].)</td>
</tr>
<tr>
<td><code>Attributes</code> [1352]</td>
<td>Gets the attributes of the <code>IDataType</code> [1986] (Inherited from <code>DataType</code> [1349].)</td>
</tr>
<tr>
<td><code>BaseType</code> [1460]</td>
<td>Gets the structs Base Type (Null if not derived). (Inherited from <code>StructType</code> [1457].)</td>
</tr>
<tr>
<td><code>BaseTypeName</code> [1461]</td>
<td>Gets the the Name of the Base class (if derived) (Inherited from <code>StructType</code> [1457].)</td>
</tr>
<tr>
<td><code>BitSize</code> [1353]</td>
<td>Gets the size of the <code>DataType</code> [1349] in bits. (Inherited from <code>DataType</code> [1349].)</td>
</tr>
<tr>
<td><code>ByteSize</code> [1353]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from <code>DataType</code> [1349].)</td>
</tr>
<tr>
<td><code>Category</code> [1354]</td>
<td>Gets the Data Type category (Inherited from <code>DataType</code> [1349].)</td>
</tr>
<tr>
<td><code>Comment</code> [1354]</td>
<td>Gets the comment. (Inherited from <code>DataType</code> [1349].)</td>
</tr>
<tr>
<td><code>FullName</code> [1355]</td>
<td>Gets the full name of the <code>IDataType</code> [1986] (Namespace + Name) (Inherited from <code>DataType</code> [1349].)</td>
</tr>
<tr>
<td><code>HasRpcMethods</code> [1450]</td>
<td>Gets a value indicating whether this instance has RPC Methods. (Overrides <code>StructType.HasRpcMethods</code> [1461].)</td>
</tr>
<tr>
<td><code>Id</code> [1355]</td>
<td>Gets the ID of the <code>DataType</code> (Inherited from <code>DataType</code> [1349].)</td>
</tr>
<tr>
<td><code>IsBitType</code> [1356]</td>
<td>Gets a value indicating whether this <code>IDataType</code> [1986] is a bit mapping Type (Inherited from <code>DataType</code> [1349].)</td>
</tr>
<tr>
<td><code>IsByteAligned</code> [1356]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <code>DataType</code> [1349].)</td>
</tr>
<tr>
<td><code>IsContainer</code> [1357]</td>
<td>Gets a value indicating whether this <code>IDataType</code> [1986] is a container type (Inherited from <code>DataType</code> [1349].)</td>
</tr>
<tr>
<td><code>IsDerived</code> [1462]</td>
<td>Gets a value indicating whether this instance is derived. (Inherited from <code>StructType</code> [1457].)</td>
</tr>
<tr>
<td><code>IsPointer</code> [1357]</td>
<td>Gets a value indicating whether this <code>IDataType</code> [1986] is a pointer type (Inherited from <code>DataType</code> [1349].)</td>
</tr>
<tr>
<td><code>IsPrimitive</code> [1358]</td>
<td>Gets a value indicating whether this <code>IDataType</code> [1986] is primitive (Inherited from <code>DataType</code> [1349].)</td>
</tr>
<tr>
<td><code>IsReference</code> [1359]</td>
<td>Gets a value indicating whether this <code>IDataType</code> [1986] is a reference type (Inherited from <code>DataType</code> [1349].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from DataType.)</td>
</tr>
<tr>
<td>Members</td>
<td>Gets a read only collection of the Members of the IStructType. (Inherited from StructType.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from DataType.)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the IDataType exists. (Inherited from DataType.)</td>
</tr>
<tr>
<td>RpcMethods</td>
<td>Gets the Method descriptions for the IRpcCallableType.</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the Size of the DataType in Bytes or bits. (Inherited from DataType.)</td>
</tr>
</tbody>
</table>

### Reference

RpcStructType Class

TwinCAT.Ads.TypeSystem Namespace

### 6.7.19.1.1 RpcStructType.HasRpcMethods Property

Gets a value indicating whether this instance has RPC Methods.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public override bool HasRpcMethods { get; }
```

**Property Value**

Type: Boolean

ture if this instance has methods; otherwise, false.

**Implements**

IStructType.HasRpcMethods

**Remarks**

The DataType (Structure) must be marked with the PlcAttribute ‘TcRpcEnable’ to enable RpcMethods, otherwise RpcMethods are not passed through to the ADS symbolic information.

**Reference**

RpcStructType Class

TwinCAT.Ads.TypeSystem Namespace
6.7.19.1.2  RpcStructType.RpcMethods Property

Gets the Method descriptions for the IRpcCallableType

Namespace:  TwinCAT.Ads.TypeSystem
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdadca3e72bc0ea15da1c14

Syntax

C#
public IRpcMethodCollection RpcMethods { get; }

Property Value

Type:  IRpcMethodCollection
The methods.

Implements

IRpcCallableType.RpcMethods

Remarks

The DataType (Structure) must be marked with the PlcAttribute 'TcRpcEnable' to enable RpcMethods, otherwise RpcMethods are not passed through to the ADS symbolic information.

Reference

RpcStructType Class [1447]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.19.2  RpcStructType Methods

The RpcStructType type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DataType.)</td>
</tr>
</tbody>
</table>

Reference

RpcStructType Class [1447]
TwinCAT.Ads.TypeSystem Namespace [1328]
6.7.20 StringType Class

String DataType

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType
  TwinCAT.Ads.TypeSystem.StringType

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll)
Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public sealed class StringType : DataType,
    IStringType, IDataType, IBitSize
```

The StringType type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1352]</td>
<td>Gets the attributes of the [IDataType [1986]] (Inherited from [DataType [1349]].)</td>
</tr>
<tr>
<td>BitSize [1353]</td>
<td>Gets the size of the [DataType [1349]] in bits. (Inherited from [DataType [1349]].)</td>
</tr>
<tr>
<td>ByteSize [1353]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from [DataType [1349]].)</td>
</tr>
<tr>
<td>Category [1354]</td>
<td>Gets the Data Type category (Inherited from [DataType [1349]].)</td>
</tr>
<tr>
<td>Comment [1354]</td>
<td>Gets the comment. (Inherited from [DataType [1349]].)</td>
</tr>
<tr>
<td>Encoding [1454]</td>
<td>Gets the encoding of the String (Encoding.Default (Ansi Codepage, STRING) or Encoding.UNICODE (WSTRING))</td>
</tr>
<tr>
<td>FullName [1355]</td>
<td>Gets the full name of the [IDataType [1986]] (Namespace + Name) (Inherited from [DataType [1349]].)</td>
</tr>
<tr>
<td>Id [1355]</td>
<td>Gets the ID of the DataType (Inherited from [DataType [1349]].)</td>
</tr>
<tr>
<td>IsBitType [1356]</td>
<td>Gets a value indicating whether this [IDataType [1986]] is a bit mapping Type (Inherited from [DataType [1349]].)</td>
</tr>
<tr>
<td>IsByteAligned [1356]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from [DataType [1349]].)</td>
</tr>
<tr>
<td>IsContainer [1357]</td>
<td>Gets a value indicating whether this [IDataType [1986]] is a container type (Inherited from [DataType [1349]].)</td>
</tr>
<tr>
<td>IsFixedSizeLength [1455]</td>
<td>Gets a value indicating whether the string is of fixed length.</td>
</tr>
<tr>
<td>IsPointer [1357]</td>
<td>Gets a value indicating whether this [IDataType [1986]] is a pointer type (Inherited from [DataType [1349]].)</td>
</tr>
<tr>
<td>IsPrimitive [1358]</td>
<td>Gets a value indicating whether this [IDataType [1986]] is primitive (Inherited from [DataType [1349]].)</td>
</tr>
<tr>
<td>IsReference [1359]</td>
<td>Gets a value indicating whether this [IDataType [1986]] is a reference type (Inherited from [DataType [1349]].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### StringType Properties

The `StringType` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the <code>IDataType</code> (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>DataType</code> in bits. (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the <code>Type/Instance</code> in Bytes. (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category. (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>Encoding</td>
<td>Gets the encoding of the String. (Default (Ansi Codepage, STRING) or Encoding.UNICODE (WSTRING))</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the <code>IDataType</code> (Namespace + Name). (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the <code>DataType</code>. (Inherited from <code>DataType</code>.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a bit mapping Type. (Inherited from <code>DataType</code>.)</td>
</tr>
</tbody>
</table>

---

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents this instance. (Overrides <code>DataType.ToString</code>.)</td>
</tr>
</tbody>
</table>

---

**Reference**

TwinCAT.Ads.TypeSystem Namespace [» 1328]

### 6.7.20.1 StringType Properties

The `StringType` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this [DataType] is a container type (Inherited from DataType)</td>
</tr>
<tr>
<td>IsFixedLength</td>
<td>Gets a value indicating whether the string is of fixed length.</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this [DataType] is a pointer type (Inherited from DataType)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this [DataType] is primitive (Inherited from DataType)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this [DataType] is a reference type (Inherited from DataType)</td>
</tr>
<tr>
<td>Length</td>
<td>Gets the number of characters within the string.</td>
</tr>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from DataType)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from DataType)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the [DataType] exists. (Inherited from DataType)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the Size of the [DataType] in Bytes or bits. (Inherited from DataType)</td>
</tr>
</tbody>
</table>

### Reference

**StringType Class** [1452]

TwinCAT.Ads.TypeSystem Namespace [1328]

#### 6.7.20.1.1 StringType.Encoding Property

Gets the encoding of the String (Encoding.Default (Ansi Codepage, STRING) or Encoding.UNICODE (WSTRING))

**Namespace:** TwinCAT.Ads.TypeSystem [1328]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Encoding Encoding { get; }
```

**Property Value**

Type: Encoding  
The encoding.

**Implements**

[2157]
6.7.20.1.2 StringType.IsFixedLength Property

Gets a value indicating whether the string is of fixed length.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool IsFixedLength { get; }
```

**Property Value**

Type: Boolean

true if this instance is fixed length; otherwise, false.

**Implements**

IStringType.IsFixedLength

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

**StringType Class [1452]**

**TwinCAT.Ads.TypeSystem Namespace [1328]**

6.7.20.1.3 StringType.Length Property

Gets the number of characters within the string.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public int Length { get; }
```
Property Value

Type: Int32
The length.

Implements

IStringType.Length [2158]

Reference

StringType Class [1452]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.20.2 StringType Methods

The StringType [1452] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Overrides DataType.ToString.)</td>
</tr>
</tbody>
</table>

Reference

StringType Class [1452]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.20.2.1 StringType.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public override string ToString()

Return Value

Type: String
A String that represents this instance.
6.7.21 StructType Class

Represents a struct type

Inheritance Hierarchy

System.Object
   TwinCAT.Ads.TypeSystem.DataType
      TwinCAT.Ads.TypeSystem.StructType
         TwinCAT.Ads.TypeSystem.RpcStructType

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b609f934fddca3e72bc0ea15da1c14

Syntax

C#
public class StructType : DataType, IStructType, IDataType, IBitSize

The StructType type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllMembers [1460]</td>
<td>Gets all members (down the derivation hierarchy)</td>
</tr>
<tr>
<td>Attributes [1352]</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>BaseType [1460]</td>
<td>Gets the structs Base Type (Null if not derived).</td>
</tr>
<tr>
<td>BaseTypeName [1461]</td>
<td>Gets the the Name of the Base class (if derived)</td>
</tr>
<tr>
<td>BitSize [1353]</td>
<td>Gets the size of the DataType [1349] in bits. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ByteSize [1353]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Category [1354]</td>
<td>Gets the Data Type category (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Comment [1354]</td>
<td>Gets the comment. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>FullName [1355]</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>HasRpcMethods [1461]</td>
<td>Gets a value indicating whether this StructType has RPC Methods.</td>
</tr>
<tr>
<td>Id [1355]</td>
<td>Gets the ID of the DataType (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsBitType [1356]</td>
<td>Gets a value indicating whether this IDataType [1986] is a bit mapping Type (Inherited from DataType [1349].)</td>
</tr>
</tbody>
</table>
### StructType Properties

The `StructType` type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllMembers</td>
<td>Gets all members (down the derivation hierarchy)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType (Inherited from DataType)</td>
</tr>
<tr>
<td>BaseType</td>
<td>Gets the structs Base Type (Null if not derived).</td>
</tr>
<tr>
<td>BaseTypeName</td>
<td>Gets the the Name of the Base class (if derived)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the DataType in bits. (Inherited from DataType)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from DataType)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from DataType)</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the IDataType (Namespace + Name) (Inherited from</td>
</tr>
<tr>
<td>HasRpcMethods</td>
<td>Gets a value indicating whether this StructType has RPC Methods.</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the DataType (Inherited from DataType)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this IDataType is a bit mapping Type</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this IDataType is a container type</td>
</tr>
<tr>
<td>IsDerived</td>
<td>Gets a value indicating whether this instance is derived.</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this IDataType is a pointer type</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IDataType is primitive</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this IDataType is a reference type</td>
</tr>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from DataType)</td>
</tr>
<tr>
<td>Members</td>
<td>Gets a read only collection of the Members of the IStructType</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from DataType)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the IDataType exists. (Inherited from</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the Size of the DataType in Bytes or bits. (Inherited from DataType)</td>
</tr>
</tbody>
</table>
6.7.21.1.1 StructType.AllMembers Property

Gets all members (down the derivation hierarchy)

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public IMemberCollection AllMembers { get; }
```

Property Value

Type: IMemberCollection

All members.

Implements

I structType.AllMembers

Reference

StructType Class

TwinCAT.Ads.TypeSystem Namespace

6.7.21.1.2 StructType.BaseType Property

Gets the structs Base Type (Null if not derived).

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public IDataType BaseType { get; }
```

Property Value

Type: IDataType

Implements

I structType.BaseType

Reference

StructType Class

TwinCAT.Ads.TypeSystem Namespace
6.7.21.1.3 **StructType.BaseTypeName Property**

Gets the the Name of the Base class (if derived)

**Namespace:**  TwinCAT.Ads.TypeSystem

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
public string BaseTypeName { get; }
```

**Property Value**

Type:  **String**

Empty if not derived.

**Implements**

**IStructType.BaseTypeName**

**Reference**

**StructType Class**

**TwinCAT.Ads.TypeSystem Namespace**

6.7.21.1.4 **StructType.HasRpcMethods Property**

Gets a value indicating whether this **StructType** has RPC Methods.

**Namespace:**  TwinCAT.Ads.TypeSystem

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
public virtual bool HasRpcMethods { get; }
```

**Property Value**

Type:  **Boolean**

true if this type has methods; otherwise, false.

**Implements**

**IStructType.HasRpcMethods**
Remarks
The DataType (Structure) must be marked with the PlcAttribute 'TcRpcEnable' to enable RpcMethods, otherwise RpcMethods are not passed through to the ADS symbolic information.

Reference
StructType Class [1457]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.21.1.5 StructType.IsDerived Property
Gets a value indicating whether this instance is derived.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool IsDerived { get; }  

Property Value
Type: Boolean
true if this instance is derived; otherwise, false.

Reference
StructType Class [1457]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.21.1.6 StructType.Members Property
Gets a read only collection of the Members [2065] of the IStructType [2162].

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public IMemberCollection Members { get; }  

Property Value
Type: IMemberCollection [2068]
The members as read only collection.

Implements
IStructType.Members [2166]
Remarks

If the IStructType is derived, only the extended members are returned. To get all supported members down the inheritance chain, use the AllMembers property.

Reference

StructType Class

TwinCAT.Ads.TypeSystem Namespace

6.7.21.2 StructType Methods

The StructType type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DataType)</td>
</tr>
</tbody>
</table>

Reference

StructType Class

TwinCAT.Ads.TypeSystem Namespace

6.7.22 SubRangeType.T Class

Represents a SubRangeType

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType
  TwinCAT.Ads.TypeSystem.SubRangeType.T.

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public sealed class SubRangeType<T> : DataType,
    ISubRangeType<T>, ISubRangeType, IDataType, IBitSize
where T : struct, new()
```
Type Parameters

T

The SubRangeType.T. type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType (Inherited from DataType.</td>
</tr>
<tr>
<td>BaseType</td>
<td>Gets the base type.</td>
</tr>
<tr>
<td>BaseTypeName</td>
<td>Gets the name of the base type.</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the DataType in bits. (Inherited from DataType.</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from DataType.</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from DataType.</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the IDataType (Namespace + Name) (Inherited from</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the Data Type (Inherited from DataType.</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this IDataType is a bit mapping Type</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this IDataType is a container type</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this IDataType is a pointer type</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IDataType is primitive</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this IDataType is a reference type</td>
</tr>
<tr>
<td>LowerBound</td>
<td>Gets the lower bound.</td>
</tr>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the IDataType exists. (Inherited from</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the Size of the DataType in Bytes or bits. (Inherited from</td>
</tr>
<tr>
<td>UpperBound</td>
<td>Gets the upper bound.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DataType.)</td>
</tr>
</tbody>
</table>

## Reference

TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.22.1 SubRangeType.T. Properties

The SubRangeType.T.[1463] generic type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>BaseType</td>
<td>Gets the base type.</td>
</tr>
<tr>
<td>BaseTypeName</td>
<td>Gets the name of the base type.</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the DataType [1349] in bits. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the DataType (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this IDataType [1986] is a bit mapping Type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this IDataType [1986] is a container type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this IDataType [1986] is a pointer type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IDataType [1986] is primitive (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this IDataType [1986] is a reference type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LowerBound [1467]</td>
<td>Gets the lower bound.</td>
</tr>
<tr>
<td>ManagedType [1467]</td>
<td>Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType [1359].)</td>
</tr>
<tr>
<td>Name [1360]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Namespace [1360]</td>
<td>Gets the namespace string within the IDataType [1986] exists. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Size [1361]</td>
<td>Gets the Size of the DataType [1349] in Bytes or bits. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>UpperBound [1468]</td>
<td>Gets the upper bound.</td>
</tr>
</tbody>
</table>

Reference

SubRangeType.T. Class [1463]

TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.22.1.1 SubRangeType.T..BaseType Property

Gets the the base type.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public IDataType BaseType { get; }
```

**Property Value**

Type: IDataType [1986]

The type of the referenced.

**Implements**

ISubRangeType.BaseType [2172]

Reference

SubRangeType.T. Class [1463]

TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.22.1.2 SubRangeType.T..BaseTypeName Property

Gets the name of the base type.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#
public string BaseTypeName { get; }

Property Value

Type: String
The name of the base type.

Implements

ISubRangeType.BaseTypeName [2173]

Reference

SubRangeType.T. Class [1463]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.22.1.3 SubRangeType.T..LowerBound Property

Gets the lower bound.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public T LowerBound { get; }

Property Value

Type: T [1463]
The lower bound.

Implements

ISubRangeType.T..LowerBound [2176]

Reference

SubRangeType.T. Class [1463]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.22.1.4 SubRangeType.T..ManagedType Property

Gets the corresponding .NET Type if attached.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
### SubRangeType.T..UpperBound Property

Gets the upper bound.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public T UpperBound { get; }
```

**Property Value**

Type: `T`

The upper bound.

**Implements**

`ISubRangeType.T..UpperBound`

**Reference**

SubRangeType.T. Class

TwinCAT.Ads.TypeSystem Namespace

### SubRangeType.T. Methods

The `SubRangeType.T.` generic type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

| equals         | Determines whether the specified object is equal to the current object.     |
| GetHashCode    | Serves as the default hash function. (Inherited from `Object`.)             |
### Symbol Class

Symbol class

#### Inheritance Hierarchy

```
System.Object
  TwinCAT.Ads.TypeSystem.Instance [1388]
  TwinCAT.Ads.TypeSystem.Symbol
```

#### Syntax

**C#**

```csharp
public class Symbol : Instance, IValueSymbol,
                    IValueRawSymbol, IHierarchicalSymbol, ISymbol, IAttributedInstance, IInstance,
                    IBitSize, IValueAnySymbol, IValueAccessorProvider, ISymbolFactoryServicesProvider, IAdsSymbol,
                    IProcessImageAddress, IContextMaskProvider
```

The Symbol type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessRights</td>
<td>Gets the access rights.</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of this Instance [1388] in bits. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the the DataTypeCategory [1649] of the Instance. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the connection that produces values for this IValueSymbol [2254]</td>
</tr>
<tr>
<td>ContextMask</td>
<td>Gets the context mask of this instance. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this instance has a value. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>ImageBaseAddress</td>
<td>Gets the AmsAddress [648] of the Process Image</td>
</tr>
<tr>
<td>IndexGroup</td>
<td>Gets the index group of the Symbol</td>
</tr>
<tr>
<td>IndexOffset</td>
<td>Gets the index offset of the Symbol</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)) (Overrides Instance.InstancePath [1396].)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether the Symbols datatype is a Container type.</td>
</tr>
<tr>
<td>IsDereferencedPointer</td>
<td>Gets or sets a value indicating whether an ancestor is a dereferenced Pointer.</td>
</tr>
<tr>
<td>IsDereferencedReference</td>
<td>Gets or sets a value indicating whether an ancestor is a dereferenced Reference.</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Indicates that this instance is persistent. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this instance is reference. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is primitive.</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive.</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this instance is reference. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this Instance [2052] is static. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsTcComInterfacePointer</td>
<td>Indicates that this instance is a TcComInterfacePointer. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsTypeGuid</td>
<td>Indicates that this instance has set TypeGuid flag. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>IsVirtual</td>
<td>Gets a value indicating whether this instance is virtual.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace name. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>NotificationSettings</td>
<td>Gets or sets the notification settings.</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size [1401]</td>
<td>Gets the size of the IDataType [1986] in bytes or Bits dependant on IsBitType [1396] (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>SubSymbolCount [1484]</td>
<td>Gets the number of SubSymbols</td>
</tr>
<tr>
<td>SubSymbols [1484]</td>
<td>Gets the SubSymbols of the ISymbol [2176]</td>
</tr>
<tr>
<td>TypeName [1402]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052]. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>ValueEncoding [1485]</td>
<td>Gets the value encoding.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnsureRights [1488]</td>
<td>Ensures that the AccessRights are matched.</td>
</tr>
<tr>
<td>Equals [1489]</td>
<td>Equals (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>OnGetSize [1403]</td>
<td>Handler function getting the size of the Instance [1388] (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td>OnReadRawValue [1490]</td>
<td>Handler function for reading the raw value</td>
</tr>
<tr>
<td>OnReadRawValueAsync [1490]</td>
<td>Handler function for reading the raw value</td>
</tr>
<tr>
<td>OnReadValue [1491]</td>
<td>Handler function for reading the dynamic value.</td>
</tr>
<tr>
<td>OnReadValueAsync [1492]</td>
<td>Handler function for reading the dynamic value.</td>
</tr>
<tr>
<td>OnSetInstanceName [1492]</td>
<td>Sets a new InstanceName InstancePath (Overrides Instance.OnSetInstanceName(String) [1404].)</td>
</tr>
<tr>
<td>OnTryReadValue [1493]</td>
<td>Handler function for reading the dynamic value.</td>
</tr>
<tr>
<td>OnTryWriteValue [1493]</td>
<td>Handler function for writing the dynamic value.</td>
</tr>
<tr>
<td>OnWriteRawValue [1494]</td>
<td>Handler function for writing the dynamic value.</td>
</tr>
<tr>
<td>OnWriteRawValueAsync [1494]</td>
<td>Handler function for writing the dynamic value.</td>
</tr>
<tr>
<td>OnWriteValue [1495]</td>
<td>Handler function for writing the dynamic value.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>OnWriteValueAsync</td>
<td>Handler function for writing the dynamic value</td>
</tr>
<tr>
<td>ReadAnyValue(Type)</td>
<td>Reads the value of this Value into a new created instance of the managed type</td>
</tr>
<tr>
<td>ReadAnyValue(Type, Int32)</td>
<td>Reads the value of this Value into a new created instance of the managed type</td>
</tr>
<tr>
<td>ReadAnyValueAsync</td>
<td>Reads the (AnyType) value asynchronously.</td>
</tr>
<tr>
<td>ReadRawValue</td>
<td>Reads the raw value of the IValueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the raw value of the IValueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>ReadRawValueAsync</td>
<td>Reads the raw value of the IValueSymbol asynchronously.</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Reads the Value of the IValueSymbol</td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the Value of the IValueSymbol</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the IValueSymbol asynchronously.</td>
</tr>
<tr>
<td>SetAttributes</td>
<td>Sets the type attributes (Inherited from Instance.)</td>
</tr>
<tr>
<td>SetContextMask</td>
<td>Sets the context mask. (Inherited from Instance.)</td>
</tr>
<tr>
<td>SetParent</td>
<td>Sets the parent symbol.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Overrides Instance.ToString.)</td>
</tr>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol</td>
</tr>
<tr>
<td>TryWriteValue</td>
<td>Writes the Value of the IValueSymbol</td>
</tr>
<tr>
<td>UpdateAnyValue(Object)</td>
<td>Reads the value of this Value into the specified managed value.</td>
</tr>
<tr>
<td>UpdateAnyValue(Object, Int32)</td>
<td>Reads the value of this Value into the specified managed value.</td>
</tr>
<tr>
<td>WriteRawValue(Byte)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteRawValue(Byte, Int32)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteRawValueAsync</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### Name | Description
--- | ---
| WriteValue(Object) | Writes the Value of the IValueSymbol [2254] |
| WriteValue(Object, Int32) | Writes the Value of the IValueSymbol [2254] |
| WriteValueAsync | Writes the Value of the IValueSymbol [2254] |

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol [2254] has changed.</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol [2254] has changed.</td>
</tr>
</tbody>
</table>

### Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equality</td>
<td>Operator==</td>
</tr>
<tr>
<td>Inequality</td>
<td>Implements the != operator.</td>
</tr>
</tbody>
</table>

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>syncObject</td>
<td>Synchronization object</td>
</tr>
</tbody>
</table>

### Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollValuesAnnotate dojo(Observable.Unit.)</td>
<td>Overloaded. Polls the values as ValueChangedEventArgs [2439] sequence annotated value on trigger sequence (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td>PollValuesAnnotate d(TimeSpan)</td>
<td>Overloaded. Polls the values as ValueChangedEventArgs [2439] sequence with a specified period time. (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td>WhenValueChanged</td>
<td>Gets an observable sequence when the value of the IValueSymbol [2254] has changed. (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td>WriteValues(IObservable.Observable.Object.)</td>
<td>Overloaded. Subscribes the IValueSymbol [2254] to an observable sequence of values and writes them to the IValueSymbol [2254]. (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### TwinCAT.Ads.TypeSystem Namespace

TwinCAT.Ads.TypeSystem.Instance

TwinCAT.TypeSystem.IValueSymbol

TwinCAT.TypeSystem.IValueAnySymbol

TwinCAT.TypeSystem.IValueAccessorProvider

TwinCAT.TypeSystem.ISymbolFactoryServicesProvider

TwinCAT.TypeSystem.IHierarchicalSymbol

ISymbolValueChangeNotify

TwinCAT.Ads.TypeSystem.IContextMaskProvider

IInstanceInternal

ISymbolInternal

**6.7.23.1 Symbol Properties**

The Symbol type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessRights</td>
<td>Gets the access rights.</td>
</tr>
</tbody>
</table>

**Remarks**

A Symbol is a (named) memory object within the Process Image with a fixed address indicated by Index Group and Index Offset. Symbols can optionally be addressed by instance path and are bound to a specific DataType.

**Reference**

TwinCAT.Ads.TypeSystem Namespace

TwinCAT.Ads.TypeSystem.Instance

TwinCAT.TypeSystem.IValueSymbol

TwinCAT.TypeSystem.IValueAnySymbol

TwinCAT.TypeSystem.IValueAccessorProvider

TwinCAT.TypeSystem.ISymbolFactoryServicesProvider

TwinCAT.TypeSystem.IHierarchicalSymbol

ISymbolValueChangeNotify

TwinCAT.Ads.TypeSystem.IContextMaskProvider

IInstanceInternal

ISymbolInternal

TwinCAT.Ads.TypeSystem.IAdsSymbol
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from Instance.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of this Instance in bits. (Inherited from Instance.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from Instance.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the the DataTypeCategory of the Instance. (Inherited from Instance.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from Instance.)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the connection that produces values for this IValueSymbol. (Inherited from Instance.)</td>
</tr>
<tr>
<td>ContextMask</td>
<td>Gets the context mask of this instance. (Inherited from Instance.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType of the IInstance. (Inherited from Instance.)</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this instance has a value. (Inherited from Instance.)</td>
</tr>
<tr>
<td>ImageBaseAddress</td>
<td>Gets the AmsAddress of the Process Image</td>
</tr>
<tr>
<td>IndexGroup</td>
<td>Gets the index group of the Symbol</td>
</tr>
<tr>
<td>IndexOffset</td>
<td>Gets the index offset of the Symbol</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from Instance.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)) (Overrides Instance. InstancePath.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from Instance.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from Instance.)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether the Symbols datatype is a Container type.</td>
</tr>
<tr>
<td>IsDereferencedPointer</td>
<td>Gets or sets a value indicating whether an ancestor is a dereferenced Pointer</td>
</tr>
<tr>
<td>IsDereferencedReference</td>
<td>Gets or sets a value indicating whether an ancestor is a dereferenced Reference</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Indicates that this instance is persistent. (Inherited from Instance.)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this instance is reference. (Inherited from Instance.)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is primitive.</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from Instance.)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this instance is reference. (Inherited from Instance.)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this Instance is static. (Inherited from Instance.)</td>
</tr>
<tr>
<td>IsTcComInterfacePointer</td>
<td>Indicates that this instance is a TcComInterfacePointer. (Inherited from Instance.)</td>
</tr>
<tr>
<td>IsTypeGuid</td>
<td>Indicates that this instance has set TypeGuid flag. (Inherited from Instance.)</td>
</tr>
<tr>
<td>IsVirtual</td>
<td>Gets a value indicating whether this instance is virtual.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace name. (Inherited from Instance.)</td>
</tr>
<tr>
<td>NotificationSettings</td>
<td>Gets or sets the notification settings.</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the IDataType in bytes or Bits dependant on IsBitType (Inherited from Instance.)</td>
</tr>
<tr>
<td>SubSymbolCount</td>
<td>Gets the number of SubSymbols</td>
</tr>
<tr>
<td>SubSymbols</td>
<td>Gets the SubSymbols of the ISymbol</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType that is used for this Instance. (Inherited from Instance.)</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding.</td>
</tr>
</tbody>
</table>

Reference
Symbol Class | TwinCAT.Ads.TypeSystem Namespace

6.7.23.1.1 Symbol.AccessRights Property

Gets the access rights.

Namespace: TwinCAT.Ads.TypeSystem Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public SymbolAccessRights AccessRights { get; }

Property Value

Type: SymbolAccessRights
The access rights.
Implements

IValueSymbol.AccessRights [2259]

Reference

Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.1.2 Symbol.Connection Property

Gets the connection that produces values for this IValueSymbol [2254]

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public IConnection Connection { get; }

Property Value

Type: IConnection [74]
The connection object.

Implements

IValueSymbol.Connection [2260]

Reference

Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.1.3 Symbol.ImageBaseAddress Property

Gets the AmsAddress [648] of the Process Image

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public AmsAddress ImageBaseAddress { get; }

Property Value

Type: AmsAddress [648]
The address.
Implements

IAdsSymbol.ImageBaseAddress [1383]

Reference

Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.1.4 Symbol.IndexGroup Property

Gets the index group of the Symbol

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public uint IndexGroup { get; }
```

Property Value

Type: UInt32
The index group.

Implements

IProcessImageAddress.IndexGroup [2093]

Reference

Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.1.5 Symbol.IndexOffset Property

Gets the index offset of the Symbol

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public uint IndexOffset { get; }
```

Property Value

Type: UInt32
The index offset.
Implements
IProcessImageAddress.IndexOffset [2093]

Reference
Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.1.6 Symbol.InstancePath Property

Gets the relative / absolute access path to the instance (with periods (.)

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public override string InstancePath { get; }

Property Value

Type: String
The instance path.

Implements
IInstance.InstancePath [2055]

Remarks

If this path is relative or absolute depends on the context. IMember [2065] are using relative paths, ISymbol [2176]s are using absolute ones.

Reference
Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.1.7 Symbol.IsContainerType Property

Gets a value indicating whether the Symbols datatype is a Container type.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
## Syntax

### C#

```csharp
public virtual bool IsContainerType { get; }
```

## Property Value

Type: Boolean
true if this instance is container type; otherwise, false.

## Implements

`ISymbol.IsContainerType` [2179]

## Remarks

Container Types are all types that contain SubElements like

- Array [1649]
- Pointer [1649]
- Union [1649]
- Struct [1649]
- Function [1649]
- FunctionBlock [1649]
- Program [1649]

and the Alias [1649] and Reference [1649] types, if they have a container type as base type.

## Reference

- Symbol Class [1469]
- TwinCAT.Ads.TypeSystem Namespace [1328]
- `IDataType.Category` [1988]

### 6.7.23.1.8 Symbol.IsDereferencedPointer Property

Gets or sets a value indicating whether an ancestor is a dereferenced Pointer

#### Namespace

TwinCAT.Ads.TypeSystem [1328]

#### Assembly

TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

```csharp
public bool IsDereferencedPointer { get; }
```

## Property Value

Type: Boolean
true if this instance is ancestor is pointer; otherwise, false.
**6.7.23.1.9 Symbol.IsDereferencedReference Property**

Gets or sets a value indicating whether an ancestor is a dereferenced Reference

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public bool IsDereferencedReference { get; }
```

**Property Value**

Type: Boolean  
true if this instance is ancestor is reference; otherwise, false.

---

**6.7.23.1.10 Symbol.IsPrimitiveType Property**

Gets a value indicating whether this instance is primitive.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public virtual bool IsPrimitiveType { get; }
```

**Property Value**

Type: Boolean  
true if this instance is primitive; otherwise, false.

**Implements**

ISymbol.IsPrimitiveType [2180]
6.7.23.1.11 Symbol.IsRecursive Property

Gets a value indicating whether this instance is recursive.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool IsRecursive { get; }
```

**Property Value**

Type: Boolean
true if this instance is recursive; otherwise, false.

**Implements**

ISymbol.IsRecursive

**Reference**

Symbol Class

TwinCAT.Ads.TypeSystem Namespace

6.7.23.1.12 Symbol.IsVirtual Property

Gets a value indicating whether this instance is virtual.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool IsVirtual { get; }
```

**Property Value**

Type: Boolean
true if this instance is virtual; otherwise, false.

**Implements**

IProcessImageAddress.IsVirtual

**Remarks**

Virtual symbols are are only organizational elements within the Symbols Hierarchy and cannot be accessed seperately by IndexGroup/IndexOffset, Value Read/Writes, notifications or handles.
6.7.23.1.13  Symbol.NotificationSettings Property

Gets or sets the notification settings.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```
C#
public INotificationSettings NotificationSettings { get; set; }
```

**Property Value**

Type: INotificationSettings

The notification settings.

**Implements**

IValueSymbol.NotificationSettings

**Remarks**

The NotificationSettings will be inherited from Parent if the setting is not overwritten. If the Root Symbol also doesn't contain the settings, then the DefaultNotificationSettings will be returned.

**Reference**

Symbol Class  
TwinCAT.Ads.TypeSystem Namespace

6.7.23.1.14  Symbol.Parent Property

Gets the parent Symbol

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```
C#
public ISymbol Parent { get; }
```

**Property Value**

Type: ISymbol

The parent.
6.7.23.1.15 Symbol.SubSymbolCount Property

Gets the number of SubSymbols

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public int SubSymbolCount { get; }
```

**Property Value**

Type: Int32  
The Number of SubSymbols.

**Remarks**

If the SubSymbols collection is not generated yet (WeakReference), then this method is less memory and cpu consuming to use for just determining the the number of child symbols (instead of using SubSymbols.Count)/>

Reference

Symbol Class [1469]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.1.16 Symbol.SubSymbols Property

Gets the SubSymbols of the [Symbol] [2176]

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public ISymbolCollection<ISymbol> SubSymbols { get; }
```

**Property Value**

Type: ISymbolCollection [2185] [Symbol] [2176].
### 6.7.23.1.17 Symbol.ValueEncoding Property

Gets the value encoding.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public virtual Encoding ValueEncoding { get; }
```

#### Property Value

**Type:** Encoding

The value encoding.

**Implements**

IAttributedInstance.ValueEncoding

### 6.7.23.2 Symbol Methods

The **Symbol** type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnsureRights</td>
<td>Ensures that the AccessRights are matched.</td>
</tr>
<tr>
<td>Equals</td>
<td>Equals (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Overrides Object.GetHashCode.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>OnGetSize</td>
<td>Handler function getting the size of the Instance (Inherited from Instance.)</td>
</tr>
<tr>
<td>OnReadRawValue</td>
<td>Handler function for reading the raw value</td>
</tr>
<tr>
<td>OnReadRawValueAsync</td>
<td>Handler function for reading the raw value</td>
</tr>
<tr>
<td>OnReadValue</td>
<td>Handler function for reading the dynamic value.</td>
</tr>
<tr>
<td>OnReadValueAsync</td>
<td>Handler function for reading the dynamic value.</td>
</tr>
<tr>
<td>OnSetInstanceName</td>
<td>Sets a new InstanceName InstancePath (Overides Instance.OnSetInstanceName(String))</td>
</tr>
<tr>
<td>OnTryReadValue</td>
<td>Handler function for reading the dynamic value.</td>
</tr>
<tr>
<td>OnTryWriteValue</td>
<td>Handler function for writing the dynamic value.</td>
</tr>
<tr>
<td>OnWriteRawValue</td>
<td>Handler function for writing the dynamic value.</td>
</tr>
<tr>
<td>OnWriteRawValueAsync</td>
<td>Handler function for writing the dynamic value.</td>
</tr>
<tr>
<td>ReadAnyValue(Type)</td>
<td>Reads the value of this Value into a new created instance of the managed type</td>
</tr>
<tr>
<td>ReadAnyValue(Type, Int32)</td>
<td>Reads the value of this Value into a new created instance of the managed type</td>
</tr>
<tr>
<td>ReadAnyValueAsync</td>
<td>Reads the (AnyType) value asynchronously.</td>
</tr>
<tr>
<td>ReadRawValue</td>
<td>Reads the raw value of the IValueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the raw value of the IValueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>ReadRawValueAsyc</td>
<td>Reads the raw value of the IValueSymbol (Ads Read / Write) asynchronously.</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Reads the Value of the IValueSymbol</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ReadValue(Int32)</strong> [1502]</td>
<td>Reads the Value of the IValueSymbol [2254]</td>
</tr>
<tr>
<td><strong>ReadValueAsync</strong> [1503]</td>
<td>Reads the Value of the IValueSymbol [2254] asynchronously.</td>
</tr>
<tr>
<td><strong>SetAttributes</strong> [1404]</td>
<td>Sets the type attributes (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td><strong>SetContextMask</strong> [1405]</td>
<td>Sets the context mask. (Inherited from Instance [1388].)</td>
</tr>
<tr>
<td><strong>SetParent</strong> [1504]</td>
<td>Sets the parent symbol.</td>
</tr>
<tr>
<td><strong>ToString</strong> [1505]</td>
<td>Returns a String that represents this instance. (Overrids Instance.ToString [1405].)</td>
</tr>
<tr>
<td><strong>TryReadValue</strong> [1505]</td>
<td>Reads the Value of the IValueSymbol [2254]</td>
</tr>
<tr>
<td><strong>TryWriteValue</strong> [1506]</td>
<td>Writes the Value of the IValueSymbol [2254]</td>
</tr>
<tr>
<td><strong>UpdateAnyValue(Object)</strong> [1507]</td>
<td>Reads the value of this Value [2254] into the specified managed value.</td>
</tr>
<tr>
<td><strong>UpdateAnyValue(Object, Int32)</strong> [1508]</td>
<td>Reads the value of this Value [2254] into the specified managed value.</td>
</tr>
<tr>
<td><strong>WriteRawValue(Byte)</strong> [1509]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td><strong>WriteRawValue(Byte, Int32)</strong> [1509]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td><strong>WriteRawValueAsync</strong> [1510]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td><strong>WriteValue(Object)</strong> [1511]</td>
<td>Writes the Value of the IValueSymbol [2254]</td>
</tr>
<tr>
<td><strong>WriteValue(Object, Int32)</strong> [1512]</td>
<td>Writes the Value of the IValueSymbol [2254]</td>
</tr>
<tr>
<td><strong>WriteValueAsync</strong> [1513]</td>
<td>Writes the Value of the IValueSymbol [2254]</td>
</tr>
</tbody>
</table>

### Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollValuesAnnotated(IObservable.Unit.) [1111]</td>
<td>Overloaded. Polls the values as ValueChangedEventArgs [2439] sequence annotated value on trigger sequence (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td>PollValuesAnnotated(TimeSpan) [1112]</td>
<td>Overloaded. Polls the values as ValueChangedEventArgs [2439] sequence with a specified period time. (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WhenValueChanged</td>
<td>Gets an observable sequence when the value of the IValueSymbol has changed. (Defined by ValueSymbolExtensions.)</td>
</tr>
<tr>
<td>WriteValues(IObservable.Object.)</td>
<td>Overloaded. Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol. (Defined by ValueSymbolExtensions.)</td>
</tr>
<tr>
<td>WriteValues(IObservable.Object, Action.Exception.)</td>
<td>Overloaded. Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol. (Defined by ValueSymbolExtensions.)</td>
</tr>
<tr>
<td>WriteValues(IObservable.Object, CancellationToken)</td>
<td>Overloaded. Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol. (Defined by ValueSymbolExtensions.)</td>
</tr>
<tr>
<td>WriteValues(IObservable.Object, Action.Exception, CancellationToken)</td>
<td>Overloaded. Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol. (Defined by ValueSymbolExtensions.)</td>
</tr>
</tbody>
</table>

**Reference**

Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.23.2.1 Symbol.EnsureRights Method

Ensures that the AccessRights are matched.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected void EnsureRights(
    SymbolAccessRights requested
)
```

**Parameters**

requested Type: TwinCAT.TypeSystem.SymbolAccessRights [2396]
The requested rights.
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>InsufficientAccessRightsException</td>
<td>[2076]</td>
</tr>
</tbody>
</table>

### Reference

- **Symbol Class**: [1469]
- **TwinCAT.Ads.TypeSystem Namespace**: [1328]

### 6.7.23.2.2 Symbol.Equals Method

**Equals**

**Namespace**: TwinCAT.Ads.TypeSystem [1328]

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public override bool Equals(
    Object obj
)
```

**Parameters**

- **obj**
  - **Type**: System.Object
  - The object to compare with the current object.

**Return Value**

- **Type**: Boolean
  - true if the specified Object is equal to this instance; otherwise, false.

### Reference

- **Symbol Class**: [1469]
- **TwinCAT.Ads.TypeSystem Namespace**: [1328]

### 6.7.23.2.3 Symbol.GetHashCode Method

**Gets the HashCode of the Address**

**Namespace**: TwinCAT.Ads.TypeSystem [1328]

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public override int GetHashCode()
```
Return Value
Type: Int32
A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

Reference
Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.2.4 Symbol.OnReadRawValue Method
Handler function for reading the raw value
Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
protected virtual byte[] OnReadRawValue(
    int timeout
)

Parameters
timeout Type: System.Int32

Return Value
Type: Byte[]. System.Byte[].

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsErrorException [583]</td>
<td></td>
</tr>
<tr>
<td>AdsErrorException [583]</td>
<td></td>
</tr>
</tbody>
</table>

Reference
Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.2.5 Symbol.OnReadRawValueAsync Method
Handler function for reading the raw value
Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
protected virtual Task<ResultReadRawAccess> OnReadRawValueAsync(
    CancellationToken cancel
)
```

Parameters

cancel Type: System.Threading.CancellationToken

Return Value

Type: Task<ResultReadRawAccess> [2564].

System.Byte[].

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdsErrorException [583]</td>
<td></td>
</tr>
<tr>
<td>AdsErrorException [583]</td>
<td></td>
</tr>
</tbody>
</table>

Reference

Symbol Class [1469]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.2.6 Symbol.OnReadValue Method

Handler function for reading the dynamic value.

Namespace: TwinCAT.Ads.TypeSystem [1328]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected virtual Object OnReadValue(
    int timeout
)
```

Parameters

timeout Type: System.Int32

Return Value

Type: Object

The Value

Reference

Symbol Class [1469]
6.7.23.2.7 Symbol.OnReadValueAsync Method

Handler function for reading the dynamic value.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# protected virtual Task<ResultReadValueAccess> OnReadValueAsync( CancellationToken cancel )

**Parameters**

cancel Type: System.Threading.CancellationToken

**Return Value**

Type: Task.ResultReadValueAccess

The Value

**Reference**

Symbol Class
TwinCAT.Ads.TypeSystem Namespace

6.7.23.2.8 Symbol.OnSetInstanceName Method

Sets a new InstanceName InstancePath

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# protected override void OnSetInstanceName( string instanceName )

**Parameters**

instanceName Type: System.String

Instance name.

**Reference**

Symbol Class
TwinCAT.Ads.TypeSystem Namespace
6.7.23.2.9 Symbol.OnTryReadValue Method

Handler function for reading the dynamic value.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
protected virtual int OnTryReadValue(
    int timeout,
    out Object value
)
```

**Parameters**

- **timeout**
  - Type: System.Int32

- **value**
  - Type: System.Object

**Return Value**

- Type: Int32
  - The Value

**Reference**

- Symbol Class [1469]
- TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.2.10 Symbol.OnTryWriteValue Method

Handler function for writing the dynamic value

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
protected virtual int OnTryWriteValue(
    Object value,
    int timeout
)
```

**Parameters**

- **value**
  - Type: System.Object
  - The value.

- **timeout**
  - Type: System.Int32
  - The timeout.
Return Value

Type: Int32
System.Int32.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotAccessVirtualSymbolException [1641]</td>
<td></td>
</tr>
<tr>
<td>CannotAccessVirtualSymbolException [1641]</td>
<td></td>
</tr>
</tbody>
</table>

Reference

Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.2.11 Symbol.OnWriteRawValue Method

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

protected virtual void OnWriteRawValue(
    ReadOnlyMemory value,
    void timeout
)

Parameters

value | Type: ReadOnlyMemory

timeout | Type: System.Void

Reference

Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.2.12 Symbol.OnWriteRawValueAsync Method

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
### Syntax

#### C#

```csharp
protected virtual Task<ResultWriteAccess> OnWriteRawValueAsync(
    ReadOnlyMemory value,
    void cancel)
```

#### Parameters

- **value**
  - Type: `ReadOnlyMemory`

- **cancel**
  - Type: `System.Void`

#### Return Value

- Type: `Task<ResultWriteAccess>`

### Reference

- **Symbol Class**: [1469]
- **TwinCAT.Ads.TypeSystem Namespace**: [1328]

#### 6.7.23.2.13 Symbol.OnWriteValue Method

Handler function for writing the dynamic value

**Namespace**: TwinCAT.Ads.TypeSystem [1328]

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

#### Syntax

```csharp
protected virtual void OnWriteValue(
    Object value,
    int timeout)
```

#### Parameters

- **value**
  - Type: `System.Object`
  - The value.

- **timeout**
  - Type: `System.Int32`
  - The timeout.

#### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotAccessVirtualSymbolException [1641]</td>
<td></td>
</tr>
<tr>
<td>CannotAccessVirtualSymbolException [1641]</td>
<td></td>
</tr>
</tbody>
</table>
### 6.7.23.2.14 Symbol.OnWriteValueAsync Method

Handler function for writing the dynamic value

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected virtual Task<ResultWriteAccess> OnWriteValueAsync(
    Object value,
    CancellationToken cancel
)
```

**Parameters**

- `value`: Type: `System.Object`
  The value.

- `cancel`: Type: `System.Threading.CancellationToken`
  The cancellation token.

**Return Value**

Type: `Task<ResultWriteAccess>`

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotAccessVirtualSymbolException [1641]</td>
<td></td>
</tr>
</tbody>
</table>

### 6.7.23.2.15 Symbol.ReadAnyValue Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReadAnyValue(Type)</code></td>
<td>Reads the value of this <code>Value</code> into a new created instance of the managed type</td>
</tr>
</tbody>
</table>
TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAnyValue(Type, Int32)</td>
<td>Reads the value of this Value into a new created instance of the managed type</td>
</tr>
</tbody>
</table>

**Reference**

**Symbol Class** [1469]

TwinCAT.Ads.TypeSystem Namespace [1328]

**Symbol.ReadAnyValue Method (Type)**

Reads the value of this Value into a new created instance of the managed type

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Object ReadAnyValue(
    Type managedType
)
```

**Parameters**

managedType Type: System.Type
The tp.

**Return Value**

Type: Object
Read value (System.Object).

**Implements**

IValueAnySymbol.ReadAnyValue(Type) [2239]

**Reference**

**Symbol Class** [1469]

ReadAnyValue Overload [1496]

TwinCAT.Ads.TypeSystem Namespace [1328]

IValueAnySymbol.WriteAnyValue(Object) [2243]

IValueAnySymbol.UpdateAnyValue(Object) [2242]

**Symbol.ReadAnyValue Method (Type, Int32)**

Reads the value of this Value into a new created instance of the managed type
**TwinCAT.Ads Namespaces**

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Object ReadAnyValue(
    Type managedType,
    int timeout
)
```

**Parameters**

managedType

Type: System.Type

The tp.

timeout

Type: System.Int32

The timeout in ms.

**Return Value**

Type: Object

Read value (System.Object).

**Implements**

IValueAnySymbol.ReadAnyValue(Type, Int32)

**Reference**

Symbol Class

ReadAnyValue Overload

TwinCAT.Ads.TypeSystem Namespace

IValueAnySymbol.WriteAnyValue(Object)

IValueAnySymbol.UpdateAnyValue(Object.)

**6.7.23.2.16 Symbol.ReadAnyValueAsync Method**

Reads the (AnyType) value asynchronously.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public Task<ResultReadValueAccess> ReadAnyValueAsync(
    Type managedType,
    CancellationToken cancel
)
```
Parameters

managedType  Type:  System.Type
Managed type of the value to read.
cancel     Type:  System.Threading.CancellationToken
The cancellation token.

Return Value

Type:  Task<ResultReadValueAccess>  
A task object that is representing the asynchronous 'ReadAnyValue' operation. The result will be returned in a ResultReadValueAccess, which contains the Value and the ErrorCode.

Implements

IValueAnySymbol.ReadAnyValueAsync(Type, CancellationToken)  

Reference

Symbol Class  
TwinCAT.Ads.TypeSystem Namespace  

6.7.23.2.17  Symbol.ReadRawValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>➔ ReadRawValue</td>
<td>Reads the raw value of the IValueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>➔ ReadRawValue(Int32)</td>
<td>Reads the raw value of the IValueSymbol (Ads Read / Write)</td>
</tr>
</tbody>
</table>

Reference

Symbol Class  
TwinCAT.Ads.TypeSystem Namespace  

Symbol.ReadRawValue Method

Reads the raw value of the IValueSymbol (Ads Read / Write)

Namespace:  TwinCAT.Ads.TypeSystem  
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public byte[] ReadRawValue()
Field Value

Type: .Byte
The raw value.

Return Value

Type: .Byte
System.Byte[].

Implements

IValueRawSymbol.ReadRawValue

Reference

Symbol Class [1469]
ReadRawValue Overload [1499]
TwinCAT.Ads.TypeSystem Namespace [1328]

Symbol.ReadRawValue Method (Int32)

Reads the raw value of the IValueSymbol (Ads Read / Write)

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdca3e72bc0ea15da1c14

Syntax

C#

public byte[] ReadRawValue(
    int timeout
)

Parameters

timeout Type: System.Int32

Field Value

Type: .Byte
The raw value.

Return Value

Type: .Byte
System.Byte[].

Implements

IValueRawSymbol.ReadRawValue(Int32) [2250]
6.7.23.2.18 Symbol.ReadRawValueAsync Method

Reads the raw value of the [IValueSymbol](#2254) (Ads Read / Write) asynchronously.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public Task<ResultReadRawAccess> ReadRawValueAsync(
    CancellationToken cancel)
```

**Parameters**

- `cancel` Type: `System.Threading.CancellationToken`
  The cancellation token.

**Field Value**

Type: `Task<ResultReadRawAccess>` [2564].
The raw value.

**Return Value**

Type: `Task<ResultReadRawAccess>` [2564].
`System.Byte[]`.

**Implements**

`IValueRawSymbol.ReadRawValueAsync(CancellationToken)` [2251]

Reference

**Symbol Class** [1469]

**TwinCAT.Ads.TypeSystem Namespace** [1328]

6.7.23.2.19 Symbol.ReadValue Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReadValue()</code> [1502]</td>
<td>Reads the Value of the <a href="#2254">IValueSymbol</a></td>
</tr>
<tr>
<td><code>ReadValue(Int32)</code></td>
<td>[1502] Reads the Value of the <a href="#2254">IValueSymbol</a></td>
</tr>
</tbody>
</table>
TwinCAT.Ads Namespaces

Reference
Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

Symbol.ReadValue Method

Reads the Value of the IValueSymbol [2254]

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public Object ReadValue()'

Field Value

Type: Object
The value.

Return Value

Type: Object
System.Object.

Implements

IValueSymbol.ReadValue [2263]

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly.

Reference
Symbol Class [1469]
ReadValue Overload [1501]
TwinCAT.Ads.TypeSystem Namespace [1328]

Symbol.ReadValue Method (Int32)

Reads the Value of the IValueSymbol [2254]

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public Object ReadValue(
    int timeout
)
```

Parameters

timeout Type: System.Int32
The timeout in ms.

Field Value

Type: Object
The value.

Return Value

Type: Object
System.Object.

Implements

IValueSymbol.ReadValue(Int32) [2263]

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

Symbol Class [1469]
ReadValue Overload [1501]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.2.20 Symbol.ReadValueAsync Method

Reads the Value of the IValueSymbol [2254] asynchronously.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultReadValueAccess> ReadValueAsync(
    CancellationToken cancel
)
```
Parameters

cancel

Type: `System.Threading.CancellationToken`
The cancellation token.

Return Value

Type: `Task<ReadValueAccess>`
A task that represents the asynchronous 'ReadValue' operation. The read result is stored in the `ReadValueAccess` return value and contains the `Value` and the `ErrorCode`.

Implements

`IValueSymbol.ReadValueAsync(CancellationToken)`

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the `ISymbolLoader` settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly.

Reference

Symbol Class

TwinCAT.Ads.TypeSystem Namespace

6.7.23.2.21 Symbol.SetParent Method

Sets the parent symbol.

Namespace: `TwinCAT.Ads.TypeSystem`

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb91a43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

class

public void SetParent(
    ISymbol parent
)

Parameters

parent

Type: `TwinCAT.TypeSystem.ISymbol`
The parent.

Implements

`IHierarchicalSymbol.SetParent(ISymbol)`

Reference

Symbol Class

TwinCAT.Ads.TypeSystem Namespace
6.7.23.2.22 Symbol.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads.dll Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

Symbol Class
TwinCAT.Ads.TypeSystem

6.7.23.2.23 Symbol.TryReadValue Method

Reads the Value of the IValueSymbol

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads.dll Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int TryReadValue(
    int timeout,
    out Object value
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| timeout   | System.Int32
            The timeout in ms.|
| value     | System.Object
            The read value.|

Return Value

Type: Int32
The error Code.

Implements

IValueSymbol.TryReadValue(Int32, Object)
Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.2.24 Symbol.TryWriteValue Method

Writes the Value of the IValueSymbol [2254]

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sh.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int TryWriteValue(
    Object value,
    int timeout
)
```

Parameters

- value: Type: System.Object
  The value.
- timeout: Type: System.Int32
  The timeout in ms.

Return Value

Type: Int32
The error code.

Implements

IValueSymbol.TryWriteValue(Object, Int32) [2265]

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the SymbolLoader settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]
## Symbol.UpdateAnyValue Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateAnyValue(Object.)</td>
<td>Reads the value of this Value[2254] into the specified managed value.</td>
</tr>
<tr>
<td>UpdateAnyValue(Object., Int32)</td>
<td>Reads the value of this Value[2254] into the specified managed value.</td>
</tr>
</tbody>
</table>

### Reference

**Symbol Class** [1469]

**TwinCAT.Ads.TypeSystem Namespace** [1328]

## Symbol.UpdateAnyValue Method (Object.)

Reads the value of this Value [2254] into the specified managed value.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

C#  
```csharp
public void UpdateAnyValue(
    ref Object managedObject
)
```

### Parameters

- **managedObject**  
  Type: System.Object.
  The managed object.

### Return Value

Type: Read value (System.Object).

### Implements

IValueAnySymbol.UpdateAnyValue(Object.) [2242]

### Reference

**Symbol Class** [1469]

**UpdateAnyValue Overload** [1507]

**TwinCAT.Ads.TypeSystem Namespace** [1328]

IValueAnySymbol.ReadAnyValue(Type) [2239]
Symbol.UpdateAnyValue Method (Object, Int32)

Reads the value of this Value [2254] into the specified managed value.

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void UpdateAnyValue(
    ref Object managedObject,
    int timeout
)
```

Parameters

- **managedObject**: Type: System.Object
  The managed object.
- **timeout**: Type: System.Int32
  The timeout.

Return Value

Type: Read value (System.Object).

Implements

IValueAnySymbol.UpdateAnyValue(Object, Int32) [2242]

Reference

Symbol Class [1469]
UpdateAnyValue Overload [1507]
TwinCAT.Ads.TypeSystem Namespace [1328]
IValueAnySymbol.ReadAnyValue(Type) [2239]
IValueAnySymbol.WriteAnyValue(Object) [2243]

6.7.23.2.26 Symbol.WriteRawValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteRawValue(Byte...)</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WriteRawValue (.Byte., Int32)</td>
<td>Writes the raw value of the [IValueSymbol][2254] (Ads Read / Write)</td>
</tr>
</tbody>
</table>

**Reference**

Symbol Class [1469]

TwinCAT.Ads.TypeSystem Namespace [1328]

**Symbol.WriteRawValue Method (.Byte.)**

 Writes the raw value of the [IValueSymbol][2254] (Ads Read / Write)

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddc2a3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void WriteRawValue(
    byte[] rawValue
)
```

**Parameters**

rawValue Type: `System.Byte`

The value as byte array.

**Field Value**

Type: The value.

**Implements**

[IValueRawSymbol.WriteRawValue (.Byte.)][2252]

**Reference**

Symbol Class [1469]

WriteRawValue Overload [1508]

TwinCAT.Ads.TypeSystem Namespace [1328]

**Symbol.WriteRawValue Method (.Byte., Int32)**

 Writes the raw value of the [IValueSymbol][2254] (Ads Read / Write)

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddc2a3e72bc0ea15da1c14
Syntax

C#

```csharp
public void WriteRawValue(
    byte[] rawValue,
    int timeout
)
```

**Parameters**


**Field Value**

Type: The raw value.

**Implements**

- `IValueRawSymbol.WriteRawValue(Byte, Int32)` [2252]

**Remarks**

A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

**Reference**

- Symbol Class [1469]
- WriteRawValue Overload [1508]
- TwinCAT.Ads.TypeSystem Namespace [1328]

### 6.7.23.2.27 Symbol.WriteRawValueAsync Method

Writes the raw value of the `IValueSymbol` (Ads Read / Write)

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultWriteAccess> WriteRawValueAsync(
    byte[] rawValue,
    CancellationToken cancel
)
```

**Parameters**

Return Value

Type: Task.ResultWriteAccess
A task that represents the asynchronous read operation. The ResultRead parameter contains the total number of bytes read into the buffer (ReadBytes) and the ErrorCode after execution.

Implements

IValueRawSymbol.WriteRawValueAsync(Byte, CancellationToken)

Reference

Symbol Class
TwinCAT.Ads.TypeSystem Namespace

6.7.23.2.28 Symbol.WriteValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValue(Object)</td>
<td>Writes the Value of the IValueSymbol</td>
</tr>
<tr>
<td>WriteValue(Object, int32)</td>
<td>Writes the Value of the IValueSymbol</td>
</tr>
</tbody>
</table>

Reference

Symbol Class
TwinCAT.Ads.TypeSystem Namespace

Symbol.WriteValue Method (Object)

Writes the Value of the IValueSymbol

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteValue(
    Object value
)
```

Parameters

value          Type: System.Object
The value.
Implements

IValueSymbol.WriteValue(Object) [» 2266]

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader [» 2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also.

Reference

Symbol Class [» 1469]

WriteValue Overload [» 1511]

TwinCAT.Ads.TypeSystem Namespace [» 1328]

Symbol.WriteValue Method (Object, Int32)

Writes the Value of the IValueSymbol [» 2254]

Namespace: TwinCAT.Ads.TypeSystem [» 1328]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public void WriteValue(
    Object value,
    int timeout
)

Parameters

value Type: System.Object
The value.

timeout Type: System.Int32
The timeout in ms.

Implements

IValueSymbol.WriteValue(Object, Int32) [» 2267]

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader [» 2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

Symbol Class [» 1469]
6.7.23.2.29 Symbol.WriteValueAsync Method

Writes the Value of the IValueSymbol

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd5ca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultWriteAccess> WriteValueAsync(
    Object value,
    CancellationToken cancel)
```

Parameters

- **value**
  - Type: System.Object
  - The value.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token.

Return Value

- **Type**: Task<ResultWriteAccess>

A task that represents the asynchronous 'ReadValue' operation. The read result is stored in the ResultWriteAccess return value and contains the ErrorCode.

Implements

IValueSymbol.WriteValueAsync(Object, CancellationToken)

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also.

Reference

Symbol Class
TwinCAT.Ads.TypeSystem Namespace

6.7.23.3 Symbol Events

The Symbol type exposes the following members.
Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol has changed.</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol has changed.</td>
</tr>
</tbody>
</table>

Reference

Symbol Class [1469]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.3.1 Symbol.RawValueChanged Event

Occurs when the RawValue of the IValueSymbol has changed.

Namespace: TwinCAT.Ads.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public event EventHandler<RawValueChangedEventArgs> RawValueChanged
```

Value

Type: System.EventHandler<RawValueChangedEventArgs>

Implements

IValueRawSymbol.RawValueChanged

Reference

Symbol Class [1469]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.3.2 SymbolValueChanged Event

Occurs when the (Primitive) value of the IValueSymbol has changed.

Namespace: TwinCAT.Ads.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public event EventHandler<ValueChangedEventArgs> ValueChanged
```
Value
Type: System.EventHandler_ValueChangedEventArgs

Implements
ValueSymbol_ValueChanged

Reference
Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.4 Symbol Operators

The Symbol [1469] type exposes the following members.

Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operator==</td>
<td>Operator==</td>
</tr>
<tr>
<td>Operator!=</td>
<td>Implements the != operator.</td>
</tr>
</tbody>
</table>

Reference
Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.4.1 Symbol.Equality Operator

Operator==

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public static bool operator ==(
    Symbol o1,
    Symbol o2
)

Parameters

o1 Type: TwinCAT.Ads.TypeSystem.Symbol [1469]
The o1.
6.7.23.4.2  Symbol.Inequality Operator

Implements the != operator.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public static bool operator !=(
    Symbol o1,
    Symbol o2
)
```

**Parameters**

<table>
<thead>
<tr>
<th>o1</th>
<th>Type: TwinCAT.Ads.TypeSystem.Symbol [1469]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The o1.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>o2</th>
<th>Type: TwinCAT.Ads.TypeSystem.Symbol [1469]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The o2.</td>
</tr>
</tbody>
</table>

**Return Value**

Type: Boolean
The result of the operator.

**Reference**

Symbol Class [1469]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.5  Symbol Fields

The Symbol [1469] type exposes the following members.

**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>syncObject</td>
<td>Synchronization object</td>
</tr>
</tbody>
</table>
Reference
Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.23.5.1 Symbol.syncObject Field
Synchronization object

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected Object syncObject

Field Value
Type: Object

Reference
Symbol Class [1469]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.24 SymbolIterator Class
Iterator class for enumerations of Symbols [2176].

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Generic.SymbolIterator [2532]: ISymbol [2176].
    TwinCAT.Ads.TypeSystem.SymbolIterator

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public class SymbolIterator : SymbolIterator<ISymbol>

The SymbolIterator type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolIterator(InstanceIdCollection.ISymbol) [1520]</td>
<td>Initializes a new instance of the SymbolIterator class.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### TwinCAT.Ads.NameSpace

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolIterator(IInstanceCollection.ISymbol, Func.ISymbol, Boolean)</td>
<td>Initializes a new instance of the SymbolIterator class.</td>
</tr>
<tr>
<td>SymbolIterator(IEnumerable.ISymbol, Boolean)</td>
<td>Initializes a new instance of the SymbolIterator class.</td>
</tr>
<tr>
<td>SymbolIterator(IEnumerable.ISymbol, Boolean, Func.ISymbol, Boolean)</td>
<td>Initializes a new instance of the SymbolIterator class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask</td>
<td>Gets or sets the SymbolIterationMask (Inherited from SymbolIterator.T.)</td>
</tr>
<tr>
<td>SymbolRecursionDetection</td>
<td>Gets or sets a value indicating whether the iterator checks for Symbol recursions (true by default). (Inherited from SymbolIterator.T.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Gets the enumerator that enumerates through a collection (Inherited from SymbolIterator.T.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Remarks

This iterator class can be used to iterate over collections of symbol trees (root symbols + sub symbols). By constructor the user can choose if the iterator works recursively within the symbol tree and optionally a filter function to select only specific symbols (predicate).

### Examples

The following example shows how to determine, browse and filter symbols.
Browsing and filtering Symbols

```csharp
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;
    uint valueToRead = 0;
    uint valueToWrite = 42;
    client.Connect(AmsNetId.Local, 851);
    // Load all Symbols + DataTypes
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    ResultSymbols resultSymbols = await loader.GetSymbolsAsync(cancel);
    if (resultSymbols.Succeeded)
    {
        Symbol symbol = (Symbol)resultSymbols.Symbols["MAIN.nCounter"];
        // Works for ALL Primitive 'ANY TYPES' Symbols
        ResultWriteAccess resultWrite = await symbol.WriteValueAsync(valueToWrite, cancel);
        ResultReadValueAccess resultRead = await symbol.ReadValueAsync(cancel);
        if (resultRead.Succeeded)
        {
            valueToRead = (uint)resultRead.Value;
        }
    }
}
```

Reference

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.24.1 SymbolIterator Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="" alt="" /> SymbolIterator(IInstanceCollection.ISymbol) [1520]</td>
<td>Initializes a new instance of the SymbolIterator [1517] class.</td>
</tr>
<tr>
<td><img src="" alt="" /> SymbolIterator(IInstanceCollection.ISymbol, Func.ISymbol, Boolean) [1520]</td>
<td>Initializes a new instance of the SymbolIterator [1517] class.</td>
</tr>
<tr>
<td><img src="" alt="" /> SymbolIterator(IEnumerable.ISymbol, Boolean) [1521]</td>
<td>Initializes a new instance of the SymbolIterator [1517] class.</td>
</tr>
</tbody>
</table>
SymbolIterator Constructor (IInstanceCollection.ISymbol.)

Initializes a new instance of the SymbolIterator class.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public SymbolIterator(IInstanceCollection<ISymbol> symbols)
```

**Parameters**

symbols Type: TwinCAT.TypeSystem.IInstanceCollection.ISymbol.

The symbol collection

**Reference**

SymbolIterator Class

SymbolIterator Overload

TwinCAT.Ads.TypeSystem Namespace

6.7.24.1.2 SymbolIterator Constructor (IInstanceCollection.ISymbol., Func.ISymbol, Boolean.)

Initializes a new instance of the SymbolIterator class.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
### Syntax

**C#**

```csharp
public SymbolIterator(
    IInstanceCollection<ISymbol> symbols,
    Func<ISymbol, bool> predicate
)
```

**Parameters**

- **symbols**
  - Type: `TwinCAT.TypeSystem.IInstanceCollection<ISymbol>`
  - The symbol collection.

- **predicate**
  - Type: `System.Func<ISymbol, Boolean>`
  - The predicate / filter function

**Reference**

SymbolIterator Class [1517]
SymbolIterator Overload [1519]
TwinCAT.Ads.TypeSystem Namespace [1328]

#### 6.7.24.1.3 SymbolIterator Constructor (IEnumerable<ISymbol>, Boolean)

Initializes a new instance of the `SymbolIterator` class.

**Namespace:** TwinCAT.Ads.TypeSystem [1328]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public SymbolIterator(
    IEnumerable<ISymbol> symbols,
    bool recurse
)
```

**Parameters**

- **symbols**
  - Type: `System.Collections.Generic(IEnumerable<ISymbol>>`
  - The symbol enumeration.

- **recurse**
  - Type: `System.Boolean`
  - if set to true, the iterator works recursively over all subsymbols.

**Reference**

SymbolIterator Class [1517]
SymbolIterator Overload [1519]
TwinCAT.Ads.TypeSystem Namespace [1328]

#### 6.7.24.1.4 SymbolIterator Constructor (IEnumerable<ISymbol>, Boolean, Func<ISymbol, Boolean>)

Initializes a new instance of the `SymbolIterator` class.
TwinCAT.Ads Namespaces

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public SymbolIterator(
    IEnumerable<ISymbol> symbols,
    bool recurse,
    Func<ISymbol, bool> predicate
)
```

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbols</td>
<td>IEnumerable&lt;ISymbol&gt;</td>
<td>The symbol collection.</td>
</tr>
<tr>
<td>recurse</td>
<td>System.Boolean</td>
<td>if set to true, the iterator works recursively over all subsymbols.</td>
</tr>
<tr>
<td>predicate</td>
<td>System.Func&lt;ISymbol, Boolean&gt;</td>
<td>The predicate / filter function.</td>
</tr>
</tbody>
</table>

Reference

SymbolIterator Class [1517]
SymbolIterator Overload [1519]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.24.2 SymbolIterator Properties

The SymbolIterator [1517] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask [2538]</td>
<td>Gets or sets the SymbolIterationMask [2532] (Inherited from SymbolIterator.T. [2532].)</td>
</tr>
<tr>
<td>SymbolRecursionDetection [2538]</td>
<td>Gets or sets a value indicating whether the iterator checks for Symbol recursions (true by default). (Inherited from SymbolIterator.T. [2532].)</td>
</tr>
</tbody>
</table>

Reference

SymbolIterator Class [1517]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.24.3 SymbolIterator Methods

The SymbolIterator [1517] type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Gets the enumerator that enumerates through a collection (Inherited from SymbolIterator.T. [2539].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

SymbolIterator Class [1517]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.25 SymbolLoaderFactory Class

The class SymbolLoaderFactory is used to create a new instance of the AdsSymbolLoader initialized to the parametrized mode (SymbolBrowser V2, new Version)

Inheritance Hierarchy

System.Object
   TwinCAT.Ads.TypeSystem.SymbolLoaderFactory

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public static class SymbolLoaderFactory

The SymbolLoaderFactory type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>Creates the specified connection.</td>
</tr>
</tbody>
</table>

TC1000 Version: 1.1 1523
Remarks

The Symbol Loader (V2) supports the following modes. The flat mode organizes the Symbols in a flat list. This mode is available in all .NET versions. The virtual tree mode organizes the Symbols hierarchically with parent-child relationships. This mode is available in all .NET Versions. The Dynamic tree mode organizes the Symbols hierarchically and creates struct members, array elements and enum fields on the fly. This feature is only available on platforms that support the Dynamic Language Runtime (DLR), actually all .NET Framework Version larger than 4.0. Virtual instances means, that all Symbols are ordered within a tree structure. For that symbol nodes that are not located on a fixed address, a Virtual Symbol will be created. Setting the virtualInstance parameter to 'false' means, that the located symbols will be returned in a flattened list.

Reference

TwinCAT.Ads.TypeSystem Namespace

TwinCAT.SymbolLoaderSettings

6.7.25.1 SymbolLoaderFactory Methods

The SymbolLoaderFactory type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>Creates the specified connection.</td>
</tr>
</tbody>
</table>

Reference

SymbolLoaderFactory Class

TwinCAT.Ads.TypeSystem Namespace

6.7.25.1.1 SymbolLoaderFactory.Create Method

Creates the specified connection.

Namespace: TwinCAT.Ads.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static ISymbolLoader Create(
    IConnection connection,
    ISymbolLoaderSettings settings
)
```

Parameters

connection Type: TwinCAT.IConnection

The connection.
settings  

Type: TwinCAT.ISymbolLoaderSettings [99]

The settings.

Return Value

Type: ISymbolLoader [2200]

ISymbolLoader.

Examples

The following sample shows how to create a dynamic version of the SymbolLoader V2. The dynamic symbol loader makes use of the Dynamic Language Runtime (DLR) of the .NET Framework. That means Structures, Arrays and Enumeration types and instances are generated 'on-the-fly' during symbol Browsing. These created dynamic objects are a one to one representation of the Symbol Server target objects (e.g the IEC61131 types on the PLC). Dynamic language features are only available from .NET4 upwards.

Dynamic Tree Mode

namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT;
    using TwinCAT.Ads;
    using TwinCAT.Ads.TypeSystem;
    using TwinCAT.Ads.ValueAccess;
    using TwinCAT.TypeSystem;
    using TwinCAT.TypeSystem.Generic;
    using TwinCAT.ValueAccess;
    class SymbolBrowserProgramV2DynamicTree
    {
        #region CODE_SAMPLE_SIMPLEDYNAMIC
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static async void Main(string[] args)
        {
            // Get the AdsAddress from command-line arguments
            AmsAddress address = ArgParser.Parse(args);
            CancellationTokenSource cancelSource = new CancellationTokenSource();
            CancellationToken cancel = cancelSource.Token;

            using (AdsClient client = new AdsClient())
            {
                // Connect to the target device
                client.Connect(address);

                // Usage of "dynamic" Type and Symbols (>= .NET4 only)
                SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
                IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);
                
                // Set the Default setting for Notifications
                dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransModeOnChange, 200, 2000);

                // Get the Symbols (Dynamic Symbols)
                var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);

                dynamic dynamicSymbols = resultSymbols.Symbols;
                dynamic adsPort = dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.AdsPort;

                #region CODE_SAMPLE_SIMPLEDYNAMIC
                // Access Main Symbol with Dynamic Language Runtime support (DLR)
                // Dynamically created property "Main"
                dynamic symMain = dynamicSymbols.Main;
        
        #endregion
        
        #endregion
    }
}
// Main is an 'VirtualSymbol' / Organizational unit that doesn't have a value
// Calling ReadValue is not allowed
//bool test = symMain.HasValue;
//dynamic invalid = symMain.ReadValue();

//Reading TaskInfo Value

With calling ReadValueAsync() a 'snapshot' of the Symbols Instance is taken (reading async)
ResultReadValueAccess resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo.
ReadValueAsyncAsync(cancel);
dynamic vTaskInfoArray = resultRead.Value;

// Getting the Snapshot time in UTC format
DateTimeOffset timeStamp1 = vTaskInfoArray.TimeStamp;

// Getting TaskInfo Symbol for Task 1
dynamic symTaskInfo1 = dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo[1];

// Getting CycleCount Symbol
dynamic symCycleCount = symTaskInfo1.CycleCount;

// Take Snapshot value of the ApplicationInfo struct
resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.ReadValueAsyncAsync(cancel);
dynamic vAppInfo = resultRead.Value;

// Get the UTC Timestamp of the snapshot
DateTimeOffset timeStamp2 = vAppInfo.TimeStamp;

// Access the ProjectName of the ApplicationInfo Snapshot (type-safe!)
string projectNameValue = vAppInfo.ProjectName;

// Reading the CycleCount Value
resultRead = await symTaskInfo1.CycleCount.ReadValueAsyncAsync(cancel); // Taking a Value Snap
pshot
int cycleCountValue = (int)resultRead.Value;
#endif

// Registering for dynamic "ValueChanged" events for the Values
// Using Default Notification settings
symCycleCount.ValueChanged += new EventHandler<ValueChangedEventArgs>(cycleCount_ValueChanged);

// Override default notification settings
symTaskInfo1NotificationSettings = new NotificationSettings(AdsTransMode.Cyclic, 500, 0);

// Register forValueChanged event.
symTaskInfo1ValueChanged += new EventHandler<ValueChangedEventArgs>(taskInfo1ValueChanged); // Struct Type

Thread.Sleep(10000); // Sleep main thread for 10 Seconds
}
Console.WriteLine("CycleCount Changed events received: {0}", _cycleCountEvents);
Console.WriteLine("taskInfo1 Changed events received: {0}", _taskInfo1Events);
Console.WriteLine("Press [Enter] for leave:");
Console.ReadLine();

static object _notificationSynchronizer = new object();
static int _cycleCountEvents = 0;

/// <summary>
/// Handler function for the CycleCount ValueChanged event.
/// </summary>
/// <param name="sender">The sender.</param>
/// <param name="e">The e.</param>
static void cycleCount_ValueChanged(object sender, ValueChangedEventArgs e)
{
lock(_notificationSynchronizer)
{
Interlocked.Increment(ref _cycleCountEvents);
// val is a type safe value of int!
dynamic val = e.Value;
uint intVal = val;

DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert UTC to local time
Console.WriteLine("CycleCount changed to: {0}, TimeStamp: {1}", intVal, changedTime.ToString
("HH:mm:ss:fff"));
}
The following sample shows how to create a static (non dynamic) version of the SymbolLoader V2. The static symbol loader in version 2 is a nearly code compatible version of the Dynamic Loader, only the dynamic creation of objects is not available. The reason for supporting this mode is that .NET Framework Versions lower than Version 4.0 (CLR2) doesn’t support the Dynamic Language Runtime (DLR). The SymbolLoader V2 static object is supported from .NET 2.0 on.

**Virtual Tree Mode**

```csharp
using System;
using System.Threading;
using System.Diagnostics;
using TwinCAT;
using TwinCAT.Ads;
using TwinCAT.TypeSystem;
using TwinCAT.TypeSystem.Generic;
using TwinCAT.Ads.ValueAccess;
using TwinCAT.Ads.TypeSystem;
namespace Sample
{
    class SymbolBrowserProgramV2VirtualTree
    {
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static void Main(string[] args)
        {
            ConsoleLogger logger = new ConsoleLogger();
            Console.WriteLine("\n");
            Console.WriteLine("Press [Enter] for start:\n");
            Console.ReadLine();
            //logger.Active = false;
            Stopwatch stopper = new Stopwatch();
            // Parse the command-line arguments
            AmsAddress address = ArgParser.Parse(args);
            stopper.Start();
            using (AdsClient client = new AdsClient())
            {
                //client.Synchronize = false;
                // Connect the AdsClient to the device target.
            }
        }
    }
}
```
// Creates the Symbol Objects as hierarchical tree
SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree, ValueAccessMode.IndexGroupOffsetPreferred);
ISymbolLoader symbolLoader = SymbolLoaderFactory.Create(client, settings);

// Dump Datatypes from Target Device
Console.WriteLine($"Dumping '{symbolLoader.DataTypes.Count}' DataTypes:");
foreach (IDataType type in symbolLoader.DataTypes)
{
    logger.DumpType(type);
}
Console.WriteLine("\n");

// Dump Symbols from target device
Console.WriteLine($"Dumping '{symbolLoader.Symbols.Count}' Symbols:");
foreach (ISymbol symbol in symbolLoader.Symbols)
{
    logger.DumpSymbol(symbol, 0);
}
stopper.Stop();
TimeSpan elapsed = stopper.Elapsed;

Console.WriteLine("\n")
Console.WriteLine($"Browsing complete tree: {elapsed},

{{1} DataTypes, {2} Symbols}, elapsed, logger.DataTypesCount, logger.DataTypesCount};
Console.WriteLine("Press [Enter] for leave:");
Console.ReadLine();
}

Examples
The SymbolLoader V2 static object is supported from .NET 2.0 on.

Flat Mode
using System;
using System.Diagnostics;
using System.Threading;
using TwinCAT;
using TwinCAT.Ads;
using TwinCAT.Ads.TypeSystem;
using TwinCAT.Ads.ValueAccess;
using TwinCAT.TypeSystem;
using TwinCAT.TypeSystem.Generic;
namespace Sample
{
    class SymbolBrowserProgramV2Flat
    {
        static void Main(string[] args)
        {
            ConsoleLogger logger = new ConsoleLogger();
            Console.WriteLine("\n");
            Console.WriteLine("Press [Enter] for start:");
            Console.ReadLine();
            logger.Active = false;
            Stopwatch stopper = new Stopwatch();
            // Parse the command line arguments
            AmsAddress address = ArgParser.Parse(args);
            stopper.Start();
            // Create the ADS Client
            using (AdsClient client = new AdsClient())
            {
                //client.Synchronize = false;
                // Connect to Address
                client.Timeout = 30000;
client.Connect(address);

// Creates the Symbol Objects in Flat Mode (Flat list)
SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.Flat, ValueAccessMode.IndexGroupOffsetPreferred);
ISymbolLoader symbolLoader = SymbolLoaderFactory.Create(client, settings);

// Dump Datatypes from Target Device
Console.WriteLine(string.Format("Dumping '{0}' DataTypes:",symbolLoader.DataTypes.Count));
foreach (IDataType type in symbolLoader.DataTypes)
{
    logger.DumpType(type);
}
Console.WriteLine("\n");

// Dump Symbols from target device
Console.WriteLine("Dumping '{0}' Symbols:",symbolLoader.Symbols.Count);
foreach (ISymbol symbol in symbolLoader.Symbols)
{
    logger.DumpSymbol(symbol,0);
}
stopper.Stop();
TimeSpan elapsed = stopper.Elapsed;
Console.WriteLine("\n");
Console.WriteLine("Browsing complete tree: {0}, (1) DataTypes, (2) Symbols", elapsed, logger.DataTypesCount, logger.DataTypesCount);
Console.WriteLine("Press [Enter] for leave:");
Console.ReadLine();

Examples

Argument Parser

public static class ArgParser
{
    /// <summary>
    /// Parses the arguments.
    /// </summary>
    /// <param name="args">The arguments.</param>
    /// <returns>AmsAddress.</returns>
    public static AmsAddress Parse(string[] args)
    {
        AmsNetId netId = AmsNetId.Local;
        int port = 851;
        if (args != null)
        {
            if (args.Length > 0 && args[0] != null)
                netId = AmsNetId.Parse(args[0]);
            if (args.Length > 1 && args[1] != null)
                port = int.Parse(args[1]);
        }
        return new AmsAddress(netId, port);
    }
}

Dumping Symbols

/// <summary>
/// Console logger
/// </summary>
/// <summary>
public class ConsoleLogger
{
    public ConsoleLogger()
    {
    }
    bool _active = true;

    /// <summary>
    /// Gets or sets a value indicating whether this ConsoleLogger is active.
    /// </summary>
    /// <value><c>true</c> if active; otherwise, <c>false</c>.</value>
    public bool Active
```csharp
int _dataTypes = 0;

/// <summary>
/// Gets the number of dumped dataTypes.
/// </summary>
/// <value>The data types count.</value>
public int DataTypesCount
{
    get { return _dataTypes; }
}

int _symbols = 0;

/// <summary>
/// Gets the number of dumped symbols
/// </summary>
/// <value>The symbols count.</value>
public int SymbolsCount
{
    get { return _symbols; }
}

/// <summary>
/// Dumps the data type.
/// </summary>
/// <param name="dataType">Data Type.</param>
public void DumpType(IDataType dataType)
{
    WriteLine(string.Format("DataType: {0}, Category: {1}, Size: {2}", dataType.Name, dataType.Category, dataType.Size));
    switch (dataType.Category)
    {
    case DataTypeCategory.Alias:
        IAliasType alias = (IAliasType)dataType;
        WriteLine(GetPrefix(1) + string.Format("Alias BaseType: {0}", alias.BaseTypeName));
        break;

    case DataTypeCategory.Enum:
        //IEnumType<ushort> enumType = (IEnumType<ushort>)dataType;
        IEnumType enumType = (IEnumType)dataType;
        WriteLine(GetPrefix(1) + string.Format("Enum BaseType: {0}", enumType.BaseTypeName));
        foreach (IEnumValue enumValue in enumType.EnumValues)
        {
            WriteLine(GetPrefix(2) + string.Format("Name: {0}, Value: {1}", enumValue.Name, enumValue.Primitive));
        }
        break;

    case DataTypeCategory.Array:
        IArrayType arrayType = (IArrayType)dataType;
        int i = 0;
        foreach (IDimension dim in arrayType.Dimensions)
        {
            WriteLine(GetPrefix(2) + string.Format("{0}: LowerBound: {1}, Elements: {2}"", i++, dim.LowerBound, dim.ElementCount));
        }
        break;

    case DataTypeCategory.Struct:
        IStructType structType = (IStructType)dataType;
        foreach (IMember member in structType.Members)
        {
            WriteLine(GetPrefix(2) + string.Format("Offset {0}: Name: {1}, Type: {2}"", member.Offset, member.InstanceName, member.TypeName));
        }
        break;
    default:
```
foreach (ITypeAttribute attribute in dataType.Attributes)
{
    WriteLine(GetPrefix(1) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Value));
}
if (!string.IsNullOrEmpty(dataType.Comment))
{
    WriteLine(GetPrefix(1) + string.Format("Comment: [{0}]", dataType.Comment));
}
IRpcCallableType rpcCallable = dataType as IRpcCallableType;
if (rpcCallable != null)
{
    foreach (IRpcMethod rpcMethod in rpcCallable.RpcMethods)
    {
        if (string.IsNullOrEmpty(rpcMethod.Comment))
            WriteLine(GetPrefix(1) + string.Format("Method: [{0}]", rpcMethod));
        else
            WriteLine(GetPrefix(1) + string.Format("Method: [{0}], Comment: [{1}]", rpcMethod, rpcMethod.Comment));
    }
    _dataTypes++;
}
///// <summary>
///// Dumps the Datatype to Console
///// </summary>
///// <param name="dataType">DataType.</param>
//public void DumpType(ITcAdsDataType dataType)
//{
//    // Dump the Attributes (PLC Metadata)
//    foreach (ITypeAttribute attribute in dataType.Attributes)
//    {
//        WriteLine(GetPrefix(1) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Value));
//    }
//    WriteLine(string.Format("DataType: [{0}], Category: [{1}], Size: [{2}]", dataType.Name, dataType.Category, dataType.Size));
//    if (dataType.BaseType != null)
//    {
//        WriteLine(GetPrefix(1) + string.Format("BaseType: [{0}]", dataType.BaseType));
//    }
//    switch (dataType.Category)
//    {
//        case DataTypeCategory.Enum:
//            foreach (IEnumValue enumValue in dataType.EnumValues)
//            {
//                WriteLine(GetPrefix(2) + string.Format("Name: [{0}], Value: [{1}]", enumValue.Name, enumValue.Primitive));
//            }
//            break;
//        case DataTypeCategory.Array:
//            int i = 0;
//            foreach (IDimension dim in dataType.Dimensions)
//            {
//                WriteLine(GetPrefix(2) + string.Format("{0}: LowerBound: [{1}], Elements: [{2}]"), i++, dim.LowerBound, dim.ElementCount));
//            }
//            break;
//        case DataTypeCategory.Struct:
//            foreach (ITcAdsSubItem subItem in dataType.SubItems)
//            {
//                WriteLine(GetPrefix(2) + string.Format("Offset [{0}]: Name: [{1}], Type: [{2}]", subItem.Offset, subItem.SubItemName, subItem.Name));
//            }
//            break;
//        default:
//            break;
//    }
//    _dataTypes++;
//}
///// <summary>
/// Dump Symbol
/// </summary>
/// <param name="symbol">The symbol.</param>
/// <param name="level">Output indentation level</param>
public void DumpSymbol(ISymbol symbol, int level)
{
    IDdataType type = symbol.DataType as IDdataType;
    foreach (ITypeAttribute attribute in symbol.Attributes)
    {
        WriteLine(GetPrefix(level) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Value));
    }
    WriteLine(GetPrefix(level) + string.Format("{0} : {1} (IG: 0x{2} IO: 0x{3} size: {4})", symbol.InstanceName, symbol.TypeName, ((IAdsSymbol)symbol).IndexGroup.ToString("x"), ((IAdsSymbol)symbol).IndexOffset.ToString("x"), symbol.Size));
    if (symbol.Category == DataTypeCategory.Array)
    {
        IArrayInstance arrInstance = (IArrayInstance)symbol;
        IArrayType arrType = (IArrayType)symbol.DataType;
        int count = 0;
        level++;
        foreach (ISymbol arrayElement in arrInstance.Elements)
        {
            DumpSymbol(arrayElement, level);
            count++;
            if (count > 20) // Write only the first 20 to limit output
                break;
        }
    } else if (symbol.Category == DataTypeCategory.Struct)
    {
        IStructInstance structInstance = (IStructInstance)symbol;
        IStructType structType = (IStructType)symbol.DataType;
        level++;
        foreach (ISymbol member in structInstance.MemberInstances)
        {
            DumpSymbol(member, level);
        }
    }
    symbols++;
}

///// Dumps the specified Symbol to the Console
///// </summary>
///// <param name="symbol">The symbol.</param>
///// <param name="level">The level.</param>
//public void DumpSymbol(IAdsSymbol2 symbol, int level)
//{
//    // Dump Attributes of the Symbol
//    foreach (ITypeAttribute attribute in symbol.Attributes)
//    {
//        WriteLine(GetPrefix(level) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Value));
//    }
//    ITcAdsSymbolBrowser subSymbolProvider = (ITcAdsSymbolBrowser)symbol;
//    // Dump The Symbol
//    WriteLine(GetPrefix(level) + string.Format("{0} : {1} ({2}, IG: 0x{3} IO: 0x{4} size: {5})", symbol.Name, symbol.TypeName, symbol.DataTypeId, symbol.IndexGroup.ToString("x"), symbol.IndexOffset.ToString("x"), subSymbolProvider.SubSymbols.Count, symbol.Size));
//    level++;
//    // Dump all SubSymbols with indentation
//    foreach (IAdsSymbol2 subSymbol in ((ITcAdsSymbolBrowser)symbol).SubSymbols)
//    {
//        DumpSymbol(subSymbol, level);
//    }
//    _symbols++;
//}
/// <summary>
/// Dump namespace.
/// </summary>
/// <param name="ns">The namespace.</param>
public void DumpNamespace(INamespace<IDataType> ns)
{
    WriteLine("Namespace: {0}, DataTypes: {1}", ns.Name, ns.DataTypes.Count);
    foreach (IDataType type in ns.DataTypes)
    {
        DumpType(type);
    }
}

/// <summary>
/// Get the indentation prefix
/// </summary>
/// <param name="level">The level.</param>
/// <returns>System.String.</returns>
public string GetPrefix(int level)
{
    return ".".PadLeft(level * 3);
}

/// <summary>
/// Writes a line to the Console
/// </summary>
/// <param name="message">The message.</param>
public void WriteLine(string message)
{
    if (Active)
    {
        Console.WriteLine(message);
    }
}

/// <summary>
/// Writes a line to the console
/// </summary>
/// <param name="format">The format.</param>
/// <param name="args">The arguments.</param>
public void WriteLine(string format, params object[] args)
{
    if (Active)
    {
        Console.WriteLine(format, args);
    }
}

Reference
SymbolLoaderFactory Class [1523]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.26 UnionType Class

Represents a union type

Inheritance Hierarchy
System.Object
TwinCAT.Ads.TypeSystem.DataType [1349]
TwinCAT.Ads.TypeSystem.UnionType

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

public sealed class UnionType : DataType,
    IUnionType, IDataType, IBitSize

The UnionType type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType (Inherited from DataType.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the DataType in bits. (Inherited from DataType.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from DataType.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from DataType.)</td>
</tr>
<tr>
<td>Fields</td>
<td>Gets a read only collection of the Fields of the IUnionType.</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the IDataType (Namespace + Name) (Inherited from DataType.)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the DataType (Inherited from DataType.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this IDataType is a bit mapping Type (Inherited from DataType.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType.)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this IDataType is a container type (Inherited from DataType.)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this IDataType is a pointer type (Inherited from DataType.)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IDataType is primitive (Inherited from DataType.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this IDataType is a reference type (Inherited from DataType.)</td>
</tr>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from DataType.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from DataType.)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the IDataType exists. (Inherited from DataType.)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the Size of the DataType in Bytes or bits. (Inherited from DataType.)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DataType.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.Ads.TypeSystem Namespace

### 6.7.26.1 UnionType Properties

The `UnionType` type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the <code>IDataType</code> (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>DataType</code> in bits. (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes. (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>Fields</td>
<td>Gets a read only collection of the Fields of the <code>IUnionType</code>.</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the Data Type (Namespace + Name)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the Data Type (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a bit mapping Type.</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0).</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a container type.</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a pointer type.</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this <code>IDataType</code> is primitive.</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a reference type.</td>
</tr>
<tr>
<td>ManagedType</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from <code>DataType</code>).</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the <code>IDataType</code> exists. (Inherited from <code>DataType</code>).</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>Gets the Size of the <strong>DataType</strong> in Bytes or bits. (Inherited from <strong>DataType</strong>.)</td>
</tr>
</tbody>
</table>

### Reference
- **UnionType Class**
- **TwinCAT.Ads.TypeSystem Namespace**

### 6.7.26.1.1 UnionType.Fields Property

Gets a read only collection of the **Fields** of the **IUnionType**.

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 6.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
public IFieldCollection Fields { get; }

**Property Value**

Type: **IFieldCollection**

**Implements**

**IUnionType.Fields**

### Reference
- **UnionType Class**
- **TwinCAT.Ads.TypeSystem Namespace**

### 6.7.26.2 UnionType Methods

The **UnionType** type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object.</strong>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <strong>Object.</strong>)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object.</strong>)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <strong>String</strong> that represents this instance. (Inherited from <strong>DataType</strong>.)</td>
</tr>
</tbody>
</table>
### WStringType Class

Represents an Unicode string (Wide string)

#### Inheritance Hierarchy

- System.Object
  - TwinCAT.Ads.TypeSystem.DataType
    - TwinCAT.Ads.TypeSystem.WStringType

**Namespace:** TwinCAT.Ads.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public sealed class WStringType : DataType, IStringType, IDataType, IBitSize
```

The WStringType type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the <code>IDataType</code> (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>DataType</code> in bits. (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>Encoding</td>
<td>Gets the encoding of the String (Encoding.Default (STRING) or Encoding.UNICODE (WSTRING))</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the <code>IDataType</code> (Namespace + Name) (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the DataType (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a bit mapping Type (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a container type (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>IsFixedLength</td>
<td>Gets a value indicating whether the string is of fixed length.</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the <code>IDatatype</code> (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>Datatype</code> in bits (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the aligned size of the Type/Instance in Bytes (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment. (Inherited from <code>DataType</code>)</td>
</tr>
<tr>
<td>Encoding</td>
<td>Gets the encoding of the String (Encoding.Default (STRING) or Encoding.UNICODE (WSTRING))</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FullName [1355]</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Id [1355]</td>
<td>Gets the ID of the DataType (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsBitType [1356]</td>
<td>Gets a value indicating whether this IDataType [1986] is a bit mapping Type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsByteAligned [1356]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsContainer [1357]</td>
<td>Gets a value indicating whether this IDataType [1986] is a container type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsFixedLength [1540]</td>
<td>Gets a value indicating whether the string is of fixed length.</td>
</tr>
<tr>
<td>IsPointer [1357]</td>
<td>Gets a value indicating whether this IDataType [1986] is a pointer type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsPrimitive [1358]</td>
<td>Gets a value indicating whether this IDataType [1986] is primitive (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>IsReference [1359]</td>
<td>Gets a value indicating whether this IDataType [1986] is a reference type (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Length [1540]</td>
<td>Gets the number of characters within the string.</td>
</tr>
<tr>
<td>ManagedType [1359]</td>
<td>Gets the corresponding .NET Type if attached. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Name [1360]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Namespace [1360]</td>
<td>Gets the namespace string within the IDataType [1986] exists. (Inherited from DataType [1349].)</td>
</tr>
<tr>
<td>Size [1361]</td>
<td>Gets the Size of the DataType [1349] in Bytes or bits. (Inherited from DataType [1349].)</td>
</tr>
</tbody>
</table>

Reference

WStringType Class [1537]

TwinCAT.Ads.TypeSystem Namespace [1328]

6.7.27.1.1 WStringType.Encoding Property

Gets the encoding of the String (Encoding.Default (STRING) or Encoding.UNICODE (WSTRING))

Namespace: TwinCAT.Ads.TypeSystem [1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Encoding Encoding { get; }
```

Property Value

Type: Encoding
The encoding.
6.7.27.1.2  **WStringType.IsFixedLength Property**

Gets a value indicating whether the string is of fixed length.

**Namespace:**  TwinCAT.Ads.TypeSystem  
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) 
Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public bool IsFixedLength { get; }
```

**Property Value**

Type:  Boolean  
true if this instance is fixed length; otherwise, false.

**Implements**

IStringType.IsFixedLength  
[2157]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

WStringType Class  
[1537]

TwinCAT.Ads.TypeSystem Namespace  
[1328]

6.7.27.1.3  **WStringType.Length Property**

Gets the number of characters within the string.

**Namespace:**  TwinCAT.Ads.TypeSystem  
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) 
Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public int Length { get; }
```

Property Value

Type: Int32
The length.

Implements

IStringType.Length

Reference

WStringType Class

TwinCAT.Ads.TypeSystem Namespace

6.7.27.2 WStringType Methods

The WStringType type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString(...)</td>
<td>Returns a String that represents this instance. (Overrides DataType.ToString.)</td>
</tr>
</tbody>
</table>

Reference

WStringType Class

TwinCAT.Ads.TypeSystem Namespace

6.7.27.2.1 WStringType.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.Ads.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public override string ToString()
```
Return Value

Type: String
A String that represents this instance.

Reference

WStringType Class [1537]
TwinCAT.Ads.TypeSystem Namespace [1328]

6.8 TwinCAT.Ads.ValueAccess Namespace

Root namespace for ADS value access.

Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueAccessMode</td>
<td>Enum ValueAccessMethod</td>
</tr>
</tbody>
</table>

6.8.1 ValueAccessMode Enumeration

Enum ValueAccessMethod

Namespace: TwinCAT.Ads.ValueAccess [1542]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd0ca3e72bc0ea15da1c14

Syntax

C#

public enum ValueAccessMode

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>None / Uninitialized</td>
</tr>
<tr>
<td>IndexGroupOffset</td>
<td>1</td>
<td>Value access via Index Group and Offset Only</td>
</tr>
<tr>
<td>Symbolic</td>
<td>2</td>
<td>Symbolic access via Instance Path only.</td>
</tr>
<tr>
<td>IndexGroupOffsetPreferred</td>
<td>3</td>
<td>Uses IndexGroup IndexOffset Preferred (and Symbolic for Dereferenced Pointers / References)</td>
</tr>
<tr>
<td>Default</td>
<td>2</td>
<td>The Default access mode (Symbolic)</td>
</tr>
</tbody>
</table>

Remarks

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>None/Uninitialized. No Valid mode.</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndexGroupOffset</td>
<td>Communicates over IndexGroup/IndexOffset only. This is the most direct/efficient access into the Process image. The advantage is that, the symbol access is done via 1 ADS round trip. Disadvantages are that not all Symbols can be accessed via IG/IO (e.g. References) and IndexOffsets could be invalid after online changes / PlcProgram downloads. Detection of these events and following invalidation of all changed symbols need to be done within the user application.</td>
</tr>
<tr>
<td>Symbolic</td>
<td>The Symbolic-only mode is the most safe mode to use but needs more time than the IndexGroupOffset. It could need up to 3 ADS round trips (create handle, access value, close handle) but is not influenced by online changes or/PlcProgram downloads.</td>
</tr>
<tr>
<td>IndexGroupOffsetPreferred</td>
<td>This is a mixed access mode. For symbols, where it is possible it uses the IndexGroup/IndexOffset. For others it chooses the Symbolic access.</td>
</tr>
<tr>
<td>Default</td>
<td>The Default-Mode setting if no other ValueAccessMode is specified. This is set to Symbolic.</td>
</tr>
</tbody>
</table>

### Reference

[TwinCAT.Ads.ValueAccess Namespace](#)

## 6.9 TwinCAT.Ams Namespace

### Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmsConfiguration</td>
<td>Static configuration of the Ams Router system.</td>
</tr>
<tr>
<td>AmsServerException</td>
<td>Ams Server Exception class Implements the Exception</td>
</tr>
</tbody>
</table>

### Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmsServerErrorCode</td>
<td>Ams Server Error Codes.</td>
</tr>
</tbody>
</table>

### 6.9.1 AmsConfiguration Class

Static configuration of the Ams Router system.

**Inheritance Hierarchy**

- System.Object
- TwinCAT.Ams.AmsConfiguration

**Namespace:** TwinCAT.Ams [1543]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public static class AmsConfiguration
```

The `AmsConfiguration` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DefaultRouterEndPoint</td>
<td>Gets the default router end point (IPAddress.Loopback, Port 0xBF02)</td>
</tr>
<tr>
<td>RouterAddress</td>
<td>Gets the actually configured router address.</td>
</tr>
<tr>
<td>RouterEndPoint</td>
<td>Gets or sets the default router <code>IPEndPoint</code>.</td>
</tr>
<tr>
<td>RouterPort</td>
<td>Gets actually configured router port.</td>
</tr>
</tbody>
</table>

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT_TCP_PORT</td>
<td>The default TCP port (0xBF02, 48898)</td>
</tr>
</tbody>
</table>

### Remarks

If the router is intended to run separately of the (virtual) system that instantiates the `AdsClient` or `AdsServer` (e.g. in UnitTests), some global static settings must be adopted to enable different network scenerios. In the default case the `AdsServers` and `AdsClients` uses the `Loopback` TCP port `0xBF02` to communicate internally. If the systems of `AdsServer/AdsClient` is split from the router (e.g. running the router isolated in a virtual machine), the Router Endpoint must be set via `RouterEndPoint` setter.

### Reference

[TwinCAT.Ams Namespace](#)
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RouterAddress</td>
<td>Gets the actually configured router address.</td>
</tr>
<tr>
<td>RouterEndPoint</td>
<td>Gets or sets the default router IPEndPoint.</td>
</tr>
<tr>
<td>RouterPort</td>
<td>Gets actually configured router port.</td>
</tr>
</tbody>
</table>

### 6.9.1.1.1 AmsConfiguration.DefaultRouterEndPoint Property

Gets the default router end point (IPAddress.Loopback, Port 0xBF02)

**Namespace:** TwinCAT.Ams

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#
```csharp
public static IPEndPoint DefaultRouterEndPoint { get; }
```

**Property Value**

Type: `IPEndPoint`

The default router end point.

### Reference

AmsConfiguration Class [1543]

TwinCAT.Ams Namespace [1543]

### 6.9.1.1.2 AmsConfiguration.RouterAddress Property

Gets the actually configured router address.

**Namespace:** TwinCAT.Ams

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#
```csharp
public static IPAddress RouterAddress { get; }
```
**Property Value**

Type: IPAddress
The router address.

**Reference**

AmsConfiguration Class [1543]
AmsConfiguration.RouterEndPoint [1546]

---

### 6.9.1.1.3 AmsConfiguration.RouterEndPoint Property

Gets or sets the default router IPEndPoint.

**Namespace:** TwinCAT.Ams [1543]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static IPEndPoint RouterEndPoint { get; set; }
```

**Property Value**

Type: IPEndPoint
The IPEndPoint used by Ams Clients/Servers.

**Remarks**

This setting allows to change the Router Endpoint - which listens by default on IPAddress 127.0.0.1, Port 0xBF02. An application for this could be to move the router to a different (virtual) system, separated from the system running the AdsServer or AdsClient. This endpoint can only be changed process wide and should be done before the first access to the Router occurs (e.g. getting the local NetId).

**ATTENTION:** Bending this Endpoint is not possible with the standard TwinCAT Router. As security feature the TwinCAT Router only accepts Loopback connections to 127.0.0.1, connection requests from the outside will be closed immediately.

The simplistic .NET Core Router implementation in class ‘TwinCAT.Ads.AdsRouter.AmsTcpIpRouter’ from nuget package ‘Beckhoff.TwinCAT.Ads.TcpRouter’ or the ‘Beckhoff.TwinCAT.Ads.AdsRouterConsole’ doesn’t have that restriction and can be used more flexible for use with virtual environments like VirtualMachines or Docker. flexible.

The following sample demonstrates how to configure the (internal) RouterEndPoint to IPAddress 1.2.3.4:42. This must be done before AdsServer or AdsClient are instantiated.

**C#**

```csharp
AmsConfiguration.RouterEndPoint = new IPEndPoint(IPAddress.Parse("1.2.3.4"),42);
```

**Reference**

AmsConfiguration Class [1543]
AmsConfiguration.RouterEndPoint [1546]
6.9.1.4 AmsConfiguration.RouterPort Property

Gets actually configured router port.

**Namespace:** TwinCAT.Ams

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```
public static int RouterPort { get; }
```

**Property Value**

Type: `Int32`

The router port.

**Reference**

AmsConfiguration Class

TwinCAT.Ams Namespace

AmsConfiguration.RouterEndPoint

6.9.1.2 AmsConfiguration Fields

The AmsConfiguration type exposes the following members.

**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT_TCP_PORT</td>
<td>The default TCP port (0xBF02, 48898)</td>
</tr>
</tbody>
</table>

**Reference**

AmsConfiguration Class

TwinCAT.Ams Namespace

6.9.1.2.1 AmsConfiguration.DEFAULT_TCP_PORT Field

The default TCP port (0xBF02, 48898)

**Namespace:** TwinCAT.Ams

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```
public const int DEFAULT_TCP_PORT = 48898
```
Field Value
Type: Int32

Reference
AmsConfiguration Class [1543]
TwinCAT.Ams Namespace [1543]

6.9.2 AmsServerErrorCode Enumeration

Ams Server Error Codes.

Namespace: TwinCAT.Ams [1543]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public enum AmsServerErrorCode

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>None / Uninitialized</td>
</tr>
<tr>
<td>ConnectPortFailed</td>
<td>1</td>
<td>Connecting port failed.</td>
</tr>
<tr>
<td>DisconnectPortFailed</td>
<td>2</td>
<td>Disconnecting port failed.</td>
</tr>
<tr>
<td>ReceiveQueueOverflow</td>
<td>3</td>
<td>ReceiveQueue overflow</td>
</tr>
<tr>
<td>ReceiveNotificationQueueOverflow</td>
<td>4</td>
<td>Receive Notification Queue overflow.</td>
</tr>
</tbody>
</table>

Reference
TwinCAT.Ams Namespace [1543]

6.9.3 AmsServerException Class

Ams Server Exception class Implements the Exception

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.Ams.AmsServerException

Namespace: TwinCAT.Ams [1543]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
[SerializableAttribute]
public class AmsServerException : Exception
The AmsServerException type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmsServerException</td>
<td>Initializes a new instance of the AmsServerException class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined</td>
</tr>
<tr>
<td></td>
<td>information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a</td>
</tr>
<tr>
<td></td>
<td>specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Exception.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root</td>
</tr>
<tr>
<td></td>
<td>cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with</td>
</tr>
<tr>
<td></td>
<td>information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object</td>
</tr>
<tr>
<td></td>
<td>that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
6.9.3.1 AmsServerException Constructor

Initializes a new instance of the AmsServerException class.

**Namespace**: TwinCAT.Ams

**Assembly**: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# protected AmsServerException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)

**Parameters**

- `serializationInfo` Type: System.Runtime.Serialization.SerializationInfo
  The serialization information.
  The streaming context.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>serializationInfo</td>
</tr>
</tbody>
</table>

**Reference**

AmsServerException Class

TwinCAT.Ams Namespace [1543]

6.9.3.2 AmsServerException Properties

The AmsServerException type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
### AmsServerException Methods

The `AmsServerException` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the <code>Exception</code> that is the root cause of one or more subsequent exceptions. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the <code>SerializationInfo</code> with information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

### AmsServerException Events

The `AmsServerException` type exposes the following members.
TwinCAT.Ads Namespaces

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

AmsServerException Class [1548]
TwinCAT.Ams Namespace [1543]

6.10 TwinCAT.PlcOpen Namespace

Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE [1552]</td>
<td>PlcOpen DATE class (32-bit)</td>
</tr>
<tr>
<td>DateBase [1559]</td>
<td>PlcOpen Date base class (32-Bit)</td>
</tr>
<tr>
<td>DT [1570]</td>
<td>PlcOpen DT (DATE_AND_TIME) datatype.</td>
</tr>
<tr>
<td>LTIME [1581]</td>
<td>PlcOpen LTIME class</td>
</tr>
<tr>
<td>LTimeBase [1589]</td>
<td>Time base class</td>
</tr>
<tr>
<td>TIME [1598]</td>
<td>PlcOpen TIME class</td>
</tr>
<tr>
<td>TimeBase [1605]</td>
<td>Base class for PlcOpen Time types.</td>
</tr>
<tr>
<td>TOD [1615]</td>
<td>PLCOpen TimeOfDay class (32-Bit)</td>
</tr>
</tbody>
</table>

Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPlcOpenTimeBase</td>
<td>Interface IPlcOpenType</td>
</tr>
<tr>
<td>IPlcOpenTimeBase.T1, T2 [1579]</td>
<td>Interface IPlcOpenType</td>
</tr>
</tbody>
</table>

6.10.1 DATE Class

PlcOpen DATE class (32-bit)

Inheritance Hierarchy

System.Object
TwinCAT.PlcOpen.DateBase [1559]
TwinCAT.PlcOpen.DATE
**Namespace**: TwinCAT.PlcOpen

**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdcd3e72bc0ea15da1c14

**Syntax**

```csharp
public sealed class DATE : DateBase
```

The DATE type exposes the following members.

**Constructors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>Initializes a new instance of the DATE class.</td>
</tr>
<tr>
<td>DATE(DateTimeOffset)</td>
<td>Initializes a new instance of the DATE class.</td>
</tr>
<tr>
<td>DATE(Int64)</td>
<td>Initializes a new instance of the DATE class.</td>
</tr>
<tr>
<td>DATE(UInt32)</td>
<td>Initializes a new instance of the DATE class.</td>
</tr>
<tr>
<td>DATE(Int32, Int32, Int32)</td>
<td>Initializes a new instance of the DATE class.</td>
</tr>
</tbody>
</table>

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Gets or sets the date value. (Inherited from DateBase.)</td>
</tr>
<tr>
<td>Ticks</td>
<td>Returns the number of ticks that represent the value of this DateBase. (Inherited from DateBase.)</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to this instance. (Inherited from DateBase.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Inherited from DateBase.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse</td>
<td>Parses the specified string to a DATE object.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Overrides Object.ToString.)</td>
</tr>
<tr>
<td>TryParse</td>
<td>Tries to parse the specified string to a DATE object.</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.PlcOpen Namespace
6.10.1.1 DATE Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE()</td>
<td>Initializes a new instance of the DATE [1552] class.</td>
</tr>
<tr>
<td>DATE(DateTimeOffset)</td>
<td>Initializes a new instance of the DATE [1552] class.</td>
</tr>
<tr>
<td>DATE(Int64)</td>
<td>Initializes a new instance of the DATE [1552] class.</td>
</tr>
<tr>
<td>DATE(UInt32)</td>
<td>Initializes a new instance of the DATE [1552] class.</td>
</tr>
<tr>
<td>DATE(Int32, Int32, Int32)</td>
<td>Initializes a new instance of the DATE [1552] class.</td>
</tr>
</tbody>
</table>

Reference

DATE Class [1552]

TwinCAT.PlcOpen Namespace [1552]

6.10.1.1.1 DATE Constructor

Initializes a new instance of the DATE [1552] class.

Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdcdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public DATE()
```

Reference

DATE Class [1552]
DATE Overload [1554]
TwinCAT.PlcOpen Namespace [1552]

6.10.1.1.2 DATE Constructor (DateTimeOffset)

Initializes a new instance of the DATE [1552] class.

Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractsions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdcdca3e72bc0ea15da1c14
Syntax

C#

```csharp
public DATE(
    DateTimeOffset date
)
```

Parameters

date Type: System.DateTimeOffset
The date.

Reference

DATE Class [1552]
DATE Overload [1554]
TwinCAT.PlcOpen Namespace [1552]

6.10.1.1.3 DATE Constructor (Int64)

Initializes a new instance of the DATE [1552] class.

Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public DATE(
    long dateValue
)
```

Parameters

dateValue Type: System.Int64
The date value in PlcOpen Ticks.

Reference

DATE Class [1552]
DATE Overload [1554]
TwinCAT.PlcOpen Namespace [1552]

6.10.1.1.4 DATE Constructor (UInt32)

Initializes a new instance of the DATE [1552] class.

Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public DATE(
    uint dateValue
)
```

Parameters

dateValue Type: System.UInt32
The date value in PlcOpen Ticks.

Reference

DATE Class [1552]
DATE Overload [1554]
TwinCAT.PlcOpen Namespace [1552]

6.10.1.1.5 DATE Constructor (Int32, Int32, Int32)

Initializes a new instance of the DATE [1552] class.

Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public DATE(
    int year,
    int month,
    int day
)
```

Parameters

year Type: System.Int32
The year.

month Type: System.Int32
The month.

day Type: System.Int32
The day.

Reference

DATE Class [1552]
DATE Overload [1554]
TwinCAT.PlcOpen Namespace [1552]

6.10.1.2 DATE Properties

The DATE [1552] type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Gets or sets the date value. (Inherited from DateBase.)</td>
</tr>
<tr>
<td>Ticks</td>
<td>Returns the number of ticks that represent the value of this DateBase. (Inherited from DateBase.)</td>
</tr>
</tbody>
</table>

Reference

DATE Class

TwinCAT.PlcOpen Namespace

6.10.1.3 DATE Methods

The DATE type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to this instance. (Inherited from DateBase.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Inherited from DateBase.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse</td>
<td>Parses the specified string to a DATE object.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Overrides Object.ToString.)</td>
</tr>
<tr>
<td>TryParse</td>
<td>Tries to parse the specified string to a DATE object.</td>
</tr>
</tbody>
</table>

Reference

DATE Class

TwinCAT.PlcOpen Namespace

6.10.1.3.1 DATE.Parse Method

Parses the specified string to a DATE object.

Namespace: TwinCAT.PlcOpen
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static DATE Parse(
    string s
)
```
Parameters

s
Type: System.String
The s.

Return Value

Type: DATE

DATE.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormatException</td>
<td>Cannot parse DATE object!</td>
</tr>
</tbody>
</table>

Reference

DATE Class

TwinCAT.PlcOpen Namespace

6.10.1.3.2 DATE.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.PlcOpen

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

DATE Class

TwinCAT.PlcOpen Namespace

6.10.1.3.3 DATE.TryParse Method

Tries to parse the specified string to a DATE object.

Namespace: TwinCAT.PlcOpen

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public static bool TryParse(
    string s,
    out DATE date
)
```

Parameters

- **s**
  - Type: `System.String`
  - The `s`.

- **date**
  - Type: `TwinCAT.PlcOpen.DATE` [1552]
  - The `date`.

Return Value

- Type: `Boolean`
  - true if XXXX, false otherwise.

Reference

- **DATE Class** [1552]
- **TwinCAT.PlcOpen Namespace** [1552]

6.10.2 DateBase Class

PlcOpen Date base class (32-Bit)

Inheritance Hierarchy

- `System.Object`
- `TwinCAT.PlcOpen.DateBase`
  - `TwinCAT.PlcOpen.DATE` [1552]
  - `TwinCAT.PlcOpen.DT` [1570]

Namespace: `TwinCAT.PlcOpen` [1552]

Assembly: `TwinCAT.Ads.Abstractions` (in `TwinCAT.Ads.Abstractions.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public abstract class DateBase : IPlcOpenTimeBase<DateTimeOffset, uint>,
    IPlcOpenTimeBase
```

The `DateBase` type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="1561" alt="DateBase" /></td>
<td>Initializes a new instance of the <code>DateBase</code> class.</td>
</tr>
<tr>
<td><img src="1561" alt="DateBase" /></td>
<td>Initializes a new instance of the <code>DateBase</code> class.</td>
</tr>
</tbody>
</table>

- **DateBase(DateTime Offset)** [1561]
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateBase(Int64)</td>
<td>Initializes a new instance of the DateBase class.</td>
</tr>
<tr>
<td>DateBase(UInt32)</td>
<td>Initializes a new instance of the DateBase class.</td>
</tr>
</tbody>
</table>

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Gets or sets the date value.</td>
</tr>
<tr>
<td>MarshalSize</td>
<td>Gets the marshal size in bytes.</td>
</tr>
<tr>
<td>Ticks</td>
<td>Returns the number of ticks that represent the value of this DateBase.</td>
</tr>
</tbody>
</table>

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateToValue</td>
<td>Converts the specified DateTime value to PlcOpen Ticks.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to this instance. (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Overrides Object.GetHashCode.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ParseToTicks</td>
<td>Parses the specified PlcOpen Date string to PlcOpen ticks.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ValueToDate(Int64)</td>
<td>Converts the specified PlcOpen dateValue in ticks to a DateTime Object</td>
</tr>
<tr>
<td>ValueToDate(UInt32)</td>
<td>Converts the specified PlcOpen dateValue in ticks to a DateTime Object</td>
</tr>
</tbody>
</table>

#### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plcTimeSeconds</td>
<td>PlcTime Seconds (Ticks)</td>
</tr>
</tbody>
</table>

#### Reference

TwinCAT.PlcOpen Namespace [1552]
### 6.10.2.1 DateBase Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateBase ()</td>
<td>Initializes a new instance of the DateBase class.</td>
</tr>
<tr>
<td>DateBase(DateTimeOffset)</td>
<td>Initializes a new instance of the DateBase class.</td>
</tr>
<tr>
<td>DateBase(Int64)</td>
<td>Initializes a new instance of the DateBase class.</td>
</tr>
<tr>
<td>DateBase(UInt32)</td>
<td>Initializes a new instance of the DateBase class.</td>
</tr>
</tbody>
</table>

#### Reference

- **DateBase Class**
- **TwinCAT.PlcOpen Namespace**

### 6.10.2.1.1 DateBase Constructor

Initializes a new instance of the DateBase class.

**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
protected DateBase()
```

#### Reference

- **DateBase Class**
- **DateBase Overload**
- **TwinCAT.PlcOpen Namespace**

### 6.10.2.1.2 DateBase Constructor (DateTimeOffset)

Initializes a new instance of the DateBase class.

**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
protected DateBase(
    DateTimeOffset date
)
```
**Parameters**

date

Type: `System.DateTimeOffset`

The date.

**Reference**

DateBase Class [1559]

DateBase Overload [1561]

TwinCAT.PlcOpen Namespace [1552]

### 6.10.2.1.3 DateBase Constructor (Int64)

Initializes a new instance of the DateBase [1559] class.

**Namespace:** TwinCAT.PlcOpen [1552]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
protected DateBase(
    long dateValue
)
```

**Parameters**

dateValue

Type: `System.Int64`

The date value in PlcOpen Ticks.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td>The date value in PlcOpen Ticks.</td>
</tr>
</tbody>
</table>

**Reference**

DateBase Class [1559]

DateBase Overload [1561]

TwinCAT.PlcOpen Namespace [1552]

### 6.10.2.1.4 DateBase Constructor (UInt32)

Initializes a new instance of the DateBase [1559] class.

**Namespace:** TwinCAT.PlcOpen [1552]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
### Syntax

**C#**

```csharp
protected DateTime
    (uint dateValue)
```

### Parameters

- **dateValue**
  - Type: `System.UInt32`
  - The date value in PlcOpen Ticks.

### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td></td>
</tr>
</tbody>
</table>

### Reference

- **DataBase Class [1559]**
- **DataBase Overload [1561]**
- **TwinCAT.PlcOpen Namespace [1552]**

### 6.10.2.2 DateBase Properties

The `DateBase [1559]` type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date [1563]</td>
<td>Gets or sets the date value.</td>
</tr>
<tr>
<td>MarshalSize [1564]</td>
<td>Gets the marshal size in bytes.</td>
</tr>
<tr>
<td>Ticks [1564]</td>
<td>Returns the number of ticks that represent the value of this <code>DateBase [1559]</code>.</td>
</tr>
</tbody>
</table>

### Reference

- **DataBase Class [1559]**
- **TwinCAT.PlcOpen Namespace [1552]**

### 6.10.2.2.1 DateBase.Date Property

Gets or sets the date value.

- **Namespace:** TwinCAT.PlcOpen [1552]
- **Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public DateTimeOffset Date { get; }
```

**Property Value**

Type: `DateTimeOffset`
The date.

**Reference**

[DateBase Class [1559]](https://example.com/databasestructure)

[TwinCAT.PlcOpen Namespace [1552]](https://example.com/twincatplcopen)

---

### 6.10.2.2.2 **DateBase.MarshalSize Property**

Gets the marshal size in bytes.

**Namespace:** TwinCAT.PlcOpen [1552]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static int MarshalSize { get; }
```

**Property Value**

Type: `Int32`
Marshal size in bytes.

**Reference**

[DateBase Class [1559]](https://example.com/databasestructure)

[TwinCAT.PlcOpen Namespace [1552]](https://example.com/twincatplcopen)

---

### 6.10.2.2.3 **DateBase.Ticks Property**

Returns the number of ticks that represent the value of this `DateBase` [1559].

**Namespace:** TwinCAT.PlcOpen [1552]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public uint Ticks { get; }
```

**Property Value**

Type: `UInt32`
The ticks.
Implements

IPlcOpenTimeBase.T1, T2..Ticks [1580]

Reference

DateBase Class [1559]

TwinCAT.PlcOpen Namespace [1552]

6.10.2.3 DateBase Methods

The DateBase [1559] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateToValue</td>
<td>Converts the specified DateTime value to PlcOpen Ticks.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to this instance. (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Overrides Object.GetHashCode.).</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ParseToTicks</td>
<td>Parses the specified PlcOpen Date string to PlcOpen ticks.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ValueToDate(Int64)</td>
<td>Converts the specified PlcOpen dateValue in ticks to a DateTime Object</td>
</tr>
<tr>
<td>ValueToDate(UInt32)</td>
<td>Converts the specified PlcOpen dateValue in ticks to a DateTime Object</td>
</tr>
</tbody>
</table>

Reference

DateBase Class [1559]

TwinCAT.PlcOpen Namespace [1552]

6.10.2.3.1 DateBase.DateToValue Method

Converts the specified DateTime value to PlcOpen Ticks.

Namespace: TwinCAT.PlcOpen [1552]  
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public static long DateToValue(
    DateTimeOffset date
)
```

**Parameters**

*date*  
Type: `System.DateTimeOffset`  
The date.

**Return Value**

Type: `Int64`  

**Reference**

[DateBase Class](#)  
[TwinCAT.PlcOpen Namespace](#)

### 6.10.2.3.2 DateBase.Equals Method

Determines whether the specified `Object` is equal to this instance.

**Namespace:**  
[TwinCAT.PlcOpen](#)  
**Assembly:**  
[TwinCAT.Ads.Abstractions.dll](#) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public override bool Equals(
    Object obj
)
```

**Parameters**

*obj*  
Type: `System.Object`  
The object to compare with the current object.

**Return Value**

Type: `Boolean`  
true if the specified `Object` is equal to this instance; otherwise, false.

**Reference**

[DateBase Class](#)  
[TwinCAT.PlcOpen Namespace](#)

### 6.10.2.3.3 DateBase.GetHashCode Method

Gets the HashCode of the Address

---

1566  
Version: 1.1  
TC1000
Namespace: TwinCAT.PlcOpen

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public override int GetHashCode()
```

Return Value

Type: Int32

Reference

Database Class

TwinCAT.PlcOpen Namespace

6.10.2.3.4 Database.ParseToTicks Method

 Parses the specified PlcOpen Date string to PlcOpen ticks.

Namespace: TwinCAT.PlcOpen

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected abstract long ParseToTicks(
    string s
)
```

Parameters

s Type: System.String
The s.

Return Value

Type: Int64

Reference

Database Class

TwinCAT.PlcOpen Namespace
6.10.2.3.5 **DataBase.ValueToDate Method**

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueToDate(Int64)</td>
<td>Converts the specified PlcOpen dateValue in ticks to a DateTime Object</td>
</tr>
<tr>
<td>ValueToDate(UInt32)</td>
<td>Converts the specified PlcOpen dateValue in ticks to a DateTime Object</td>
</tr>
</tbody>
</table>

**Reference**

- **DataBase Class** [1559]
- **TwinCAT.PlcOpen Namespace** [1552]

**DataBase.ValueToDate Method (Int64)**

Converts the specified PlcOpen dateValue in ticks to a DateTime Object

**Namespace:** TwinCAT.PlcOpen [1552]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public static DateTimeOffset ValueToDate(
    long dateValue
)
```

**Parameters**

dateValue  
Type: System.Int64  
The date value.

**Return Value**

Type: DateTimeOffset

**Reference**

- **DataBase Class** [1559]
- **ValueToDate Overload** [1568]
- **TwinCAT.PlcOpen Namespace** [1552]

**DataBase.ValueToDate Method (UInt32)**

Converts the specified PlcOpen dateValue in ticks to a DateTime Object

**Namespace:** TwinCAT.PlcOpen [1552]
**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```
public static DateTimeOffset ValueToDate(
    uint dateValue
)
```

**Parameters**

- **dateValue**
  - Type: System.UInt32
  - The date value.

**Return Value**

Type: DateTimeOffset

**Reference**

- DateBase Class [1559]
- ValueToDate Overload [1568]
- TwinCAT.PlcOpen Namespace [1552]

### 6.10.2.4 DateBase Fields

The **DateBase** [1559] type exposes the following members.

**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plcTimeSeconds</td>
<td>PlcTime Seconds (Ticks)</td>
</tr>
</tbody>
</table>

**Reference**

- DateBase Class [1559]
- TwinCAT.PlcOpen Namespace [1552]

### 6.10.2.4.1 DateBase.plcTimeSeconds Field

PlcTime Seconds (Ticks)

**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```
protected uint plcTimeSeconds
```
Field Value
Type: UInt32

Remarks
Seconds from 1/1/1970 in Local Time Zone (different from UnixTimeSeconds which is UTC!). The is exactly the same value and layout, like it is stored in PlcControl for the DATE and DT type!

Reference
DateBase Class [1559]
TwinCAT.PlcOpen Namespace [1552]

6.10.3 DT Class
PlcOpen DT (DATE_AND_TIME) datatype.

Inheritance Hierarchy
System.Object
    TwinCAT.PlcOpen.DateBase [1559]
    TwinCAT.PlcOpen.DT

Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
public sealed class DT : DateBase

The DT type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>Initializes a new instance of the DT class.</td>
</tr>
<tr>
<td>DT(DateTimeOffset)</td>
<td>Initializes a new instance of the DT class.</td>
</tr>
<tr>
<td>DT(Int64)</td>
<td>Initializes a new instance of the DT class.</td>
</tr>
<tr>
<td>DT(UInt32)</td>
<td>Initializes a new instance of the DT class.</td>
</tr>
<tr>
<td>DT(Int32, Int32, Int32, Int32, Int32)</td>
<td>Initializes a new instance of the DT class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Gets or sets the date value. (Inherited from DateBase.)</td>
</tr>
<tr>
<td>Ticks</td>
<td>Returns the number of ticks that represent the value of this DateBase.</td>
</tr>
</tbody>
</table>
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1566]</td>
<td>Determines whether the specified Object is equal to this instance. (Inherited from Database [1559].)</td>
</tr>
<tr>
<td>GetHashCode [1566]</td>
<td>Gets the HashCode of the Address (Inherited from Database [1559].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse [1575]</td>
<td>Parses the specified string to the DT object.</td>
</tr>
<tr>
<td>ToString [1576]</td>
<td>Returns a String that represents this instance. (Overides Object.ToString.)</td>
</tr>
<tr>
<td>TryParse [1576]</td>
<td>Tries to parse the specified string to a DT object.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.PlcOpen Namespace [1552]

6.10.3.1  DT Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT [1571]</td>
<td>Initializes a new instance of the DT [1570] class.</td>
</tr>
<tr>
<td>DT(DateTimeOffset) [1572]</td>
<td>Initializes a new instance of the DT [1570] class.</td>
</tr>
<tr>
<td>DT(Int64) [1572]</td>
<td>Initializes a new instance of the DT [1570] class.</td>
</tr>
<tr>
<td>DT(UInt32) [1573]</td>
<td>Initializes a new instance of the DT [1570] class.</td>
</tr>
</tbody>
</table>

Reference

DT Class [1570]

TwinCAT.PlcOpen Namespace [1552]
6.10.3.1.2 DT Constructor (DateTimeOffset)

Initializes a new instance of the DT class.

**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public DT()
```

**Parameters**

- `date` Type: System.DateTimeOffset
  The date.

**Reference**

DT Class
DT Overload
TwinCAT.PlcOpen Namespace

6.10.3.1.3 DT Constructor (Int64)

Initializes a new instance of the DT class.

**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public DT(
    long dateValue
)
```
Parameters

dateValue  Type: System.Int64
  The date value in PlcOpen Ticks.

Reference

DT Class [1570]

DT Overload [1571]

TwinCAT.PlcOpen Namespace [1552]

6.10.3.1.4  DT Constructor (UInt32)

Initializes a new instance of the DT [1570] class.

Namespace:  TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public DT(
    uint dateValue
)
```

Parameters

dateValue  Type: System.UInt32
  The date value in PlcOpen Ticks.

Reference

DT Class [1570]

DT Overload [1571]

TwinCAT.PlcOpen Namespace [1552]

6.10.3.1.5  DT Constructor (Int32, Int32, Int32, Int32, Int32, Int32)

Initializes a new instance of the DT [1570] class.

Namespace:  TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public DT(
    int year,
    int month,
    int day,
    int hour,
)
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>year</td>
<td>System.Int32</td>
<td>The year.</td>
</tr>
<tr>
<td>month</td>
<td>System.Int32</td>
<td>The month.</td>
</tr>
<tr>
<td>day</td>
<td>System.Int32</td>
<td>The day.</td>
</tr>
<tr>
<td>hour</td>
<td>System.Int32</td>
<td>The hour.</td>
</tr>
<tr>
<td>minute</td>
<td>System.Int32</td>
<td>The minute.</td>
</tr>
<tr>
<td>second</td>
<td>System.Int32</td>
<td>The second.</td>
</tr>
</tbody>
</table>

Reference

DT Class [1570]
DT Overload [1571]
TwinCAT.PlcOpen Namespace [1552]

6.10.3.2 DT Properties

The DT [1570] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Gets or sets the date value. (Inherited from DateBase [1559].)</td>
</tr>
<tr>
<td>Ticks</td>
<td>Returns the number of ticks that represent the value of this DateBase</td>
</tr>
<tr>
<td></td>
<td>[1559]. (Inherited from DateBase [1559].)</td>
</tr>
</tbody>
</table>

Reference

DT Class [1570]
TwinCAT.PlcOpen Namespace [1552]

6.10.3.3 DT Methods

The DT [1570] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to this instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DateBase [1559].)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Inherited from Database)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object)</td>
</tr>
<tr>
<td>Parse [1575]</td>
<td>Parses the specified string to the DT[1570] object.</td>
</tr>
<tr>
<td>ToString [1576]</td>
<td>Returns a String that represents this instance. (Overrides Object.ToString.)</td>
</tr>
<tr>
<td>TryParse [1576]</td>
<td>Tries to parse the specified string to a DT[1570] object.</td>
</tr>
</tbody>
</table>

### Reference

DT Class [1570]

TwinCAT.PlcOpen Namespace [1552]

### 6.10.3.3.1 DT.Parse Method

Parses the specified string to the DT[1570] object.

**Namespace:** TwinCAT.PlcOpen [1552]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static DT Parse(
    string s)
```

**Parameters**

- **s** Type: System.String
  The s.

**Return Value**

Type: DT[1570]

DT.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormatException</td>
<td>Cannot parse DT object!</td>
</tr>
</tbody>
</table>

**Reference**

DT Class [1570]

TwinCAT.PlcOpen Namespace [1552]
6.10.3.3.2 DT.ToString Method

Returns a `String` that represents this instance.

**Namespace:** TwinCAT.PlcOpen [1552]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public override string ToString()
```

**Return Value**

Type: `String`  
A `String` that represents this instance.

**Reference**

DT Class [1570]  
TwinCAT.PlcOpen Namespace [1552]

6.10.3.3.3 DT.TryParse Method

Tries to parse the specified string to a `DT` object.

**Namespace:** TwinCAT.PlcOpen [1552]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public static bool TryParse(
    string s,
    out DT dt
)
```

**Parameters**

- `s`  
  Type: `System.String`  
The s.

- `dt`  
  Type: TwinCAT.PlcOpen.DT [1570].  
The dt.

**Return Value**

Type: `Boolean`  
true if XXXX, false otherwise.

**Reference**

DT Class [1570]  
TwinCAT.PlcOpen Namespace [1552]
6.10.4   IPlcOpenTimeBase Interface

Interface IPlcOpenType

Namespace:  TwinCAT.PlcOpen [1552]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IPlcOpenTimeBase
```

The IPlcOpenTimeBase type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagedValueType</td>
<td>Gets the type of the underlying human readable type (DateTime or TimeSpan)</td>
</tr>
<tr>
<td>TicksValueType</td>
<td>Gets the type of the underlying ticks resolution (uint32 or uint64)</td>
</tr>
<tr>
<td>UntypedValue</td>
<td>Returns the 'Value' as object type.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.PlcOpen Namespace [1552]

6.10.4.1  IPlcOpenTimeBase Properties

The IPlcOpenTimeBase [1577] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagedValueType</td>
<td>Gets the type of the underlying human readable type (DateTime or TimeSpan)</td>
</tr>
<tr>
<td>TicksValueType</td>
<td>Gets the type of the underlying ticks resolution (uint32 or uint64)</td>
</tr>
<tr>
<td>UntypedValue</td>
<td>Returns the 'Value' as object type.</td>
</tr>
</tbody>
</table>

Reference

IPlcOpenTimeBase Interface [1577]

TwinCAT.PlcOpen Namespace [1552]

6.10.4.1.1  IPlcOpenTimeBase.ManagedValueType Property

Gets the type of the underlying human readable type (DateTime or TimeSpan)
**Namespace:**  TwinCAT.PlcOpen [](#)

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

`Type ManagedValueType { get; }`

**Property Value**

Type: `Type`  
The type of the managed value.

**Reference**

IPlcOpenTimeBase Interface [](#)

TwinCAT.PlcOpen Namespace [](#)

6.10.4.1.2  **IPlcOpenTimeBase.TicksValueType Property**

Gets the type of the underlying ticks resolution (uint32 or uint64)

**Namespace:**  TwinCAT.PlcOpen [](#)

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

`Type TicksValueType { get; }`

**Property Value**

Type: `Type`  
The type of the ticks value.

**Reference**

IPlcOpenTimeBase Interface [](#)

TwinCAT.PlcOpen Namespace [](#)

6.10.4.1.3  **IPlcOpenTimeBase.UntypedValue Property**

Returns the 'Value' as object type.

**Namespace:**  TwinCAT.PlcOpen [](#)

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

`Object UntypedValue { get; }`
6.10.5  IPlcOpenTimeBase.T1, T2. Interface

The IPlcOpenTimeBase.T1, T2. type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagedValueType</td>
<td>Gets the type of the underlying human readable type (DateTime or TimeSpan) (Inherited from IPlcOpenTimeBase [1577]).</td>
</tr>
<tr>
<td>Ticks [1580]</td>
<td>Returns the number of ticks that represent the value of this IPlcOpenTimeBase.T1, T2. (uint32 or uint64).</td>
</tr>
<tr>
<td>TicksValueType</td>
<td>Gets the type of the underlying ticks resolution (uint32 or uint64) (Inherited from IPlcOpenTimeBase [1577]).</td>
</tr>
<tr>
<td>UntypedValue</td>
<td>Returns the 'Value' as object type. (Inherited from IPlcOpenTimeBase [1577]).</td>
</tr>
<tr>
<td>Value [1580]</td>
<td>Returns the value of this IPlcOpenTimeBase.T1, T2. as Managed base type (DateTime or Timespan).</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.PlcOpen Namespace [1552]

TwinCAT.PlcOpen.IPlcOpenTimeBase [1577]

6.10.5.1  IPlcOpenTimeBase.T1, T2. Properties

The IPlcOpenTimeBase.T1, T2. [1579] generic type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagedValueType</td>
<td>Gets the type of the underlying human readable type (DateTime or Timespan)</td>
</tr>
<tr>
<td>Ticks</td>
<td>Returns the number of ticks that represent the value of this IPlcOpenTimeBase.T1, T2. (Inherited from IPlcOpenTimeBase.)</td>
</tr>
<tr>
<td>TicksValueType</td>
<td>Gets the type of the underlying ticks resolution (uint32 or uint64) (Inherited from IPlcOpenTimeBase.)</td>
</tr>
<tr>
<td>UntypedValue</td>
<td>Returns the 'Value' as object type. (Inherited from IPlcOpenTimeBase.)</td>
</tr>
<tr>
<td>Value</td>
<td>Returns the value of this IPlcOpenTimeBase.T1, T2. as Managed base type (DateTime or Timespan)</td>
</tr>
</tbody>
</table>

### Reference

IPlcOpenTimeBase.T1, T2. Interface

TwinCAT.PlcOpen Namespace

### 6.10.5.1.1 IPlcOpenTimeBase.T1, T2..Ticks Property

Returns the number of ticks that represent the value of this IPlcOpenTimeBase.T1, T2. (uint32 or uint64).

**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
T2 Ticks { get; }
```

**Property Value**

Type: T2

The ticks.

### Reference

IPlcOpenTimeBase.T1, T2. Interface

TwinCAT.PlcOpen Namespace

### 6.10.5.1.2 IPlcOpenTimeBase.T1, T2..Value Property

Returns the value of this IPlcOpenTimeBase.T1, T2. as Managed base type (DateTime or Timespan)

**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

T1 Value { get; }

Property Value

Type: T1

The value.

Reference

IPlcOpenTimeBase.T1, T2. Interface

TwinCAT.PlcOpen Namespace

6.10.6 LTIME Class

PlcOpen LTIME class

Inheritance Hierarchy

System.Object
  TwinCAT.PlcOpen.LTimeBase
  TwinCAT.PlcOpen.LTIME

Namespace: TwinCAT.PlcOpen
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public sealed class LTIME : LTimeBase

The LTIME type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTIME</td>
<td>Initializes a new instance of the TIME class.</td>
</tr>
<tr>
<td>LTIME(Int64)</td>
<td>Initializes a new instance of the TIME class.</td>
</tr>
<tr>
<td>LTIME(TimeSpan)</td>
<td>Initializes a new instance of the TIME class.</td>
</tr>
<tr>
<td>LTIME(UInt64)</td>
<td>Initializes a new instance of the TIME class.</td>
</tr>
<tr>
<td>LTIME(Int32, Int32, Int32)</td>
<td>Initializes a new instance of the LTIME class.</td>
</tr>
<tr>
<td>LTIME(Int32, Int32, Int32, Int32)</td>
<td>Initializes a new instance of the LTIME class.</td>
</tr>
</tbody>
</table>
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticks</td>
<td>Returns the number of ticks that represent the value of this _LTimeBase [1589]. (Inherited from _LTimeBase [1589].)</td>
</tr>
<tr>
<td>Time</td>
<td>Gets or the time value. (Inherited from _LTimeBase [1589].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to this instance. (Inherited from _LTimeBase [1589].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Inherited from _LTimeBase [1589].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse</td>
<td>Parses the specified string to a LTIME object.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Overrides Object.ToString.)</td>
</tr>
<tr>
<td>TryParse</td>
<td>Tries to parse the string to a LTIME object.</td>
</tr>
</tbody>
</table>

Remarks

This class has an internal requestation of an UINT64 (8 Bytes)

Reference

TwinCAT.PlcOpen Namespace \[1552\]

6.10.6.1 LTIME Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTIME</td>
<td>Initializes a new instance of the TIME [1598] class.</td>
</tr>
<tr>
<td>LTIME(Int64)</td>
<td>Initializes a new instance of the TIME [1598] class.</td>
</tr>
<tr>
<td>LTIME(TimeSpan)</td>
<td>Initializes a new instance of the TIME [1598] class.</td>
</tr>
<tr>
<td>LTIME(UInt64)</td>
<td>Initializes a new instance of the TIME [1598] class.</td>
</tr>
<tr>
<td>LTIME(Int32, Int32, Int32)</td>
<td>Initializes a new instance of the LTIME [1581] class.</td>
</tr>
<tr>
<td>LTIME(Int32, Int32, Int32)</td>
<td>Initializes a new instance of the LTIME [1581] class.</td>
</tr>
<tr>
<td>LTIME(Int32, Int32, Int32)</td>
<td>Initializes a new instance of the LTIME [1581] class.</td>
</tr>
</tbody>
</table>
6.10.6.1.1 LTIME Constructor

Initializes a new instance of the TIME class.

Namespace: TwinCAT.PlcOpen
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public LTIME()
6.10.6.1.3  LTIME Constructor (TimeSpan)

Initializes a new instance of the TIME class.

Namespace:  TwinCAT.PlcOpen
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public LTIME(
    TimeSpan time
)
```

Parameters

time  Type: System.TimeSpan
The time.

Reference

LTIME Class
LTIME Overload
TwinCAT.PlcOpen Namespace

6.10.6.1.4  LTIME Constructor (UInt64)

Initializes a new instance of the TIME class.

Namespace:  TwinCAT.PlcOpen
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public LTIME(
    ulong timeValue
)
```

Parameters

timeValue  Type: System.UInt64
The time value.

Reference

LTIME Class
LTIME Overload
TwinCAT.PlcOpen Namespace
6.10.6.1.5 LTIME Constructor (Int32, Int32, Int32)

Initializes a new instance of the LTIME [1581] class.

**Namespace:** TwinCAT.PlcOpen [1552]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public LTIME(
    int seconds,
    int milliseconds,
    int microseconds
)
```

**Parameters**

- **seconds**  
  Type: System.Int32  
  The seconds.
- **milliseconds**  
  Type: System.Int32  
  The milliseconds.
- **microseconds**  
  Type: System.Int32  
  The microseconds.

**Reference**

- LTIME Class [1581]
- LTIME Overload [1582]
- TwinCAT.PlcOpen Namespace [1552]

6.10.6.1.6 LTIME Constructor (Int32, Int32, Int32, Int32)

Initializes a new instance of the LTIME [1581] class.

**Namespace:** TwinCAT.PlcOpen [1552]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public LTIME(
    int seconds,
    int milliseconds,
    int microseconds,
    int nanoseconds
)
```

**Parameters**

- **seconds**  
  Type: System.Int32  
  The seconds.
- **milliseconds**  
  Type: System.Int32  
  The milliseconds.
**microseconds**  
Type: System.Int32  
The microseconds.

**nanoseconds**  
Type: System.Int32  
The nanoseconds.

**Reference**

**LTIME Class [1581]**

**LTIME Overload [1582]**

**TwinCAT.PlcOpen Namespace [1552]**

### 6.10.6.1.7 LTIME Constructor (Int32, Int32, Int32, Int32, Int32, Int32, Int32)

Initializes a new instance of the LTIME [1581] class.

**Namespace:** TwinCAT.PlcOpen [1552]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public LTIME(
    int days,
    int hours,
    int minutes,
    int seconds,
    int milliseconds,
    int microseconds,
    int nanoseconds
)
```

**Parameters**

**days**  
Type: System.Int32  
The days.

**hours**  
Type: System.Int32  
The hours.

**minutes**  
Type: System.Int32  
The minutes.

**seconds**  
Type: System.Int32  
The seconds.

**milliseconds**  
Type: System.Int32  
The milliseconds.

**microseconds**  
Type: System.Int32  
The microseconds.

**nanoseconds**  
Type: System.Int32  
The nanoseconds.

**Reference**

**LTIME Class [1581]**

**LTIME Overload [1582]**

**TwinCAT.PlcOpen Namespace [1552]**
### 6.10.6.2 LTIME Properties

The LTIME[1581] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticks[1593]</td>
<td>Returns the number of ticks that represent the value of this LTimeBase[1589]. (Inherited from LTimeBase[1589].)</td>
</tr>
<tr>
<td>Time[1593]</td>
<td>Gets or the time value. (Inherited from LTimeBase[1589].)</td>
</tr>
</tbody>
</table>

#### Reference

LTIME Class[1581]

TwinCAT.PlcOpen Namespace[1552]

### 6.10.6.3 LTIME Methods

The LTIME[1581] type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals[1594]</td>
<td>Determines whether the specified Object is equal to this instance. (Inherited from LTimeBase[1589].)</td>
</tr>
<tr>
<td>GetHashCode[1595]</td>
<td>Gets the HashCode of the Address (Inherited from LTimeBase[1589].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse[1587]</td>
<td>Parses the specified string to a LTIME[1581] object.</td>
</tr>
<tr>
<td>ToString[1588]</td>
<td>Returns a String that represents this instance. (Overrides Object.ToString.)</td>
</tr>
<tr>
<td>TryParse[1588]</td>
<td>Tries to parse the string to a LTIME[1581] object.</td>
</tr>
</tbody>
</table>

#### Reference

LTIME Class[1581]

TwinCAT.PlcOpen Namespace[1552]

### 6.10.6.3.1 LTIME.Parse Method

Parses the specified string to a LTIME[1581] object.

**Namespace:** TwinCAT.PlcOpen[1552]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public static LTIME Parse(
    string str
)
```

Parameters

<table>
<thead>
<tr>
<th>str</th>
<th>Type: System.String</th>
</tr>
</thead>
<tbody>
<tr>
<td>The string.</td>
<td></td>
</tr>
</tbody>
</table>

Return Value

Type: LTIME

LTIME.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormatException</td>
<td>Cannot create TIME DataType!</td>
</tr>
</tbody>
</table>

Reference

LTIME Class [1581]

TwinCAT.PlcOpen Namespace [1552]

6.10.6.3.2 LTIME.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.PlcOpen [1552]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public override string ToString()
```

Return Value

Type: String

A String that represents this instance.

Reference

LTIME Class [1581]

TwinCAT.PlcOpen Namespace [1552]

6.10.6.3.3 LTIME.TryParse Method

Tries to parse the string to a LTIME object.
TwinCAT.Ads Namespaces

Namespace:  TwinCAT.PlcOpen  [1552]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static bool TryParse(
    string str,
    out LTIME ret
)

Parameters

str  Type: System.String
     The string.
ret  Type: TwinCAT.PlcOpen.LTIME  [1581].
     The ret.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

LTIME Class  [1581]
TwinCAT.PlcOpen Namespace  [1552]

6.10.7 LTimeBase Class

Time base class

Inheritance Hierarchy

System.Object
   TwinCAT.PlcOpen.LTimeBase
      TwinCAT.PlcOpen.LTIME  [1581]

Namespace:  TwinCAT.PlcOpen  [1552]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public abstract class LTimeBase : IPlcOpenTimeBase<Timespan, ulong>, IPlcOpenTimeBase

The LTimeBase type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
**Name** | **Description**
--- | ---
LTimeBase(UInt64) | Initializes a new instance of the TimeBase [1605] class.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MarshalSize [1592]</td>
<td>Gets the marshal size in bytes.</td>
</tr>
<tr>
<td>Ticks [1593]</td>
<td>Returns the number of ticks that represent the value of this LTimeBase.</td>
</tr>
<tr>
<td>Time [1593]</td>
<td>Gets or the time value.</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1594]</td>
<td>Determines whether the specified Object is equal to this instance. (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TimeToValue [1595]</td>
<td>Converts the Timespan to PlcOpen ticks.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ValueToTime(Int64) [1596]</td>
<td>Converts the timeValue (PlcOpen ticks) to TimeSpan</td>
</tr>
<tr>
<td>ValueToTime(UInt64) [1597]</td>
<td>Converts the timeValue (PlcOpen ticks) to TimeSpan</td>
</tr>
</tbody>
</table>

**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>internalTimeValue [1597]</td>
<td>The internal time value</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.PlcOpen Namespace [1552]
6.10.7.1  LTimeBase Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTimeBase.</td>
<td>Initializes a new instance of the TimeBase class.</td>
</tr>
<tr>
<td>LTimeBase(UInt64)</td>
<td>Initializes a new instance of the TimeBase class.</td>
</tr>
</tbody>
</table>

Reference

LTimeBase Class [1589]
TwinCAT.PlcOpen Namespace [1552]

6.10.7.1.1  LTimeBase Constructor

Initializes a new instance of the TimeBase class.

Namespace:  TwinCAT.PlcOpen [1552]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  
protected LTimeBase()  

Reference

LTimeBase Class [1589]
LTimeBase Overload [1591]
TwinCAT.PlcOpen Namespace [1552]

6.10.7.1.2  LTimeBase Constructor (UInt64)

Initializes a new instance of the TimeBase class.

Namespace:  TwinCAT.PlcOpen [1552]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  
protected LTimeBase(  
    ulong timeValue  
  )  

Parameters

timeValue  
Type: System.UInt64  
The time value.
Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reference

LTimeBase Class [1589]
LTimeBase Overload [1591]
TwinCAT.PlcOpen Namespace [1552]

6.10.7.2 LTimeBase Properties

The LTimeBase [1589] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MarshalSize</td>
<td>Gets the marshal size in bytes.</td>
</tr>
<tr>
<td>Ticks</td>
<td>Returns the number of ticks that represent the</td>
</tr>
<tr>
<td></td>
<td>value of this LTimeBase [1589].</td>
</tr>
<tr>
<td>Time</td>
<td>Gets or the time value.</td>
</tr>
</tbody>
</table>

Reference

LTimeBase Class [1589]
TwinCAT.PlcOpen Namespace [1552]

6.10.7.2.1 LTimeBase.MarshalSize Property

Gets the marshal size in bytes.

Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static int MarshalSize { get; }
```

Property Value

Type: Int32
Marshal size in bytes.
6.10.7.2.2  **LTimeBase.Ticks Property**

Returns the number of ticks that represent the value of this [LTimeBase](#).  

**Namespace:**  TwinCAT.PlcOpen  
**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ulong Ticks { get; }
```

**Property Value**

Type:  UInt64  
The ticks.

**Implements**

IPlcOpenTimeBase.T1, T2..Ticks [1580]

---

6.10.7.2.3  **LTimeBase.Time Property**

Gets or the time value.

**Namespace:**  TwinCAT.PlcOpen  
**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public TimeSpan Time { get; }
```

**Property Value**

Type:  TimeSpan  
The time.

**Reference**

LTimeBase Class [1589]  
TwinCAT.PlcOpen Namespace [1552]
6.10.7.3  LTimeBase Methods

The LTimeBase [1589] type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1594]</td>
<td>Determines whether the specified Object is equal to this instance. (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TimeToValue [1595]</td>
<td>Converts the Timespan to PlcOpen ticks.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ValueToTime(Int64) [1596]</td>
<td>Converts the timeValue (PlcOpen ticks) to TimeSpan</td>
</tr>
<tr>
<td>ValueToTime(UInt64)</td>
<td>Converts the timeValue (PlcOpen ticks) to TimeSpan</td>
</tr>
</tbody>
</table>

**Reference**

LTimeBase Class [1589]

TwinCAT.PlcOpen Namespace [1552]

### 6.10.7.3.1  LTimeBase.Equals Method

Determines whether the specified Object is equal to this instance.

**Namespace:** TwinCAT.PlcOpen [1552]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public override bool Equals(
    Object obj
)
```

**Parameters**

- **obj**
  - Type: System.Object
  - The object to compare with the current object.
Return Value

Type: Boolean
true if the specified Object is equal to this instance; otherwise, false.

Reference

LTimeBase Class [1589]
TwinCAT.PlcOpen Namespace [1552]

6.10.7.3.2 LTimeBase.GetHashCode Method

 Gets the HashCode of the Address

Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public override int GetHashCode();

Return Value

Type: Int32

Reference

LTimeBase Class [1589]
TwinCAT.PlcOpen Namespace [1552]

6.10.7.3.3 LTimeBase.TimeToValue Method

 Converts the Timespan to PlcOpen ticks.

Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static ulong TimeToValue(
    TimeSpan time
)

Parameters

time Type: System.TimeSpan
The time.
**Return Value**
Type: `UInt64`

**Reference**
- `LTimeBase Class [1589]`
- `TwinCAT.PlcOpen Namespace [1552]`

### 6.10.7.3.4 LTimeBase.ValueToTime Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueToTime(Int64) [1596]</td>
<td>Converts the timeValue (PlcOpen ticks) to TimeSpan</td>
</tr>
<tr>
<td>ValueToTime(UInt64) [1597]</td>
<td>Converts the timeValue (PlcOpen ticks) to TimeSpan</td>
</tr>
</tbody>
</table>

**Reference**
- `LTimeBase Class [1589]`
- `TwinCAT.PlcOpen Namespace [1552]`

### LTimeBase.ValueToTime Method (Int64)
Converts the timeValue (PlcOpen ticks) to TimeSpan

**Namespace:** `TwinCAT.PlcOpen [1552]`

**Assembly:** `TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public static TimeSpan ValueToTime(
    long nanoseconds
)
```

**Parameters**

- `nanoseconds` Type: `System.Int64`
  The time value.

**Return Value**
Type: `TimeSpan`

**Reference**
- `LTimeBase Class [1589]`
**ValueToTime Overload** [1596]

**TwinCAT.PlcOpen Namespace** [1552]

---

**LTimeBase.ValueToTime Method (UInt64)**

Converts the timeValue (PlcOpen ticks) to TimeSpan

**Namespace:** TwinCAT.PlcOpen [1552]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public static TimeSpan ValueToTime(
    ulong nanoseconds
)
```

**Parameters**

- **nanoseconds**
  
  **Type:** System.UInt64
  
  The time value.

**Return Value**

Type: **TimeSpan**

---

**Reference**

- LTimeBase Class [1589]
- ValueToTime Overload [1596]
- TwinCAT.PlcOpen Namespace [1552]

---

**6.10.7.4 LTimeBase Fields**

The LTimeBase [1589] type exposes the following members.

**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![internalTimeValue icon]</td>
<td>internalTimeValue [1597]</td>
</tr>
</tbody>
</table>

**The internal time value**

---

**Reference**

- LTimeBase Class [1589]
- TwinCAT.PlcOpen Namespace [1552]

---

**6.10.7.4.1 LTimeBase.internalTimeValue Field**

The internal time value
TwinCAT.Ads Namespaces

Namespace: TwinCAT.PlcOpen

Schema 6.10.8 TIME Class

PlcOpen TIME class

Inheritance Hierarchy

System.Object
  TwinCAT.PlcOpen.TimeBase
  TwinCAT.PlcOpen.TIME

Namespace: TwinCAT.PlcOpen

Field Value

Type: UInt64

Reference

LTimeBase Class

TwinCAT.PlcOpen Namespace

Syntax

C# protected ulong internalTimeValue

The TIME type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME() [1600]</td>
<td>Initializes a new instance of the TIME class.</td>
</tr>
<tr>
<td>TIME(Int64) [1600]</td>
<td>Initializes a new instance of the TIME class.</td>
</tr>
<tr>
<td>TIME(TimeSpan) [1601]</td>
<td>Initializes a new instance of the TIME class.</td>
</tr>
<tr>
<td>TIME(UInt32) [1601]</td>
<td>Initializes a new instance of the TIME class.</td>
</tr>
<tr>
<td>TIME(Int32, Int32) [1602]</td>
<td>Initializes a new instance of the TIME class.</td>
</tr>
<tr>
<td>TIME(Int32, Int32, Int32) [1602]</td>
<td>Initializes a new instance of the TIME class.</td>
</tr>
</tbody>
</table>
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Time Value</td>
<td>Gets the internal time value. (Inherited from TimeBase [1605].)</td>
</tr>
<tr>
<td>Ticks [1610]</td>
<td>Returns the number of ticks that represent the value of this TimeBase [1605] (uint32 or uint64). (Inherited from TimeBase [1605].)</td>
</tr>
<tr>
<td>Time [1611]</td>
<td>Gets the time value. (Inherited from TimeBase [1605].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1612]</td>
<td>Determines whether the specified Object is equal to this instance. (Inherited from TimeBase [1605].)</td>
</tr>
<tr>
<td>GetHashCode [1612]</td>
<td>Gets the HashCode of the Address (Inherited from TimeBase [1605].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse [1604]</td>
<td>Parses the specified string to a TIME object.</td>
</tr>
<tr>
<td>Tostring [1604]</td>
<td>Returns a String that represents this instance. (Overrides Object.ToString.)</td>
</tr>
<tr>
<td>TryParse [1605]</td>
<td>Tries to parse the TIME object from string.</td>
</tr>
</tbody>
</table>

Remarks

This class has an internal representation of an UINT32 (4 bytes).

Reference

TwinCAT.PlcOpen Namespace [1552]

6.10.8.1 TIME Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME [1600]</td>
<td>Initializes a new instance of the TIME [1598] class.</td>
</tr>
<tr>
<td>TIME(Int64) [1600]</td>
<td>Initializes a new instance of the TIME [1598] class.</td>
</tr>
<tr>
<td>TIME(TimeSpan) [1601]</td>
<td>Initializes a new instance of the TIME [1598] class.</td>
</tr>
<tr>
<td>TIME(UInt32) [1601]</td>
<td>Initializes a new instance of the TIME [1598] class.</td>
</tr>
<tr>
<td>TIME(Int32, Int32) [1602]</td>
<td>Initializes a new instance of the TIME [1598] class.</td>
</tr>
</tbody>
</table>
### 6.10.8.1.1 TIME Constructor

Initializes a new instance of the **TIME** class.

**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public TIME()
```

### 6.10.8.1.2 TIME Constructor (Int64)

Initializes a new instance of the **TIME** class.

**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public TIME(
    long timeValue
)
```

**Parameters**

- **timeValue**  
  Type: System.Int64  
  The time value.

### Reference

TIME Class [1598]

TwinCAT.PlcOpen Namespace [1552]
6.10.8.1.3 TIME Constructor (TimeSpan)

Initializes a new instance of the TIME class.

Namespace: TwinCAT.PlcOpen
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public TIME(
    TimeSpan time
)
```

Parameters

time Type: System.TimeSpan
The time.

Reference

TIME Class
TIME Overload
TwinCAT.PlcOpen Namespace

6.10.8.1.4 TIME Constructor (UInt32)

Initializes a new instance of the TIME class.

Namespace: TwinCAT.PlcOpen
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public TIME(
    uint timeValue
)
```

Parameters

timeValue Type: System.UInt32
The time value.

Reference

TIME Class
TIME Overload
TwinCAT.PlcOpen Namespace
6.10.8.1.5 TIME Constructor (Int32, Int32)
Initializes a new instance of the TIME class.

Namespace: TwinCAT.PlcOpen
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C# public TIME(int seconds, int milliseconds)

Parameters

seconds Type: System.Int32
The seconds.
milliseconds Type: System.Int32
The milliseconds.

Reference

TIME Class
TIME Overload
TwinCAT.PlcOpen Namespace

6.10.8.1.6 TIME Constructor (Int32, Int32, Int32, Int32, Int32)
Initializes a new instance of the TIME class.

Namespace: TwinCAT.PlcOpen
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C# public TIME(int days, int hours, int minutes, int seconds, int milliseconds)

Parameters

days Type: System.Int32
The days.
hours Type: System.Int32
The hours.
minutes Type: System.Int32
The minutes.
seconds Type: System.Int32
The seconds.
milliseconds Type: System.Int32
The milliseconds.

Reference
TIME Class [1598]
TIME Overload [1599]
TwinCAT.PlcOpen Namespace [1552]

6.10.8.2 TIME Properties

The TIME [1598] type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalTimeValue</td>
<td>Gets the internal time value. (Inherited from TimeBase [1605].)</td>
</tr>
<tr>
<td>Ticks [1610]</td>
<td>Returns the number of ticks that represent the value of this TimeBase [1605] (uint32 or uint64). (Inherited from TimeBase [1605].)</td>
</tr>
<tr>
<td>Time [1611]</td>
<td>Gets the time value. (Inherited from TimeBase [1605].)</td>
</tr>
</tbody>
</table>

Reference
TIME Class [1598]
TwinCAT.PlcOpen Namespace [1552]

6.10.8.3 TIME Methods

The TIME [1598] type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1612]</td>
<td>Determines whether the specified Object is equal to this instance. (Inherited from TimeBase [1605].)</td>
</tr>
<tr>
<td>GetHashCode [1612]</td>
<td>Gets the HashCode of the Address (Inherited from TimeBase [1605].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse [1604]</td>
<td>Parses the specified string to a TIME [1598] object.</td>
</tr>
<tr>
<td>ToString [1604]</td>
<td>Returns a String that represents this instance. (Overrides Object.ToString.)</td>
</tr>
<tr>
<td>TryParse [1605]</td>
<td>Tries to parse the TIME [1598] object from string.</td>
</tr>
</tbody>
</table>
6.10.8.3.1 TIME.Parse Method

Parses the specified string to a TIME object.

Namespace: TwinCAT.PlcOpen
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#

```csharp
public static TIME Parse(
    string str
)
```

Parameters

- `str` Type: `System.String`
The string.

Return Value

Type: TIME
TIME.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormatException</td>
<td>Cannot create TIME DataType!</td>
</tr>
</tbody>
</table>

Reference
TIME Class
TwinCAT.PlcOpen Namespace

6.10.8.3.2 TIME.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.PlcOpen
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#

```csharp
public override string ToString()
```
Return Value

Type: String
A String that represents this instance.

Reference

TIME Class [1598]
TwinCAT.PlcOpen Namespace [1552]

6.10.8.3.3 TIME.TryParse Method

Tries to parse the TIME [1598] object from string.

Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```c#
public static bool TryParse(
    string str,
    out TIME ret
)
```

Parameters

str Type: System.String
The string.

ret Type: TwinCAT.PlcOpen.TIME [1598].
The ret.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

TIME Class [1598]
TwinCAT.PlcOpen Namespace [1552]

6.10.9 TimeBase Class

Base class for PlcOpen Time types.

Inheritance Hierarchy

System.Object
TwinCAT.PlcOpen.Base
    TwinCAT.PlcOpen.TIME [1598]
    TwinCAT.PlcOpen.TOD [1615]
Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public abstract class TimeBase : IPlcOpenTimeBase<TimeSpan, uint>, IPlcOpenTimeBase
```

The `TimeBase` type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeBase</td>
<td>Initializes a new instance of the TimeBase class.</td>
</tr>
<tr>
<td>TimeBase(Int64)</td>
<td>Initializes a new instance of the TimeBase class.</td>
</tr>
<tr>
<td>TimeBase(UInt32)</td>
<td>Initializes a new instance of the TimeBase class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalTimeValue</td>
<td>Gets the internal time value.</td>
</tr>
<tr>
<td>MarshalSize</td>
<td>Gets the marshal size in bytes.</td>
</tr>
<tr>
<td>Ticks</td>
<td>Returns the number of ticks that represent the value of this TimeBase</td>
</tr>
<tr>
<td></td>
<td>(uint32 or uint64).</td>
</tr>
<tr>
<td>Time</td>
<td>Gets the time value.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to this instance.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Overrides Object.GetHashCode.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TimeToValue</td>
<td>Converts the TimeSpan to PlcOpen ticks.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueToTime(Int64)</td>
<td>Converts the timeValue (PlcOpen ticks) to TimeSpan</td>
</tr>
<tr>
<td>ValueToTime(UInt32)</td>
<td>Converts the timeValue (PlcOpen ticks) to TimeSpan</td>
</tr>
</tbody>
</table>

**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>internalTimeValue</td>
<td>The internal time value</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.PlcOpen Namespace [1552]

### 6.10.9.1 TimeBase Constructor

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeBase()</td>
<td>Initializes a new instance of the TimeBase class.</td>
</tr>
<tr>
<td>TimeBase(Int64)</td>
<td>Initializes a new instance of the TimeBase class.</td>
</tr>
<tr>
<td>TimeBase(UInt32)</td>
<td>Initializes a new instance of the TimeBase class.</td>
</tr>
</tbody>
</table>

**Reference**

TimeBase Class [1605]

TwinCAT.PlcOpen Namespace [1552]

### 6.10.9.1.1 TimeBase Constructor

Initializes a new instance of the TimeBase class.

**Namespace**: TwinCAT.PlcOpen [1552]

**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
protected TimeBase()
```

**Reference**

TimeBase Class [1605]

TimeBase Overload [1607]
6.10.9.1.2 TimeBase Constructor (Int64)

Initializes a new instance of the TimeBase class.

**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected TimeBase(
    long timeValue
)
```

**Parameters**

- **timeValue**
  - **Type:** System.Int64
  - The time value.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

- TimeBase Class
- TimeBase Overload
- TwinCAT.PlcOpen Namespace

6.10.9.1.3 TimeBase Constructor (UInt32)

Initializes a new instance of the TimeBase class.

**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected TimeBase(
    uint timeValue
)
```

**Parameters**

- **timeValue**
  - **Type:** System.UInt32
  - The time value.
Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

**TimeBase Class [1605]**
**TimeBase Overload [1607]**
**TwinCAT.PlcOpen Namespace [1552]**

### 6.10.9.2 TimeBase Properties

The **TimeBase [1605]** type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>InternalTimeValue</code></td>
<td>Gets the internal time value.</td>
</tr>
<tr>
<td><code>MarshalSize</code></td>
<td>Gets the marshal size in bytes.</td>
</tr>
<tr>
<td><code>Ticks</code></td>
<td>Returns the number of ticks that represent the value of this TimeBase (uint32 or uint64).</td>
</tr>
<tr>
<td><code>Time</code></td>
<td>Gets the time value.</td>
</tr>
</tbody>
</table>

Reference

**TimeBase Class [1605]**
**TwinCAT.PlcOpen Namespace [1552]**

### 6.10.9.2.1 TimeBase.InternalTimeValue Property

Gets the internal time value.

**Namespace:** TwinCAT.PlcOpen [1552]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public uint InternalTimeValue { get; }
```

**Property Value**

Type: UInt32
The internal time value.
6.10.9.2.2  **TimeBase.MarshalSize Property**

Gets the marshal size in bytes.

**Namespace:**  TwinCAT.PlcOpen

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static int MarshalSize { get; }
```

**Property Value**

Type:  Int32
Marshalling size in bytes.

**Reference**

TimeBase Class [》1605]

TwinCAT.PlcOpen Namespace [》1552]

6.10.9.2.3  **TimeBase.Ticks Property**

Returns the number of ticks that represent the value of this **TimeBase** (uint32 or uint64).

**Namespace:**  TwinCAT.PlcOpen

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public uint Ticks { get; }
```

**Property Value**

Type:  UInt32
The ticks (in 100ns).

**Implements**

IPlcOpenTimeBase.T1, T2..Ticks [》1580]

**Reference**

TimeBase Class [》1605]

TwinCAT.PlcOpen Namespace [》1552]
6.10.9.2.4 TimeBase.Time Property

Gets the time value.

Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public virtual TimeSpan Time { get; }
```

Property Value

Type: TimeSpan
The time.

Reference

TimeBase Class [1605]
TwinCAT.PlcOpen Namespace [1552]

6.10.9.3 TimeBase Methods

The TimeBase [1605] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1612]</td>
<td>Determines whether the specified Object is equal to this instance.</td>
</tr>
<tr>
<td></td>
<td>(Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TimeToValue [1613]</td>
<td>Converts the TimeSpan to PlcOpen ticks.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.</td>
</tr>
<tr>
<td>ValueToTime(Int64)</td>
<td>Converts the timeValue (PlcOpen ticks) to TimeSpan</td>
</tr>
<tr>
<td></td>
<td>[1614]</td>
</tr>
<tr>
<td>ValueToTime(UInt32)</td>
<td>Converts the timeValue (PlcOpen ticks) to TimeSpan</td>
</tr>
<tr>
<td></td>
<td>[1614]</td>
</tr>
</tbody>
</table>
6.10.9.3.1 **TimeBase.Equals Method**

Determines whether the specified `Object` is equal to this instance.

**Namespace:**  TwinCAT.PlcOpen

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public override bool Equals(
    Object obj
)
```

**Parameters**

- **obj**  
  - Type:  `System.Object`
  - The object to compare with the current object.

**Return Value**

- Type:  `Boolean`
  - true if the specified `Object` is equal to this instance; otherwise, false.

---

6.10.9.3.2 **TimeBase.GetHashCode Method**

Gets the HashCode of the Address

**Namespace:**  TwinCAT.PlcOpen

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public override int GetHashCode()
```

**Return Value**

- Type:  `Int32`
6.10.9.3.3  TimeBase.TimeToValue Method

Converts the TimeSpan to PlcOpen ticks.

Namespace:  TwinCAT.PlcOpen
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static long TimeToValue(
    TimeSpan time
)

Parameters

- time  Type: System.TimeSpan
  The time.

Return Value

Type: Int64

Reference

TimeBase Class [1605]
TwinCAT.PlcOpen Namespace [1552]

6.10.9.3.4  TimeBase.ValueToTime Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueToTime(Int64)</td>
<td>Converts the timeValue (PlcOpen ticks) to TimeSpan</td>
</tr>
<tr>
<td>ValueToTime(UInt32)</td>
<td>Converts the timeValue (PlcOpen ticks) to TimeSpan</td>
</tr>
</tbody>
</table>

Reference

TimeBase Class [1605]
TwinCAT.PlcOpen Namespace [1552]
**TimeBase.ValueToTime Method (Int64)**

Converts the timeValue (PlcOpen ticks) to TimeSpan

**Namespace:** [TwinCAT.PlcOpen](#)  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public static TimeSpan ValueToTime(
    long timeValue
)
```

**Parameters**

- `timeValue`  
  Type: `System.Int64`  
  The time value.

**Return Value**

- Type: `TimeSpan`

**Reference**

- [TimeBase Class](#)  
- [ValueToTime Overload](#)  
- [TwinCAT.PlcOpen Namespace](#)

**TimeBase.ValueToTime Method (UInt32)**

Converts the timeValue (PlcOpen ticks) to TimeSpan

**Namespace:** [TwinCAT.PlcOpen](#)  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public static TimeSpan ValueToTime(
    uint timeValue
)
```

**Parameters**

- `timeValue`  
  Type: `System.UInt32`  
  The time value.

**Return Value**

- Type: `TimeSpan`
6.10.9.4 TimeBase Fields

The `TimeBase` type exposes the following members.

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>internalTimeValue</code></td>
<td>The internal time value</td>
</tr>
</tbody>
</table>

6.10.9.4.1 TimeBase.internalTimeValue Field

The internal time value

**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
protected uint internalTimeValue
```

**Field Value**

Type: `UInt32`

6.10.10 TOD Class

PLCOpen TimeOfDay class (32-Bit)

**Inheritance Hierarchy**

- System.Object
  - TwinCAT.PlcOpen.TimeBase
    - TwinCAT.PlcOpen.TOD
TwinCAT.Ads Namespaces

Namespace:  TwinCAT.PlcOpen [1552]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public sealed class TOD : TimeBase
```

The TOD type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOD</td>
<td>Initializes a new instance of the TOD class.</td>
</tr>
<tr>
<td>TOD(Int64)</td>
<td>Initializes a new instance of the TOD class.</td>
</tr>
<tr>
<td>TOD(TimeSpan)</td>
<td>Initializes a new instance of the TOD class.</td>
</tr>
<tr>
<td>TOD(UInt32)</td>
<td>Initializes a new instance of the TOD class.</td>
</tr>
<tr>
<td>TOD(Int32, Int32, Int32, Int32)</td>
<td>Initializes a new instance of the TOD class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalTimeValue</td>
<td>Gets the internal time value. (Inherited from TimeBase.</td>
</tr>
<tr>
<td>Ticks</td>
<td>Returns the number of ticks that represent the value of this TimeBase (uint32 or uint64). (Inherited from TimeBase.)</td>
</tr>
<tr>
<td>Time</td>
<td>Gets the time value. (Inherited from TimeBase.)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to this instance. (Inherited from TimeBase.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Inherited from TimeBase.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse</td>
<td>Parses the specified string to a TOD object.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Overrides Object.ToString.)</td>
</tr>
<tr>
<td>TryParse</td>
<td>Tries to parse the string to a TOD object.</td>
</tr>
</tbody>
</table>
6.10.10.1  TOD Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOD()</td>
<td>Initializes a new instance of the TOD class.</td>
</tr>
<tr>
<td>TOD(Int64)</td>
<td>Initializes a new instance of the TOD class.</td>
</tr>
<tr>
<td>TOD(TimeSpan)</td>
<td>Initializes a new instance of the TOD class.</td>
</tr>
<tr>
<td>TOD(UInt32)</td>
<td>Initializes a new instance of the TOD class.</td>
</tr>
<tr>
<td>TOD(Int32, Int32, Int32)</td>
<td>Initializes a new instance of the TOD class.</td>
</tr>
</tbody>
</table>

6.10.10.1.1  TOD Constructor (Int64)

Initializes a new instance of the TOD class.

**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd2ca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public TOD(Int64)
```

6.10.10.1.2  TOD Constructor (Int32)

Initializes a new instance of the TOD class.

**Namespace:** TwinCAT.PlcOpen

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd2ca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public TOD(
    long time
)
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>System.Int64</td>
<td>The time.</td>
</tr>
</tbody>
</table>

**Reference**

- TOD Class [1615]
- TOD Overload [1617]
- TwinCAT.PlcOpen Namespace [1552]

### 6.10.10.1.3 TOD Constructor (TimeSpan)

Initializes a new instance of the TOD class.

**Namespace:** TwinCAT.PlcOpen [1552]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public TOD(
    TimeSpan timeSpan
)
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeSpan</td>
<td>System.TimeSpan</td>
<td>The time span.</td>
</tr>
</tbody>
</table>

**Reference**

- TOD Class [1615]
- TOD Overload [1617]
- TwinCAT.PlcOpen Namespace [1552]

### 6.10.10.1.4 TOD Constructor (UInt32)

Initializes a new instance of the TOD class.

**Namespace:** TwinCAT.PlcOpen [1552]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```c#
public TOD(
    uint time
)
```

Parameters

time Type: System.UInt32
The time.

Reference

TOD Class [1615]
TOD Overload [1617]
TwinCAT.PlcOpen Namespace [1552]

6.10.10.1.5 TOD Constructor (Int32, Int32, Int32, Int32, Int32)

Initializes a new instance of the TOD [1615] class.

Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90b9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```c#
public TOD(
    int days,
    int hours,
    int minutes,
    int seconds,
    int milliseconds
)
```

Parameters

days Type: System.Int32
The days.

hours Type: System.Int32
The hours.

minutes Type: System.Int32
The minutes.

seconds Type: System.Int32
The seconds.

milliseconds Type: System.Int32
The milliseconds.

Reference

TOD Class [1615]
TOD Overload [1617]
TwinCAT.PlcOpen Namespace [1552]
6.10.10.2  TOD Properties

The TOD [1615] type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalTimeValue</td>
<td>Gets the internal time value. (Inherited from TimeBase [1605].)</td>
</tr>
<tr>
<td>Ticks [1610]</td>
<td>Returns the number of ticks that represent the value of this TimeBase [1605] (uint32 or uint64). (Inherited from TimeBase [1605].)</td>
</tr>
<tr>
<td>Time [1611]</td>
<td>Gets the time value. (Inherited from TimeBase [1605].)</td>
</tr>
</tbody>
</table>

### Reference

TOD Class [1615]

TwinCAT.PlcOpen Namespace [1552]

6.10.10.3  TOD Methods

The TOD [1615] type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1612]</td>
<td>Determines whether the specified Object is equal to this instance. (Inherited from TimeBase [1605].)</td>
</tr>
<tr>
<td>GetHashCode [1612]</td>
<td>Gets the HashCode of the Address (Inherited from TimeBase [1605].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse [1620]</td>
<td>Parses the specified string to a TOD [1615] object.</td>
</tr>
<tr>
<td>ToString [1621]</td>
<td>Returns a string that represents the current object. (Overrides Object.ToString.)</td>
</tr>
<tr>
<td>TryParse [1621]</td>
<td>Tries to parse the string to a TOD [1615] object.</td>
</tr>
</tbody>
</table>

### Reference

TOD Class [1615]

TwinCAT.PlcOpen Namespace [1552]

6.10.10.3.1  TOD.Parse Method

Parses the specified string to a TOD [1615] object.

**Namespace:** TwinCAT.PlcOpen [1552]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

```csharp
public static TOD Parse(
    string str
)
```

Parameters

str Type: System.String
The string.

Return Value

Type: TOD[1615]
TOD.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormatException</td>
<td>Cannot parse TOD object!</td>
</tr>
</tbody>
</table>

Reference

TOD Class [1615]
TwinCAT.PlcOpen Namespace [1552]

6.10.10.3.2 TOD.ToString Method

Returns a string that represents the current object.

Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public override string ToString()
```

Return Value

Type: String
A string that represents the current object.

Reference

TOD Class [1615]
TwinCAT.PlcOpen Namespace [1552]

6.10.10.3.3 TOD.TryParse Method

Tries to parse the string to a TOD[1615] object.
TwinCAT.Ads Namespaces

Namespace: TwinCAT.PlcOpen [1552]
Assembly: TwinCAT.Ads.Abstractsions (in TwinCAT.Ads.Abstractsions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static bool TryParse(
    string str,
    out TOD ret
)
```

Parameters

- `str`: Type: System.String
  The string.
- `ret`: Type: TwinCAT.PlcOpen.TOD [1615]
  The ret.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

TOD Class [1615]
TwinCAT.PlcOpen Namespace [1552]

6.11 TwinCAT.TypeSystem Namespace

Namespace for the common (non ADS dependent) type system.

Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnySymbolSpecifier [1629]</td>
<td>Class AnySymbolSpecifier.</td>
</tr>
<tr>
<td>AnyTypeSpecifier [1633]</td>
<td>Class AnyTypeSpecifier.</td>
</tr>
<tr>
<td>CannotAccessVirtualSymbolException [1641]</td>
<td>Cannot access virtual Symbol</td>
</tr>
<tr>
<td>CannotResolveDataTypeException [1644]</td>
<td>Class CannotResolveDataTypeException. Implements the DataTypeException [1658]</td>
</tr>
<tr>
<td>DataTypeCollection [1650]</td>
<td>Collection of DataTypes [1986]</td>
</tr>
<tr>
<td>DataTypeEventArgs [1656]</td>
<td>Class DataTypeEventArgs.</td>
</tr>
<tr>
<td>DataTypeException [1658]</td>
<td>Data Type Exception</td>
</tr>
<tr>
<td>Class</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>DataTypeNameEventArgs</code></td>
<td>Class <code>DataTypeNameEventArgs</code>.</td>
</tr>
<tr>
<td><code>Dimension</code></td>
<td>Represents a single dimension of an <code>IArrayType</code>.</td>
</tr>
<tr>
<td><code>DimensionCollection</code></td>
<td>Collection class for Array Dimensions.</td>
</tr>
<tr>
<td><code>DynamicAliasInstance</code></td>
<td>Class <code>DynamicAliasInstance</code>. This class cannot be inherited.</td>
</tr>
<tr>
<td><code>DynamicArrayInstance</code></td>
<td>Dynamic Array Instance</td>
</tr>
<tr>
<td><code>DynamicOversamplingArrayInstance</code></td>
<td>Dynamic Array Instance</td>
</tr>
<tr>
<td><code>DynamicPointerInstance</code></td>
<td>Dynamic Pointer Instance</td>
</tr>
<tr>
<td><code>DynamicPointerValue</code></td>
<td>Class <code>DynamicPointerValue</code>.</td>
</tr>
<tr>
<td><code>DynamicReferenceInstance</code></td>
<td>Dynamic Reference Instance</td>
</tr>
<tr>
<td><code>DynamicReferenceValue</code></td>
<td>Class <code>DynamicReferenceValue</code>.</td>
</tr>
<tr>
<td><code>DynamicRpcStructInstance</code></td>
<td>Dynamic struct instance with RPC Methods.</td>
</tr>
<tr>
<td><code>DynamicStructInstance</code></td>
<td>Dynamic struct instance</td>
</tr>
<tr>
<td><code>DynamicSymbol</code></td>
<td>Dynamic <code>Symbol</code> object.</td>
</tr>
<tr>
<td><code>DynamicSymbolsCollection</code></td>
<td>Dynamic (Expandable) Symbols collection.</td>
</tr>
<tr>
<td><code>DynamicUnionInstance</code></td>
<td>Dynamic union instance</td>
</tr>
<tr>
<td><code>DynamicValue</code></td>
<td>Dynamic value (uses <code>RuntimeBinding</code> for <code>ISymbol</code> value reading / writing).</td>
</tr>
<tr>
<td>Class</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>DynamicVirtualStructure</td>
<td>Dynamic struct instance</td>
</tr>
<tr>
<td>EnumValue.T</td>
<td>Enum Value</td>
</tr>
<tr>
<td>EnumValueCollection</td>
<td>Class EnumValueCollection</td>
</tr>
<tr>
<td>EnumValueCollection.T</td>
<td>Collection of EnumValues [1901]</td>
</tr>
<tr>
<td>FieldCollection</td>
<td>Collection of IField [2040] objects.</td>
</tr>
<tr>
<td>InsufficientAccessRightsException</td>
<td>Insufficient rights for access</td>
</tr>
<tr>
<td>MarshalException</td>
<td>Common Marshalling Exception</td>
</tr>
<tr>
<td>MemberCollection</td>
<td>Collection of IMember [2065] objects.</td>
</tr>
<tr>
<td>ReadOnlyDataTypeCollection</td>
<td>ReadOnly Collection of IDataType [1986] objects.</td>
</tr>
<tr>
<td>ReadOnlyDimensionCollection</td>
<td>ReadOnly version of the DimensionCollection [1671]</td>
</tr>
<tr>
<td>ReadOnlyEnumValueCollection</td>
<td>Read only version of the EnumValueCollection.T [1926]</td>
</tr>
<tr>
<td>ReadOnlyEnumValueCollection.T</td>
<td>Read only version of the EnumValueCollection.T [1926]</td>
</tr>
<tr>
<td>ReadOnlyFieldCollection</td>
<td>Read only collection of IField [2040] objects.</td>
</tr>
<tr>
<td>ReadOnlyMemberCollection</td>
<td>Read only collection of IMember [2065] objects.</td>
</tr>
<tr>
<td>ReadOnlyRpcMethodParameterCollection</td>
<td>Read only RpcMethodParameterCollection [2384].</td>
</tr>
<tr>
<td>ReadOnlyRpcMethodCollection</td>
<td>Read only RpcMethodCollection [2364].</td>
</tr>
<tr>
<td>Class</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>ReadOnlySymbolCollection [2336]</strong></td>
<td>ReadOnly collection containing ISymbol [2176] objects.</td>
</tr>
<tr>
<td><strong>ReadOnlyTypeAttributeCollection [2340]</strong></td>
<td>Read only version of the TypeAttributeCollection [2418]</td>
</tr>
<tr>
<td><strong>ResultDataTypes [2347]</strong></td>
<td>Class representing the asynchronous result of reading a IDataTypeCollection [1993] via ADS. Implements the ResultValue.TValue. [1029]</td>
</tr>
<tr>
<td><strong>ResultDynamicSymbols [2350]</strong></td>
<td>Class representing the asynchronous result of reading a dynamic symbol collection via ADS. Implements the ResultValue.TValue. [1029]</td>
</tr>
<tr>
<td><strong>ResultSymbols [2353]</strong></td>
<td>Class representing the asynchronous result of reading an symbol collection of type ISymbolCollection.T [2185] via ADS. Implements the ResultValue.TValue. [1029]</td>
</tr>
<tr>
<td><strong>ResultSymbols.T [2356]</strong></td>
<td>Class representing the asynchronous result of reading a symbol enumeration of type IEnumerable.T via ADS. Implements the ResultValue.TValue. [1029]</td>
</tr>
<tr>
<td><strong>RpcInvokeException [2359]</strong></td>
<td>Class RpcInvokeException. Implements the SymbolException [2401]</td>
</tr>
<tr>
<td><strong>RpcMethodCollection [2364]</strong></td>
<td>Collection of RpcMethods. [2123]</td>
</tr>
<tr>
<td><strong>RpcMethodNotSupportedException [2379]</strong></td>
<td>Symbol Exception</td>
</tr>
<tr>
<td><strong>RpcMethodParameterCollection [2384]</strong></td>
<td>Collection of RPC method parameters</td>
</tr>
<tr>
<td><strong>SymbolCollection [2396]</strong></td>
<td>Interface represents a collection of ISymbol [2176] objects.</td>
</tr>
<tr>
<td><strong>SymbolException [2401]</strong></td>
<td>Symbol bound exceptions</td>
</tr>
<tr>
<td><strong>TypeAttribute [2413]</strong></td>
<td>ADS Attribute</td>
</tr>
<tr>
<td><strong>TypeAttributeCollection [2418]</strong></td>
<td>Collection of AdsAttributes [2209]</td>
</tr>
<tr>
<td><strong>ValueChangedBaseEventArgs [2435]</strong></td>
<td>Event args for the RawValueChanged [2254] event.</td>
</tr>
<tr>
<td><strong>ValueChangedEventArgs [2439]</strong></td>
<td>Event args for the ValueChanged [2269] event.</td>
</tr>
<tr>
<td>Interface</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IAliasInstance</td>
<td>Interface representing an instance of an IAliasType [1954].</td>
</tr>
<tr>
<td>IAliasType [1954]</td>
<td>Interface representing an Alias Type</td>
</tr>
<tr>
<td>IAnyTypeMarshaler</td>
<td>Interface IAnyTypeMarshaler Implements the IGenericTypeMarshaler [2046]</td>
</tr>
<tr>
<td>IArrayInstance</td>
<td>Interface representing an array instance</td>
</tr>
<tr>
<td>IArrayType [1964]</td>
<td>Interface representing an array DataType [1986].</td>
</tr>
<tr>
<td>IArrayValue [1971]</td>
<td>Interface IArrayValue</td>
</tr>
<tr>
<td>IAttributedInstance</td>
<td>Interface IAttributedInstance</td>
</tr>
<tr>
<td>IBitSize [1980]</td>
<td>Interface IBitSize</td>
</tr>
<tr>
<td>IDataType [1986]</td>
<td>Base interface for objects representing data types</td>
</tr>
<tr>
<td>IDataTypeCollection</td>
<td>Interface IDataTypeCollection Implements the ICollection.T.</td>
</tr>
<tr>
<td>IDataTypeCollection.T.</td>
<td>Data Type container interface</td>
</tr>
<tr>
<td>IDimensionCollection</td>
<td>Interface IDimensionCollection</td>
</tr>
<tr>
<td>IDynamicSymbol</td>
<td>Interface IDynamicSymbol</td>
</tr>
<tr>
<td>IDynamicSymbolLoader</td>
<td>Dynamic symbol loader interface</td>
</tr>
<tr>
<td>IDynamicSymbolsCollection</td>
<td>Interface IDynamicSymbolsContainer Implements the IDynamicMetaObjectProvider</td>
</tr>
<tr>
<td>IDynamicValue</td>
<td>Interface IDynamicValue Implements the IDynamicMetaObjectProvider Implements the IValue [2226] Implements the IStructValue [2167] Implements the IArrayValue [1976]</td>
</tr>
<tr>
<td>IEnumType [2014]</td>
<td>Common Enum type interface</td>
</tr>
<tr>
<td>IEnumType.T. [2021]</td>
<td>Interface representing an enum type</td>
</tr>
<tr>
<td>IEnumValue [2028]</td>
<td>Generic interface for EnumValues</td>
</tr>
<tr>
<td>IEnumValueCollection</td>
<td>Interface IEnumValueCollection</td>
</tr>
<tr>
<td>Interface</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>IEnumValueCollection</td>
<td>Interface for collections of IEnumValues. Implements the ICollection.T.</td>
</tr>
<tr>
<td>IField</td>
<td>Specifies a single field/member of a Struct DataType.</td>
</tr>
<tr>
<td>IFieldCollection</td>
<td>Interface IFieldCollection Implements the IInstanceCollection.T.</td>
</tr>
<tr>
<td>IGenericTypeMarshaller</td>
<td>Interface IGenericTypeMarshaller Implements the ITypeMarshaler.</td>
</tr>
<tr>
<td>IHierarchicalSymbol</td>
<td>Bindable Symbol interface (for internal use only)</td>
</tr>
<tr>
<td>Instance</td>
<td>Interface specifying instance objects.</td>
</tr>
<tr>
<td>IInstanceCollection.T.</td>
<td>Generic InstanceCollection interface.</td>
</tr>
<tr>
<td>IMember</td>
<td>Specifies a single field/member of a Struct DataType.</td>
</tr>
<tr>
<td>IMemberCollection</td>
<td>Interface IMemberCollection Implements the IInstanceCollection.T.</td>
</tr>
<tr>
<td>INamespaceCollection</td>
<td>Interface INamespaceCollection</td>
</tr>
<tr>
<td>INamespaceCollection.T.</td>
<td>Interface INamespaceCollection</td>
</tr>
<tr>
<td>IOversamplingArrayInstance</td>
<td>Interface IOversamplingArrayInstance</td>
</tr>
<tr>
<td>IPointerType</td>
<td>Interface representing a pointer type</td>
</tr>
<tr>
<td>IPrimitiveType</td>
<td>Interface IPrimitiveType</td>
</tr>
<tr>
<td>IProcessImageAddress</td>
<td>Interface describing a Process Image Address</td>
</tr>
<tr>
<td>IReferenceInstance</td>
<td>Interface representing an instance of an IPointerType.</td>
</tr>
<tr>
<td>IPointerType</td>
<td>Interface representing a reference/pointer type</td>
</tr>
<tr>
<td>IRpcCallableInstance</td>
<td>Interface for an RPC callable PLC Method (Remote procedure call)</td>
</tr>
<tr>
<td>IRpcCallableType</td>
<td>Interface representing an RPC callable IStructType.</td>
</tr>
<tr>
<td>IRpcMethod</td>
<td>Interface describes an RPC Method</td>
</tr>
<tr>
<td>Interface</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>IRpcMethodCollection [2127]</td>
<td>Interface for RPC Method collections.</td>
</tr>
<tr>
<td>IRpcMethodParameter [2133]</td>
<td>Interface IRpcMethodParameterCollection</td>
</tr>
<tr>
<td>IRpcMethodParameterCollection [2137]</td>
<td></td>
</tr>
<tr>
<td>IRpcStructInstance [2140]</td>
<td>Interface IRpcStructInstance</td>
</tr>
<tr>
<td>IStringInstance [2145]</td>
<td>Interface IStringInstance</td>
</tr>
<tr>
<td>IStringMarshaler [2148]</td>
<td>Common interface for marshalling ADS string values.</td>
</tr>
<tr>
<td>IStringType [2155]</td>
<td>Interface representing a string IDataType [1986]</td>
</tr>
<tr>
<td>IStructInstance [2158]</td>
<td>Interface representing an instance of a IStructType [2162]</td>
</tr>
<tr>
<td>IStructType [2162]</td>
<td>Interface representing Struct data types</td>
</tr>
<tr>
<td>IStructValue [2167]</td>
<td>Interface IStructValue</td>
</tr>
<tr>
<td>ISubRangeType [2170]</td>
<td>Interface representing a SubRange type</td>
</tr>
<tr>
<td>ISubRangeType.T. [2173]</td>
<td>Interface representing a SubRange type</td>
</tr>
<tr>
<td>ISymbol [2176]</td>
<td>Interface specifying Symbols (</td>
</tr>
<tr>
<td>ISymbolCollection [2182]</td>
<td>Interface ISymbolCollection Implements the IInstanceCollection.T. [2057]</td>
</tr>
<tr>
<td>ISymbolCollection.T. [2185]</td>
<td>Interface ISymbolCollection</td>
</tr>
<tr>
<td>ISymbolFactory [2188]</td>
<td>Symbol Factory Interface</td>
</tr>
<tr>
<td>ISymbolFactoryServicesProvider [2198]</td>
<td>Symbol Value Access interface</td>
</tr>
<tr>
<td>ISymbolInfo [2199]</td>
<td>Interface ISymbolInfo</td>
</tr>
<tr>
<td>ISymbolLoader [2200]</td>
<td>Symbol Loader interface</td>
</tr>
<tr>
<td>ISymbolProvider [2203]</td>
<td>Symbol Provider interface.</td>
</tr>
<tr>
<td>ISymbolServer [2205]</td>
<td>Symbol Server Interface</td>
</tr>
<tr>
<td>ITypeAttribute [2209]</td>
<td>Interface for ADS attributes</td>
</tr>
<tr>
<td>Interface</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ITypeAttributeCollection</td>
<td>Interface ITypeAttributeCollection</td>
</tr>
<tr>
<td>ITypeMarshaler</td>
<td>Interface ITypeMarshaler</td>
</tr>
<tr>
<td>IUnionInstance</td>
<td>Interface for an Instance of the IUnionType</td>
</tr>
<tr>
<td>IUnionType</td>
<td>Interface for an union data type.</td>
</tr>
<tr>
<td>IValue</td>
<td>Symbol Value Interface</td>
</tr>
<tr>
<td>IValueAccessorProvider</td>
<td>Interface IValueAccessorProvider</td>
</tr>
<tr>
<td>IValueAnySymbol</td>
<td>Interface IValueAnySymbol</td>
</tr>
<tr>
<td>IValueRawSymbol</td>
<td>Interface IValueRawSymbol</td>
</tr>
<tr>
<td>IValueSymbol</td>
<td>Interface for a Symbol that supports values.</td>
</tr>
<tr>
<td>IVirtualStructInstance</td>
<td>Virtual Struct instance interface.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypeCategory</td>
<td>Category of a DataType / Instance</td>
</tr>
<tr>
<td>InstanceCollectionMode</td>
<td>Enum InstanceCollectionMode</td>
</tr>
<tr>
<td>MethodParamFlags</td>
<td>Flag set specifying the MethodParameter context</td>
</tr>
<tr>
<td>PrimitiveTypeFlags</td>
<td>Enum PrimitiveTypeFlags</td>
</tr>
<tr>
<td>StringConvertMode</td>
<td>Enum StringConvertMode</td>
</tr>
<tr>
<td>SymbolAccessRights</td>
<td>Enum specifying Access Rights to symbols</td>
</tr>
</tbody>
</table>

### 6.11.1 AnySymbolSpecifier Class

Class AnySymbolSpecifier.
Inheritance Hierarchy

System.Object
   TwinCAT.TypeSystem.AnySymbolSpecifier

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class AnySymbolSpecifier

The AnySymbolSpecifier type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnySymbolSpecifier</td>
<td>Initializes a new instance of the AnySymbolSpecifier class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstancePath</td>
<td>Gets the instance path.</td>
</tr>
<tr>
<td>TypeSpecifier</td>
<td>Gets the type specifier.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Remarks

Specifies an Symbol path together with an AnyTypeSpecifier [1633] to address a symbol for an Read/Write operation

Reference

TwinCAT.TypeSystem Namespace [1622]
6.11.1.1 AnySymbolSpecifier Constructor

Initializes a new instance of the AnySymbolSpecifier class.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public AnySymbolSpecifier(
    string instancePath,
    AnyTypeSpecifier spec
)
```

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>instancePath</td>
<td>System.String</td>
<td>The instance path.</td>
</tr>
<tr>
<td>spec</td>
<td>TwinCAT.TypeSystem.AnyTypeSpecifier</td>
<td>The spec.</td>
</tr>
</tbody>
</table>

**Reference**

AnySymbolSpecifier Class

TwinCAT.TypeSystem Namespace

6.11.1.2 AnySymbolSpecifier Properties

The AnySymbolSpecifier type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstancePath</td>
<td>Gets the instance path.</td>
</tr>
<tr>
<td>TypeSpecifier</td>
<td>Gets the type specifier.</td>
</tr>
</tbody>
</table>

**Reference**

AnySymbolSpecifier Class

TwinCAT.TypeSystem Namespace

6.11.1.2.1 AnySymbolSpecifier.InstancePath Property

Gets the instance path.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
## Syntax

**C#**

```csharp
public string InstancePath { get; }
```

### Property Value

**Type:** String  
The instance path.

### Reference

- **AnySymbolSpecifier Class** [› 1629]
- **TwinCAT.TypeSystem Namespace** [› 1622]

### 6.11.1.2.2 AnySymbolSpecifier.TypeSpecifier Property

**Gets the type specifier.**

**Namespace:** TwinCAT.TypeSystem [› 1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)  
**Version:** 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public AnyTypeSpecifier TypeSpecifier { get; }
```

### Property Value

**Type:** AnyTypeSpecifier [› 1633]  
The type specifier.

### Reference

- **AnySymbolSpecifier Class** [› 1629]  
- **TwinCAT.TypeSystem Namespace** [› 1622]

### 6.11.1.3 AnySymbolSpecifier Methods

The **AnySymbolSpecifier** [› 1629] type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>getHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>getType</td>
<td>Gets the Type of the current instance.</td>
</tr>
</tbody>
</table>
### AnyTypeSpecifier Class

Class AnyTypeSpecifier.

#### Inheritance Hierarchy

System.Object  
TwinCAT.TypeSystem.AnyTypeSpecifier  

**Namespace:**  TwinCAT.TypeSystem  
**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)  
**Version:**  5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b3095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public class AnyTypeSpecifier
```

The AnyTypeSpecifier type exposes the following members.

#### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="1635" alt="AnyTypeSpecifier(Object)" /></td>
<td>Initializes a new instance of the AnyTypeSpecifier class from the specified prototype</td>
</tr>
<tr>
<td><img src="1635" alt="AnyTypeSpecifier(Type)" /></td>
<td>Initializes a new instance of the AnyTypeSpecifier class.</td>
</tr>
<tr>
<td><img src="1636" alt="AnyTypeSpecifier(Type, IList&lt;IDimensionCollection&gt;)" /></td>
<td>Initializes a new instance of the AnyTypeSpecifier class.</td>
</tr>
<tr>
<td><img src="1636" alt="AnyTypeSpecifier(Type, Int32)" /></td>
<td>Initializes a new instance of the AnyTypeSpecifier class.</td>
</tr>
<tr>
<td><img src="1637" alt="AnyTypeSpecifier(Type, Int32, Int32)" /></td>
<td>Initializes a new instance of the AnyTypeSpecifier class.</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category [1638]</td>
<td>Category of the AnyTypeSpecifier</td>
</tr>
<tr>
<td>DimLengths [1639]</td>
<td>List of jagged Dimensions (Arrays and jagged arrays)</td>
</tr>
<tr>
<td>ElementType [1639]</td>
<td>The element type (Arrays)</td>
</tr>
<tr>
<td>StrLen [1639]</td>
<td>The String length (only for String [1649])</td>
</tr>
<tr>
<td>Type [1640]</td>
<td>Managed type</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Remarks

The AnyTypeSpecifier is used to specify out/return parameters for ReadAny, InvokeRpc-Methods or typed AdsNotifications.

### Reference

TwinCAT.TypeSystem Namespace [1622]

### 6.11.2.1 AnyTypeSpecifier Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnyTypeSpecifier(Object) [1635]</td>
<td>Initializes a new instance of the AnyTypeSpecifier [1633] class from the specified prototype</td>
</tr>
<tr>
<td>AnyTypeSpecifier(Type) [1635]</td>
<td>Initializes a new instance of the AnyTypeSpecifier [1633] class.</td>
</tr>
<tr>
<td>AnyTypeSpecifier(Type, IList&lt;IDimensionCollection&gt;) [1636]</td>
<td>Initializes a new instance of the AnyTypeSpecifier [1633] class.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>AnyTypeSpecifier(Ty pe, Int32)</code></td>
<td>Initializes a new instance of the <code>AnyTypeSpecifier</code> class.</td>
</tr>
<tr>
<td><code>AnyTypeSpecifier(Ty pe, .Int32.)</code></td>
<td>Initializes a new instance of the <code>AnyTypeSpecifier</code> class.</td>
</tr>
</tbody>
</table>

Reference

`AnyTypeSpecifier Class` [1633]

`TwinCAT.TypeSystem Namespace` [1622]

### 6.11.2.1.1 `AnyTypeSpecifier` Constructor (Object)

Initializes a new instance of the `AnyTypeSpecifier` class from the specified prototype.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public AnyTypeSpecifier(
    Object prototype
)
```

**Parameters**

- `prototype`  
  *Type:* `System.Object`  
  *The prototype.*

Reference

`AnyTypeSpecifier Class` [1633]

`AnyTypeSpecifier Overload` [1634]

`TwinCAT.TypeSystem Namespace` [1622]

### 6.11.2.1.2 `AnyTypeSpecifier Constructor` (Type)

Initializes a new instance of the `AnyTypeSpecifier` class.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public AnyTypeSpecifier(
    Type type
)
```
Parameters

type
  Type: System.Type
  The type.

Reference

AnyTypeSpecifier Class [1633]
AnyTypeSpecifier Overload [1634]
TwinCAT.TypeSystem Namespace [1622]

6.11.2.1.3 AnyTypeSpecifier Constructor (Type, IList.IDimensionCollection.)

Initializes a new instance of the AnyTypeSpecifier class.

Namespace: TwinCAT.TypeSystem [1622]
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AnyTypeSpecifier(
    Type type,
    IList<IDimensionCollection> dimLengths
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>System.Type</td>
<td>The type.</td>
</tr>
</tbody>
</table>

Reference

AnyTypeSpecifier Class [1633]
AnyTypeSpecifier Overload [1634]
TwinCAT.TypeSystem Namespace [1622]

6.11.2.1.4 AnyTypeSpecifier Constructor (Type, Int32)

Initializes a new instance of the AnyTypeSpecifier class.

Namespace: TwinCAT.TypeSystem [1622]
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public AnyTypeSpecifier(
    Type type,
    int strLen
)
```

Parameters

type
Type: System.Type
The type.

strLen
Type: System.Int32
Length of the string.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td>type</td>
</tr>
</tbody>
</table>

Reference

AnyTypeSpecifier Class [🔗1633]
AnyTypeSpecifier Overload [🔗1634]
TwinCAT.TypeSystem Namespace [🔗1622]

6.11.2.1.5 AnyTypeSpecifier Constructor (Type, .Int32.)

Initializes a new instance of the AnyTypeSpecifier [🔗1633] class.

Namespace: TwinCAT.TypeSystem [🔗1622]
Assembly: TwinCAT.Ads.Abstractions.dll Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public AnyTypeSpecifier(
    Type type,
    int[] dimLengths
)
```

Parameters

type
Type: System.Type
The type.

dimLengths
Type: System.Int32
The dim lengths.
Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td>type</td>
</tr>
</tbody>
</table>

Reference

AnyTypeSpecifier Class [1633]

AnyTypeSpecifier Overload [1634]

TwinCAT.TypeSystem Namespace [1622]

6.11.2.2 AnyTypeSpecifier Properties

The AnyTypeSpecifier [1633] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category [1638]</td>
<td>Category of the AnyTypeSpecifier [1633]</td>
</tr>
<tr>
<td>DimLengths [1639]</td>
<td>List of jagged Dimensions (Arrays and jagged arrays)</td>
</tr>
<tr>
<td>ElementType [1639]</td>
<td>The element type (Arrays)</td>
</tr>
<tr>
<td>StrLen [1639]</td>
<td>The String length (only for String [1649])</td>
</tr>
<tr>
<td>Type [1640]</td>
<td>Managed type</td>
</tr>
</tbody>
</table>

Reference

AnyTypeSpecifier Class [1633]

TwinCAT.TypeSystem Namespace [1622]

6.11.2.2.1 AnyTypeSpecifier.Category Property

Category of the AnyTypeSpecifier [1633]

Namespace: TwinCAT.TypeSystem [1622]

Assembly: TwinCAT.Ads.Abstracts (in TwinCAT.Ads.Abstracts.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public DataTypeCategory Category { get; }
```

Property Value

Type: DataTypeCategory [1649]
6.11.2.2.2  **AnyTypeSpecifier.DimLengths Property**

List of jagged Dimensions (Arrays and jagged arrays)

**Namespace**: TwinCAT.TypeSystem

**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public IList<IDimensionCollection> DimLengths { get; }
```

**Property Value**

Type: `IList<IDimensionCollection>`

**Reference**

AnyTypeSpecifier Class [1633]

TwinCAT.TypeSystem Namespace [1622]

6.11.2.2.3  **AnyTypeSpecifier.ElementType Property**

The element type (Arrays)

**Namespace**: TwinCAT.TypeSystem

**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public AnyTypeSpecifier ElementType { get; }
```

**Property Value**

Type: `AnyTypeSpecifier`

**Reference**

AnyTypeSpecifier Class [1633]

TwinCAT.TypeSystem Namespace [1622]

6.11.2.2.4  **AnyTypeSpecifier.StrLen Property**

The String length (only for `String`)

**Reference**

AnyTypeSpecifier Class [1633]

TwinCAT.TypeSystem Namespace [1622]
**TwinCAT.Ads Namespaces**

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)  
Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int StrLen { get; }
```

**Property Value**

Type: **Int32**

**Reference**

AnyTypeSpecifier Class [1633]  
TwinCAT.TypeSystem Namespace [1622]

6.11.2.2.5 AnyTypeSpecifier.Type Property

Managed type

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)  
Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Type Type { get; }
```

**Property Value**

Type: **Type**

**Reference**

AnyTypeSpecifier Class [1633]  
TwinCAT.TypeSystem Namespace [1622]

6.11.2.3 AnyTypeSpecifier Methods

The AnyTypeSpecifier [1633] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="emoji" alt="" /></td>
<td>Equals</td>
</tr>
<tr>
<td><img src="emoji" alt="" /></td>
<td>Finalize</td>
</tr>
<tr>
<td><img src="emoji" alt="" /></td>
<td>GetHashCode</td>
</tr>
</tbody>
</table>

Determines whether the specified object is equal to the current object. (Inherited from Object.)

Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

Serves as the default hash function. (Inherited from Object.)
### TwinCAT.Ads Namespaces

#### TwinCAT.Ads Exception [57]

#### CannotAccessVirtualSymbolException Class

**Cannot access virtual Symbol**

**Inheritance Hierarchy**

```
System.Object
  System.Exception
    TwinCAT.AdsException [57]
      TwinCAT.TypeSystem.SymbolException [2401]
        TwinCAT.TypeSystem.CannotAccessVirtualSymbolException
```

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
[SerializableAttribute]
public sealed class CannotAccessVirtualSymbolException : SymbolException
```

The CannotAccessVirtualSymbolException type exposes the following members.

#### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Data</code></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><code>HelpLink</code></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><code>HRESULT</code></td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><code>InnerException</code></td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

---

**Reference**

- AnyTypeSpecifier Class [1633]
- TwinCAT.TypeSystem Namespace [1622]
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstancePath</td>
<td>Gets the instance path. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.3.1 CannotAccessVirtualSymbolException Constructor

Initializes a new instance of the CannotAccessVirtualSymbolException [1641] class.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public CannotAccessVirtualSymbolException(
    ISymbol symbol
)
```

**Parameters**

- **symbol**
  - Type: TwinCAT.TypeSystem.ISymbol [2176]
  - The symbol.
Reference

CannotAccessVirtualSymbolException Class [1641]

TwinCAT.TypeSystem Namespace [1622]

6.11.3.2 CannotAccessVirtualSymbolException Properties

The CannotAccessVirtualSymbolException [1641] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the instance path. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

CannotAccessVirtualSymbolException Class [1641]

TwinCAT.TypeSystem Namespace [1622]

6.11.3.3 CannotAccessVirtualSymbolException Methods

The CannotAccessVirtualSymbolException [1641] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Reference

CannotAccessVirtualSymbolException Class [1641]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.4 CannotResolveDataTypeException Class

Class CannotResolveDataTypeException. Implements the DataTypeException [1658]

### Inheritance Hierarchy

System Object  
  System.Exception  
    TwinCAT.AdsException [57]  
      TwinCAT.TypeSystem.DataTypeException [1658]  
        TwinCAT.TypeSystem.CannotResolveDataTypeException

Namespace: TwinCAT.TypeSystem [1622]  
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

C#  

```csharp
[SerializableAttribute]
public class CannotResolveDataTypeException : DataTypeException
```

The CannotResolveDataTypeException type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotResolveDataTypeException(String) [1646]</td>
<td>Initializes a new instance of the CannotResolveDataTypeException class.</td>
</tr>
<tr>
<td>CannotResolveDataTypeException(IInstance) [1647]</td>
<td>Initializes a new instance of the CannotResolveDataTypeException class.</td>
</tr>
<tr>
<td>CannotResolveDataTypeException(SerializationInfo, StreamingContext) [1647]</td>
<td>Initializes a new instance of the CannotResolveDataTypeException class.</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the type of the data. (Inherited from <code>DataTypeException</code>.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the type. (Inherited from <code>DataTypeException</code>.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the <code>Exception</code> that is the root cause of one or more subsequent exceptions. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the <code>SerializationInfo</code> with information about the exception. (Inherited from <code>DataTypeException</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>
6.11.4.1 CannotResolveDataTypeException Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotResolveDataTypeException(String)</td>
<td>Initializes a new instance of the CannotResolveDataTypeException class.</td>
</tr>
<tr>
<td>CannotResolveDataTypeException(Instance)</td>
<td>Initializes a new instance of the CannotResolveDataTypeException class.</td>
</tr>
<tr>
<td>CannotResolveDataTypeException(SerializationInfo, StreamingContext)</td>
<td>Initializes a new instance of the CannotResolveDataTypeException class.</td>
</tr>
</tbody>
</table>

Reference

CannotResolveDataTypeException Class [1644]
TwinCAT.TypeSystem Namespace [1622]

6.11.4.1.1 CannotResolveDataTypeException Constructor (String)

Initializes a new instance of the CannotResolveDataTypeException class.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public CannotResolveDataTypeException(
    string typeName
)

Parameters

typeName Type: System.String
Name of the type.

Reference

CannotResolveDataTypeException Class [1644]
6.11.4.1.2 CannotResolveDataTypeException Constructor (IInstance)

Initializes a new instance of the CannotResolveDataTypeException [1644] class.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public CannotResolveDataTypeException(
    IInstance symbol
)

Parameters

symbol Type: TwinCAT.TypeSystem.IInstance [2052]
The symbol.

Reference

CannotResolveDataTypeException Class [1644]
CannotResolveDataTypeException Overload [1646]
TwinCAT.TypeSystem Namespace [1622]

6.11.4.1.3 CannotResolveDataTypeException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the CannotResolveDataTypeException [1644] class.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

protected CannotResolveDataTypeException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)

Parameters

serializationInfo Type: System.Runtime.Serialization.SerializationInfo
The serialization information.
streamingContext Type: System.Runtime.Serialization.StreamingContext
The streaming context.
6.11.4.2 CannotResolveDataTypeException Properties

The CannotResolveDataTypeException type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the type of the data. (Inherited from DataTypeException.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the type. (Inherited from DataTypeException.)</td>
</tr>
</tbody>
</table>

Reference

CannotResolveDataTypeException Class [1644]
TwinCAT.TypeSystem Namespace [1622]

6.11.4.3 CannotResolveDataTypeException Methods

The CannotResolveDataTypeException type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root</td>
</tr>
<tr>
<td></td>
<td>cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with</td>
</tr>
<tr>
<td></td>
<td>information about the exception. (Inherited from DataTypeException</td>
</tr>
<tr>
<td></td>
<td>[1658].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Exception.)</td>
</tr>
</tbody>
</table>

**Reference**

CannotResolveDataTypeException Class [1644]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.4 CannotResolveDataTypeException Events

The CannotResolveDataTypeException [1644] type exposes the following members.

**Events**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object</td>
</tr>
<tr>
<td></td>
<td>that contains serialized data about the exception. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Exception.)</td>
</tr>
</tbody>
</table>

**Reference**

CannotResolveDataTypeException Class [1644]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.5 DataTypeCategory Enumeration

Category of a DataType / Instance

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public enum DataTypeCategory
```

**Members**

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>Uninitialized / NotProcessed (0)</td>
</tr>
<tr>
<td>Member name</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Primitive</td>
<td>1</td>
<td>Simple / Base Data Type (1)</td>
</tr>
<tr>
<td>Alias</td>
<td>2</td>
<td>Alias data type (2)</td>
</tr>
<tr>
<td>Enum</td>
<td>3</td>
<td>Enumeration data type (3)</td>
</tr>
<tr>
<td>Array</td>
<td>4</td>
<td>Array data type (4)</td>
</tr>
<tr>
<td>Struct</td>
<td>5</td>
<td>Structure data type (5)</td>
</tr>
<tr>
<td>FunctionBlock</td>
<td>6</td>
<td>Function block (POU) (6)</td>
</tr>
<tr>
<td>Program</td>
<td>7</td>
<td>Program (POU) (7)</td>
</tr>
<tr>
<td>Function</td>
<td>8</td>
<td>Function (POU) (8)</td>
</tr>
<tr>
<td>SubRange</td>
<td>9</td>
<td>SubRange (9)</td>
</tr>
<tr>
<td>String</td>
<td>10</td>
<td>Fixed length string (10)</td>
</tr>
<tr>
<td>Bitset</td>
<td>12</td>
<td>Bitset (12)</td>
</tr>
<tr>
<td>Pointer</td>
<td>13</td>
<td>Pointer type (13)</td>
</tr>
<tr>
<td>Union</td>
<td>14</td>
<td>Union type (14)</td>
</tr>
<tr>
<td>Reference</td>
<td>15</td>
<td>Reference type (15)</td>
</tr>
<tr>
<td>Interface</td>
<td>16</td>
<td>The interface</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]

6.11.6 DataDictionary Class

Collection of DataTypes [1986]

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Generic.DataTypeCollection [2442], IDataType [1986].
  TwinCAT.TypeSystem.DataTypeCollection

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class DataTypeCollection : DataTypeCollection<IDataType>,
  IDataTypeCollection, IDataTypeCollection<IDataType>, ICollection<IDataType>,
  IEnumerable<IDataType>, IEnumerable
```

The DataTypeCollection type exposes the following members.

 Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(DataTypeCollection.</td>
<td>Initializes a new instance of the DataTypeCollection class.</td>
</tr>
<tr>
<td>[1652])</td>
<td></td>
</tr>
<tr>
<td>(DataTypeCollection(IEnumerable&lt;IDataType&gt;)</td>
<td>Initializes a new instance of the DataTypeCollection class (Copy</td>
</tr>
<tr>
<td>[1653])</td>
<td>constructor).</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the count of contained</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether this instance is read only. (Inherited from DataTypeCollection.)</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds the specified item to the collection. (Inherited from DataTypeCollection.)</td>
</tr>
<tr>
<td>AddRange</td>
<td>Adds a range of types (Inherited from DataTypeCollection.)</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>Returns A ReadOnly-Version of the DataTypeCollection.</td>
</tr>
<tr>
<td>Clear</td>
<td>Clears the collection. (Inherited from DataTypeCollection.)</td>
</tr>
<tr>
<td>Clone</td>
<td>Clones this DataTypeCollection (Shallow Copy)</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether this DataTypeCollection contains the specified</td>
</tr>
<tr>
<td>ContainsType</td>
<td>Determines whether the container contains the specified</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the data types to the specified array, starting at the array index. (Inherited from DataTypeCollection.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Gets the enumerator. (Inherited from DataTypeCollection.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the Index of the specified</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an</td>
</tr>
<tr>
<td>LookupType</td>
<td>Determines the specified</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the specified</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### TwinCAT.Ads_namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetType</td>
<td>Tries to get the specified IDataType from the IDataTypeCollection.T.</td>
</tr>
</tbody>
</table>

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Internal list of data types (Inherited from DataTypeCollection.T.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.6.1 DataTypeCollection Constructor

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypeCollection()</td>
<td>Initializes a new instance of the DataTypeCollection [1650] class.</td>
</tr>
<tr>
<td>DataTypeCollection(IEnumerable.IDataType)</td>
<td>Initializes a new instance of the DataTypeCollection [1650] class (Copy constructor).</td>
</tr>
</tbody>
</table>

### Reference

DataTypeCollection Class [1650]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.6.1.1 DataTypeCollection Constructor

Initializes a new instance of the DataTypeCollection [1650] class.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public DataTypeCollection()
```

### Reference

DataTypeCollection Class [1650]

DataTypeCollection Overload [1652]

TwinCAT.TypeSystem Namespace [1622]
6.11.6.1.2 **DataTypeCollection Constructor (IEnumerable.IDataType.)**

Initializes a new instance of the `DataTypeCollection` class (Copy constructor).

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd6dca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public DataTypeCollection(
    IEnumerable<IDataType> coll
)
```

**Parameters**

- `coll`  
  *Type:* System.Collections.Generic.IEnumerable<IDataType>  
  The coll.

**Reference**

- **DataTypeCollection Class**
- **DataTypeCollection Overload**
- **TwinCAT.TypeSystem Namespace**

### 6.11.6.2 **DataTypeCollection Properties**

The `DataTypeCollection` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the count of contained IDataType s. (Inherited from DataTypeCollection.T.)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether this instance is read only. (Inherited from DataTypeCollection.T.)</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the IDataType at the specified index. (Inherited from DataTypeCollection.T.)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the IDataType with the specified name. (Inherited from DataTypeCollection.T.)</td>
</tr>
</tbody>
</table>

**Reference**

- **DataTypeCollection Class**
- **TwinCAT.TypeSystem Namespace**

### 6.11.6.3 **DataTypeCollection Methods**

The `DataTypeCollection` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add [2449]</td>
<td>Adds the specified item to the collection. (Inherited from DataTypeCollection.T. [2442].)</td>
</tr>
<tr>
<td>AddRange [2450]</td>
<td>Adds a range of types (Inherited from DataTypeCollection.T. [2442].)</td>
</tr>
<tr>
<td>AsReadOnly [1654]</td>
<td>Returns A ReadOnly-Version of the DataTypeCollection [1650].</td>
</tr>
<tr>
<td>Clear [2451]</td>
<td>Clears the collection. (Inherited from DataTypeCollection.T. [2442].)</td>
</tr>
<tr>
<td>Clone [1655]</td>
<td>Clones this DataTypeCollection [1650] (Shallow Copy)</td>
</tr>
<tr>
<td>Contains [2451]</td>
<td>Determines whether this DataTypeCollection [1650] contains the specified IDataType [1986]. (Inherited from DataTypeCollection.T. [2442].)</td>
</tr>
<tr>
<td>ContainsType [2452]</td>
<td>Determines whether the container contains the specified IDataType [1986]. (Inherited from DataTypeCollection.T. [2442].)</td>
</tr>
<tr>
<td>CopyTo [2453]</td>
<td>Copies the data types to the specified array, starting at the array index. (Inherited from DataTypeCollection.T. [2442].)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator [2453]</td>
<td>Gets the enumerator. (Inherited from DataTypeCollection.T. [2442].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf [2454]</td>
<td>Determines the Index of the specified IDataType [1986]. (Inherited from DataTypeCollection.T. [2442].)</td>
</tr>
<tr>
<td>Insert [2454]</td>
<td>Inserts an IDataType [1986] into the DataTypeCollection [1650]. (Inherited from DataTypeCollection.T. [2442].)</td>
</tr>
<tr>
<td>LookupType [2455]</td>
<td>Determines the specified IDataType [1986] (Inherited from DataTypeCollection.T. [2442].)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove [2456]</td>
<td>Removes the specified IDataType [1986]. (Inherited from DataTypeCollection.T. [2442].)</td>
</tr>
<tr>
<td>RemoveAt [2456]</td>
<td>Removes the IDataType [1986] object at the specified index. (Inherited from DataTypeCollection.T. [2442].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetType [2457]</td>
<td>Tries to get the specified IDataType [1986] from the IDataTypeCollection.T. [1995]. (Inherited from DataTypeCollection.T. [2442].)</td>
</tr>
</tbody>
</table>

## Reference

DataTypeCollection Class [1650]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.6.3.1 DataTypeCollection.AsReadOnly Method

Returns A ReadOnly-Version of the DataTypeCollection [1650].
**Namespace:** TwinCAT.TypeSystem [» 1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

## Syntax

**C#**

```csharp
public ReadOnlyDataTypeCollection AsReadOnly()
```

**Return Value**

Type: **ReadOnlyDataTypeCollection** [» 2291]
A read only version of this **DataTypeCollection** [» 1650].

## Reference

**DataTypeCollection Class** [» 1650]
**TwinCAT.TypeSystem Namespace** [» 1622]

### 6.11.6.3.2 **DataTypeCollection.Clone** Method

Clones this **DataTypeCollection** [» 1650] (Shallow Copy)

**Namespace:** TwinCAT.TypeSystem [» 1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public DataTypeCollection Clone()
```

**Return Value**

Type: **DataTypeCollection** [» 1650]
A clone of this **DataTypeCollection** [» 1650].

## Reference

**DataTypeCollection Class** [» 1650]
**TwinCAT.TypeSystem Namespace** [» 1622]

### 6.11.6.4 **DataTypeCollection Fields**

The **DataTypeCollection** [» 1650] type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>list</code></td>
<td>Internal list of data types (Inherited from <strong>DataTypeCollection.T.</strong> [» 2442].)</td>
</tr>
</tbody>
</table>
### DataTypeEventArgs Class

Class `DataTypeEventArgs`.

**Inheritance Hierarchy**

```
System.Object
    ↓
System.EventArgs
        ↓
TwinCAT.TypeSystem.DataTypeEventArgs
```

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
`public class DataTypeEventArgs : EventArgs`

The `DataTypeEventArgs` type exposes the following members.

#### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DataTypeEventArgs</code></td>
<td>Initializes a new instance of the <code>DataTypeEventArgs</code> class.</td>
</tr>
</tbody>
</table>

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DataTypes</code></td>
<td>The data types</td>
</tr>
</tbody>
</table>

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]
System.EventArgs

6.11.7.1 **DataTypeEventArgs Constructor**

Initializes a new instance of the `DataTypeEventArgs` class.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#:

```csharp
public DataTypeEventArgs(
    IEnumerable<IDataType> types
)
```

**Parameters**

- `types` Type: `System.Collections.Generic.IEnumerable<IDataType>` The types.

**Reference**

- `DataTypeEventArgs Class`:
- `TwinCAT.TypeSystem Namespace`

6.11.7.2 **DataTypeEventArgs Properties**

The `DataTypeEventArgs` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypes</td>
<td>The data types</td>
</tr>
</tbody>
</table>

**Reference**

- `DataTypeEventArgs Class`:
- `TwinCAT.TypeSystem Namespace`

6.11.7.2.1 **DataTypeEventArgs.DataTypes Property**

The data types

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#:

```csharp
public IEnumerable<IDataType> DataTypes { get; }
```
Property Value
Type: `IEnumerable<IDataType>`

Reference
DataTypeEventArgs Class [1656]
TwinCAT.TypeSystem Namespace [1622]

6.11.7.3  DataTypeEventArgs Methods
The `DataTypeEventArgs` type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

Reference
DataTypeEventArgs Class [1656]
TwinCAT.TypeSystem Namespace [1622]

6.11.8  DataTypeException Class
Data Type Exception

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [57]
      TwinCAT.TypeSystem.DataTypeException
        TwinCAT.TypeSystem.CannotResolveDataTypeException [1644]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
```csharp
[SerializableAttribute]
public class DataTypeException : AdsException
```
The DataTypeException type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DataTypeException</strong>(</td>
<td>[SerializationInfo, StreamingContext]</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DataTypeException</strong>(</td>
<td>[String, String]</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DataTypeException</strong>(</td>
<td>[String, IDataType]</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>DataType</strong></td>
<td>Gets the type of the data.</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>HResult</strong></td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>TypeName</strong></td>
<td>Gets the name of the type.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.8.1 DataException Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataException(SerializationInfo, StreamingContext)</td>
<td>Initializes a new instance of the DataException [1658] class.</td>
</tr>
<tr>
<td>DataException(String, String)</td>
<td>Initializes a new instance of the DataException [1658] class.</td>
</tr>
<tr>
<td>DataException(String, IDataType)</td>
<td>Initializes a new instance of the DataException [1658] class.</td>
</tr>
</tbody>
</table>

### Reference

DataException Class [1658]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.8.1.1 DataException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the DataException [1658] class.
Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

protected DataTypeException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)

Parameters

serializationInfo Type: System.Runtime.Serialization.SerializationInfo
The serialization information.

streamingContext Type: System.Runtime.Serialization.StreamingContext
The streaming context.

Reference

DataTypeException Class [1658]
DataTypeException Overload [1660]
TwinCAT.TypeSystem Namespace [1622]

6.11.8.1.2  **DataTypeException Constructor (String, String)**

Initializes a new instance of the **DataTypeException** [1658] class.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

public DataTypeException(
    string message,
    string type
)

Parameters

message Type: System.String
The message.

type Type: System.String
The type.

Reference

DataTypeException Class [1658]
DataTypeException Overload [1660]
TwinCAT.TypeSystem Namespace [1622]
6.11.8.1.3 **DataTypeException Constructor (String, IDataType)**

Initializes a new instance of the `DataTypeException` class.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public DataTypeException(
    string message,
    IDataType type
)
```

**Parameters**

- `message` Type: `System.String`
The message.
- `type` Type: `TwinCAT.TypeSystem.IDataType`
The type.

**Reference**

- [DataTypeException Class](#)
- [DataTypeException Overload](#)
- [TwinCAT.TypeSystem Namespace](#)

6.11.8.2 **DataTypeException Properties**

The `DataTypeException` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><code>DataType</code></td>
<td>Gets the type of the data.</td>
</tr>
<tr>
<td><code>HelpLink</code></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><code>HResult</code></td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><code>InnerException</code></td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><code>Message</code></td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><code>Source</code></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><code>StackTrace</code></td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><code>TargetSite</code></td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the type.</td>
</tr>
</tbody>
</table>

**Reference**

*DataTypeException Class [1658]*

*TwinCAT.TypeSystem Namespace [1622]*

### 6.11.8.2.1 `DataTypeException.DataType` Property

Gets the type of the data.

**Namespace:** `TwinCAT.TypeSystem [1622]`

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```
public IDataType DataType { get; }
```

**Property Value**

Type: `IDataType [1986]`
The type of the data.

**Reference**

*DataTypeException Class [1658]*

*TwinCAT.TypeSystem Namespace [1622]*

### 6.11.8.2.2 `DataTypeException.TypeName` Property

Gets the name of the type.

**Namespace:** `TwinCAT.TypeSystem [1622]`

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```
public string TypeName { get; }
```

**Property Value**

Type: `String`
The name of the type.

**Reference**

*DataTypeException Class [1658]*

*TwinCAT.TypeSystem Namespace [1622]*
### 6.11.8.3  DataTypeException Methods

The `DataTypeException` type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetBaseException</code></td>
<td>When overridden in a derived class, returns the <code>Exception</code> that is the root cause of one or more subsequent exceptions. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetObjectData</code></td>
<td>When overridden in a derived class, sets the <code>SerializationInfo</code> with information about the exception. (Overrides <code>Exception.GetObjectData(SerializationInfo, StreamingContext)</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the runtime type of the current instance. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

#### Reference

- **DataTypeException Class**: [1658]
- **TwinCAT.TypeSystem Namespace**: [1622]

### 6.11.8.3.1  DataTypeException.GetObjectData Method

When overridden in a derived class, sets the `SerializationInfo` with information about the exception.

**Namespace**: TwinCAT.TypeSystem [1622]

**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch/releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>info</code></td>
<td><code>System.Runtime.Serialization.SerializationInfo</code></td>
<td>The <code>SerializationInfo</code> that holds the serialized object data about the exception being thrown.</td>
</tr>
</tbody>
</table>
6.11.8.4  

**DataException Events**

The **DataException** type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object</td>
</tr>
<tr>
<td></td>
<td>that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

6.11.9  

**DataypeNameEventArgs Class**

Class `DataypeNameEventArgs`.

Inheritance Hierarchy

- `System.Object`
  - `System.EventArgs`
    - `TwinCAT.TypeSystem.DataTypeNameEventArgs`

**Namespace:** TwinCAT.TypeSystem  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public class DataypeNameEventArgs : EventArgs
```

The `DataypeNameEventArgs` type exposes the following members.

**Constructors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataypeNameEventArgs</td>
<td>Initializes a new instance of the <code>DataypeNameEventArgs</code> class.</td>
</tr>
</tbody>
</table>
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TypeName</td>
<td>The type name</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]
System.EventArgs

6.11.9.1  DataTypeNameEventArgs Constructor

Initializes a new instance of the DataTypeNameEventArgs [1665] class.

Namespace:  TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public DataTypeNameEventArgs(
    string typeName
)
```

Parameters

typeName  Type: System.String  
Name of the type.

Reference

DataTypeNameEventArgs Class [1665]
TwinCAT.TypeSystem Namespace [1622]

6.11.9.2  DataTypeNameEventArgs Properties

The DataTypeNameEventArgs [1665] type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TypeName [1667]</td>
<td>The type name</td>
</tr>
</tbody>
</table>

Reference

DataTypeNameEventArgs Class [1665]
TwinCAT.TypeSystem Namespace [1622]

6.11.9.2.1  DataTypeNameEventArgs.TypeName Property

The type name

Namespace:  TwinCAT.TypeSystem [1622]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public string TypeName { get; }  

Property Value

Type: String

Reference

DataTypeNameEventArgs Class [1665]
TwinCAT.TypeSystem Namespace [1622]

6.11.9.3  DataTypeNameEventArgs Methods

The DataTypeNameEventArgs [1665] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>(Inherited from Object.)</td>
<td></td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td>operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
<td></td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
6.11.10 Dimension Class

Represents a single dimension of an IArrayType.

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Dimension

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class Dimension : IDimension
```

The Dimension type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ElementCount</td>
<td>Gets the number of elements within that IDimension.</td>
</tr>
<tr>
<td>LowerBound</td>
<td>Gets the lower bound of elements within that Dimension.</td>
</tr>
<tr>
<td>UpperBound</td>
<td>Gets the upper bound of elements within this Dimension.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace
6.11.10.1  Dimension Properties

The Dimension[1668] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ElementCount</td>
<td>Gets the number of elements within that Dimension[1998].</td>
</tr>
<tr>
<td>LowerBound</td>
<td>Gets the lower bound of elements within that Dimension[1998].</td>
</tr>
<tr>
<td>UpperBound</td>
<td>Gets the upper bound of elements within this Dimension[1668].</td>
</tr>
</tbody>
</table>

Reference

Dimension Class[1668]

TwinCAT.TypeSystem Namespace[1622]

6.11.10.1.1  Dimension.ElementCount Property

Gets the number of elements within that Dimension[1998].

Namespace: TwinCAT.TypeSystem[1622]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int ElementCount { get; }
```

Property Value

Type: Int32

The element count.

Implements

IDimension.ElementCount[1999]

Reference

Dimension Class[1668]

TwinCAT.TypeSystem Namespace[1622]

6.11.10.1.2  Dimension.LowerBound Property

Gets the lower bound of elements within that Dimension[1998].

Namespace: TwinCAT.TypeSystem[1622]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public int LowerBound { get; }
```

**Property Value**

Type: Int32
The lower bound.

**Implements**

[IDimension.LowerBound [1999]]

**Reference**

Dimension Class [1668]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.10.1.3 Dimension.UpperBound Property

Gets the upper bound of elements within this Dimension [1668]

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public int UpperBound { get; }
```

**Property Value**

Type: Int32
The upper bound.

**Reference**

Dimension Class [1668]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.10.2 Dimension Methods

The Dimension [1668] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Reference**

Dimension Class [1668]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.11 DimensionCollection Class

Collection class for Array Dimensions

**Inheritance Hierarchy**

System Object

TwinCAT.TypeSystem.DimensionCollection

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public class DimensionCollection : IDimensionCollection,
    IList<IDimension>, ICollection<IDimension>, IEnumerable<IDimension>,
    IEnumerable
```

The DimensionCollection type exposes the following members.

**Constructors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DimensionCollection</td>
<td>Initializes a new instance of the DimensionCollection class.</td>
</tr>
<tr>
<td>DimensionCollection(Int32)</td>
<td>Initializes a new instance of an 1-Dimensional representing DimensionCollection class.</td>
</tr>
<tr>
<td>DimensionCollection(Int32)</td>
<td>Initializes a new instance of the DimensionCollection class.</td>
</tr>
<tr>
<td>DimensionCollection(Int32)</td>
<td>Initializes a new instance of the DimensionCollection class.</td>
</tr>
<tr>
<td>DimensionCollection(IEnumerable.IDimension)</td>
<td>Initializes a new instance of the DimensionCollection class.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [1675]</td>
<td>Gets the number of elements contained in the ICollection.T.</td>
</tr>
<tr>
<td>ElementCount [1676]</td>
<td>Gets the Number of elements in all Dimensions</td>
</tr>
<tr>
<td>IsReadOnly [1677]</td>
<td>Gets a value indicating whether the ICollection.T. is read-only.</td>
</tr>
<tr>
<td>Item [1677]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>LowerBounds [1678]</td>
<td>Gets the lower bounds.</td>
</tr>
<tr>
<td>UpperBounds [1678]</td>
<td>Gets the upper bounds.</td>
</tr>
</tbody>
</table>

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add [1680]</td>
<td>Adds an item to the ICollection.T.</td>
</tr>
<tr>
<td>AsReadOnly [1680]</td>
<td>Returns a read only version of this DimensionCollection.</td>
</tr>
<tr>
<td>Clear [1681]</td>
<td>Removes all items from the ICollection.T.</td>
</tr>
<tr>
<td>Contains [1682]</td>
<td>Determines whether the ICollection.T. contains a specific value.</td>
</tr>
<tr>
<td>CopyTo [1682]</td>
<td>Copies to.</td>
</tr>
<tr>
<td>Empty [1683]</td>
<td>Gets an empty DimensionCollection</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetDimensionLengths</td>
<td>Gets an array the specifies the Lengths of each Array Dimension</td>
</tr>
<tr>
<td>GetEnumerator [1684]</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf [1685]</td>
<td>Determines the index of a specific item in the IList.T.</td>
</tr>
<tr>
<td>Insert [1685]</td>
<td>Inserts an item to the IList.T at the specified index.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove [1686]</td>
<td>Removes the first occurrence of a specific object from the ICollection.T.</td>
</tr>
<tr>
<td>RemoveAt [1687]</td>
<td>Removes the IList.T. item at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### 6.11.11.1 DimensionCollection Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DimensionCollection()</td>
<td>Initializes a new instance of the DimensionCollection class.</td>
</tr>
<tr>
<td>DimensionCollection(Int32)</td>
<td>Initializes a new instance of an 1-Dimensional representing DimensionCollection class.</td>
</tr>
<tr>
<td>DimensionCollection(Int32)</td>
<td>Initializes a new instance of the DimensionCollection class.</td>
</tr>
<tr>
<td>DimensionCollection(IEnumerable.IDimension)</td>
<td>Initializes a new instance of the DimensionCollection class.</td>
</tr>
</tbody>
</table>

Reference

DimensionCollection Class [1671]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.11.1.1 DimensionCollection Constructor

Initializes a new instance of the DimensionCollection class.

**Namespace:** TwinCAT.TypeSystem [1671]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public DimensionCollection()
```

Reference

DimensionCollection Class [1671]

DimensionCollection Overload [1673]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.11.1.2 DimensionCollection Constructor (Int32)

Initializes a new instance of an 1-Dimensional representing DimensionCollection class.
**Namespace:** TwinCAT.TypeSystem [►1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public DimensionCollection(
    int length
)
```

**Parameters**

- **length**  
  Type: System.Int32  
  The length.

**Reference**

- DimensionCollection Class [►1671]
- DimensionCollection Overload [►1673]
- TwinCAT.TypeSystem Namespace [►1622]

### 6.11.11.1.3 DimensionCollection Constructor (.Int32.)

Initializes a new instance of the DimensionCollection [►1671] class.

**Namespace:** TwinCAT.TypeSystem [►1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public DimensionCollection(
    int[] dimLengths
)
```

**Parameters**

- **dimLengths**  
  Type: System.Int32.  
  The dim lengths.

**Reference**

- DimensionCollection Class [►1671]
- DimensionCollection Overload [►1673]
- TwinCAT.TypeSystem Namespace [►1622]

### 6.11.11.1.4 DimensionCollection Constructor (IEnumerable.IDimension.)

Initializes a new instance of the DimensionCollection [►1671] class.
TwinCAT.Ads Namespaces

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public DimensionCollection(
    IEnumerable<IDimension> coll
)
```

### Parameters

- **coll**
  - The coll.

### Reference

- **DimensionCollection Class** [1671]
- **DimensionCollection Overload** [1673]
- **TwinCAT.TypeSystem Namespace** [1622]

### 6.11.11.2 DimensionCollection Properties

The **DimensionCollection** [1671] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [1675]</td>
<td>Gets the number of elements contained in the ICollection.T.</td>
</tr>
<tr>
<td>ElementCount [1676]</td>
<td>Gets the Number of elements in all Dimensions</td>
</tr>
<tr>
<td>IsReadOnly [1677]</td>
<td>Gets a value indicating whether the ICollection.T. is read-only.</td>
</tr>
<tr>
<td>Item [1677]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>LowerBounds [1678]</td>
<td>Gets the lower bounds.</td>
</tr>
<tr>
<td>UpperBounds [1678]</td>
<td>Gets the upper bounds.</td>
</tr>
</tbody>
</table>

### Reference

- **DimensionCollection Class** [1671]
- **TwinCAT.TypeSystem Namespace** [1622]

#### 6.11.11.2.1 DimensionCollection.Count Property

Gets the number of elements contained in the ICollection.T.
### DimensionCollection.ElementCount Property

Gets the Number of elements in all Dimensions

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public int ElementCount { get; }
```

**Property Value**

Type: `Int32`
The count.

**Implements**

`IDimensionCollection.ElementCount`

**Reference**

DimensionCollection Class

---

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public int Count { get; }
```

**Property Value**

Type: `Int32`
The count.

**Implements**

`ICollection.T.Count`

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

DimensionCollection Class

---

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public int ElementCount { get; }
```

**Property Value**

Type: `Int32`
The count.

**Implements**

`IDimensionCollection.ElementCount`

**Reference**

DimensionCollection Class
### DimensionCollection.IsReadOnly Property

Gets a value indicating whether the ICollection is read-only.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool IsReadOnly { get; }
```

**Property Value**

_Type:_ Boolean

_true if this instance is read only; otherwise, false._

**Implements**

ICollection<T>.IsReadOnly

---

### DimensionCollection.Item Property

Gets or sets the element at the specified index.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public IDimension this[int index] { get; set; }
```

**Parameters**

_index_  
_Type:_ System.Int32

_The index._

**Return Value**

_Type:_ IDimension

_IDimension._
**Implements**

[IList<T>.Item.Int32.](#)

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

**Remarks**

Index = 0 is the lowest dimension.

**Reference**

DimensionCollection Class [» 1671]

TwinCAT.TypeSystem Namespace [» 1622]

### 6.11.11.2.5 DimensionCollection.LowerBounds Property

Gets the lower bounds.

**Namespace:** TwinCAT.TypeSystem [» 1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int[] LowerBounds { get; }
```

**Property Value**

Type: `Int32`.

The lower bounds.

**Implements**

[IDimensionCollection.LowerBounds](#)

**Reference**

DimensionCollection Class [» 1671]

TwinCAT.TypeSystem Namespace [» 1622]

### 6.11.11.2.6 DimensionCollection.UpperBounds Property

Gets the upper bounds.

**Namespace:** TwinCAT.TypeSystem [» 1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

public int[] UpperBounds { get; }

Property Value

Type: Int32.
The upper bounds.

Implements

IDimensionCollection.UpperBounds [› 2002]

Reference

DimensionCollection Class [› 1671]

TwinCAT.TypeSystem Namespace [› 1622]

6.11.11.3 DimensionCollection Methods

The DimensionCollection [› 1671] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection.T.</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>Returns a read only version of this DimensionCollection [› 1671].</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T.</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the ICollection.T contains a specific value.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies to.</td>
</tr>
<tr>
<td>Empty</td>
<td>Gets an empty DimensionCollection [› 1671]</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetDimensionLengths</td>
<td>Gets an array the specifies the Lengths of each Array Dimension</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList.T.</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList.T. at the specified index.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove [1686]</td>
<td>Removes the first occurrence of a specific object from the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>RemoveAt [1687]</td>
<td>Removes the IList&lt;T&gt; item at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Reference**

DimensionCollection Class [1671]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.11.3.1 DimensionCollection.Add Method

Adds an item to the ICollection<T>.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void Add(
    IDimension item
)
```

**Parameters**

- **item**
  - Type: TwinCAT.TypeSystem.IDimension [1998]
  - The object to add to the ICollection<T>.

**Implements**

ICollection<T>.Add(T)

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

DimensionCollection Class [1671]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.11.3.2 DimensionCollection.AsReadOnly Method

Returns a read only version of this DimensionCollection [1671].
**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public ReadOnlyDimensionCollection AsReadOnly()
```

### Field Value

**Type:** ReadOnlyDimensionCollection [2295]

As read only.

### Return Value

**Type:** ReadOnlyDimensionCollection [2295]

ReadOnlyDimensionCollection.

### Reference

DimensionCollection Class [1671]

TwinCAT.TypeSystem Namespace [1622]

---

### DimensionCollection.Clear Method

Removes all items from the ICollection<T>.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public void Clear()
```

#### Implements

ICollection<T>.Clear.

#### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

#### Reference

DimensionCollection Class [1671]

TwinCAT.TypeSystem Namespace [1622]
6.11.11.3.4 DimensionCollection.Contains Method

Determines whether the ICollection{T} contains a specific value.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# 

```csharp
public bool Contains(
    IDimension item
)
```

**Parameters**

- **item**
  - Type: TwinCAT.TypeSystem.IDimension [1998]
  - The object to locate in the ICollection{T}.

**Return Value**

- Type: Boolean
  - true if item is found in the ICollection{T}; otherwise, false.

**Implements**

- ICollection{T}.Contains(T)

**Exceptions**

- NotImplementedException

**Reference**

- DimensionCollection Class [1671]
- TwinCAT.TypeSystem Namespace [1622]

6.11.11.3.5 DimensionCollection.CopyTo Method

Copies to.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# 

```csharp
public void CopyTo(
    IDimension[] array,
    int arrayIndex
)
```
TwinCAT.Ads Namespaces

Parameters

array Type: TwinCAT.TypeSystem.IDimension
The array.
arrayIndex Type: System.Int32
Index of the array.

Implements

ICollection.T.CopyTo(T, Int32)

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

DimensionCollection Class [1671]
TwinCAT.TypeSystem Namespace [1622]

6.11.11.3.6 DimensionCollection.Empty Method

Gets an empty DimensionCollection [1671]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static DimensionCollection Empty()

Field Value

Type: DimensionCollection [1671]
The Empty collection.

Reference

DimensionCollection Class [1671]
TwinCAT.TypeSystem Namespace [1622]

6.11.11.3.7 DimensionCollection.GetDimensionLengths Method

Gets an array the specifies the Lengths of each Array Dimension

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public int[] GetDimensionLengths()
```

Return Value

Type: .Int32,
System.Int32[].

Implements

[IDimensionCollection.GetDimensionLengths. [2004]

Reference

DimensionCollection Class [1671]

TwinCAT.TypeSystem Namespace [1622]

6.11.11.3.8 DimensionCollection.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public IEnumerator<IDimension> GetEnumerator()
```

Return Value

Type: IEnumerator.IDimension
A IEnumerator.T. that can be used to iterate through the collection.

Implements

IEnumerator.T.GetEnumerator.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

DimensionCollection Class [1671]

TwinCAT.TypeSystem Namespace [1622]
6.11.3.9 DimensionCollection.IndexOf Method

Determines the index of a specific item in the IList<T>.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int IndexOf(
    IDimension item
)
```

**Parameters**

- **item**  
  Type: TwinCAT.TypeSystem.IDimension  
  The object to locate in the IList<T>.

**Return Value**

- **Type:** Int32  
  The index of item if found in the list; otherwise, -1.

**Implements**

IList<T>.IndexOf(T)

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

- DimensionCollection Class
- TwinCAT.TypeSystem Namespace

6.11.13.10 DimensionCollection.Insert Method

Inserts an item to the IList<T> at the specified index.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void Insert(
    int index,
    IDimension item
)
```
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>System.Int32</td>
<td>The zero-based index at which item should be inserted.</td>
</tr>
<tr>
<td>item</td>
<td>TwinCAT.TypeSystem.IDimension</td>
<td>The object to insert into the <code>IList&lt;T&gt;</code></td>
</tr>
</tbody>
</table>

Implements

- `IList<T>.Insert(Int32, T)`

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

- DimensionCollection Class [1671]
- TwinCAT.TypeSystem Namespace [1622]

6.11.11.3.11 DimensionCollection.Remove Method

Removes the first occurrence of a specific object from the `ICollection<T>`.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool Remove(
    IDimension item
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>item</td>
<td>TwinCAT.TypeSystem.IDimension</td>
<td>The object to remove from the <code>ICollection&lt;T&gt;</code></td>
</tr>
</tbody>
</table>

Return Value

Type: `Boolean`
true if item was successfully removed from the `ICollection<T>`; otherwise, false. This method also returns false if item is not found in the original `ICollection<T>`.

Implements

- `ICollection<T>.Remove(T)`
6.11.11.3.12 DimensionCollection.RemoveAt Method

Removes the IList<T> item at the specified index.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public void RemoveAt(
    int index
)

Parameters

index Type: System.Int32
The zero-based index of the item to remove.

Implements

IList<T>.RemoveAt(Int32)

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

DimensionCollection Class [1671]
TwinCAT.TypeSystem Namespace [1622]

6.11.12 DynamicAliasInstance Class

Class DynamicAliasInstance. This class cannot be inherited.
Inheritance Hierarchy

```csharp
System.Dynamic.DynamicObject
    TwinCAT.TypeSystem.DynamicSymbol
        TwinCAT.TypeSystem.DynamicAliasInstance
```

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public sealed class DynamicAliasInstance : DynamicSymbol, IAliasInstance, ISymbol, IAttributedInstance, IInstance, IBitSize
```

The `DynamicAliasInstance` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_InnerSymbol</td>
<td>Gets the inner symbol of this <code>DynamicSymbol</code> (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>AccessRights</td>
<td>Gets the access rights. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the Symbol Attributes (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>IDataType</code> in bits. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the category. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the <code>Instance</code> (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the connection bound to this <code>DynamicSymbol</code> (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the <code>IDataType</code> of the <code>Instance</code> (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.) (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full <code>DataType</code> but instead of some sort of bit mapping (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this <code>ISymbol</code> is persistent. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsPointer [1807]</td>
<td>Indicates that the Instance [2052] represents a Pointer type (Pointer TO)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPrimitiveType [1807]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReadOnly [1808]</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol</td>
</tr>
<tr>
<td></td>
<td>[1791].)</td>
</tr>
<tr>
<td>IsRecursive [1808]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReference [1809]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE</td>
</tr>
<tr>
<td></td>
<td>TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsStatic [1809]</td>
<td>Gets a value indicating whether this instance is static. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>NormalizedName [1810]</td>
<td>Gets the normalized instance name (fixed name for dynamic property access</td>
</tr>
<tr>
<td></td>
<td>that doesn't contain invalid characters), (Inherited from DynamicSymbol</td>
</tr>
<tr>
<td></td>
<td>[1791].)</td>
</tr>
<tr>
<td>NotificationSettings [1810]</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Parent [1811]</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Size [1811]</td>
<td>Gets the size of the Instance [2052] in bytes. (Inherited from DynamicSymbol</td>
</tr>
<tr>
<td></td>
<td>[1791].)</td>
</tr>
<tr>
<td>SubSymbols [1812]</td>
<td>Gets the SubSymbols of the Symbol [2176] (Inherited from DynamicSymbol [1791]</td>
</tr>
<tr>
<td>TypeName [1812]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052].</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueEncoding [1813]</td>
<td>Gets the value encoding. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1816]</td>
<td>Equals (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>GetDynamicMemberNames [1694]</td>
<td>Returns the enumeration of all dynamic member names. (Overrides DynamicSymbol</td>
</tr>
<tr>
<td></td>
<td>GetDynamicMemberNames [1816].)</td>
</tr>
<tr>
<td>GetHashCode [1817]</td>
<td>Gets the HashCode of the Address (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadAnyValue [1825]</td>
<td>Reads the value of this Value [2254] into a new created instance of the</td>
</tr>
<tr>
<td></td>
<td>managed type (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue [1826]</td>
<td>Reads the raw value of the ValueSymbol [2254] (Ads Read / Write) (Inherited</td>
</tr>
<tr>
<td></td>
<td>from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue(Int32) [1826]</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValueAsync [1827]</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol</td>
</tr>
<tr>
<td></td>
<td>[1791].)</td>
</tr>
<tr>
<td>ReadValue [1828]</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol</td>
</tr>
<tr>
<td></td>
<td>[1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the value of this DynamicSymbol (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the IValueSymbol asynchronously. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>TryGetIndex</td>
<td>Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicSymbol.TryGetMember(GetMemberBinder, Object.).)</td>
</tr>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>TrySetIndex</td>
<td>Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.</td>
</tr>
<tr>
<td>TrySetMember</td>
<td>Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.</td>
</tr>
<tr>
<td>TryWriteValue</td>
<td>Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>UpdateAnyValue</td>
<td>Reads the value of this Value into the specified managed value. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteAnyValue</td>
<td>Writes the value represented by the managed value to this Value (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteRawValue(Byte)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteRawValue(Byte, Int32)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteRawValueAsync</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteValue(Object)</td>
<td>Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteValue(Object, Int32)</td>
<td>Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteValueAsync</td>
<td>Writes the Value of the IValueSymbol (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged ![1848]</td>
<td>Occurs when the RawValue of the IValueSymbol ![2254] has changed. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>ValueChanged ![1849]</td>
<td>Occurs when the (Primitive) value of the IValueSymbol ![2254] has changed. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
</tbody>
</table>

## Reference

- TwinCAT.TypeSystem Namespace ![1622]
- TwinCAT.TypeSystem.DynamicSymbol ![1791]
- TwinCAT.TypeSystem.IAliasInstance ![1952]

### 6.11.12.1 DynamicAliasInstance Properties

The DynamicAliasInstance ![1687] type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_InnerSymbol ![1798]</td>
<td>Gets the inner symbol of this DynamicSymbol ![1791] (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>AccessRights ![1799]</td>
<td>Gets the access rights. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>Attributes ![1799]</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>BitSize ![1800]</td>
<td>Gets the size of the IDDataType ![1986] in bits. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>ByteSize ![1800]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>Category ![1801]</td>
<td>Gets the category. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>Comment ![1801]</td>
<td>Gets the comment of the IInstance ![2052] (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>Connection ![1802]</td>
<td>Gets the connection bound to this DynamicSymbol ![1791] (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>DataType ![1802]</td>
<td>Gets the IDDataType ![1986] of the IInstance ![2052]. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>HasValue ![1803]</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>InstanceName ![1803]</td>
<td>Gets the name of the instance (without periods .) (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>InstancePath ![1804]</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>IsBitType ![1805]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>IsByteAligned ![1805]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsContainerType [1806]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPersistent [1806]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPointer [1807]</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPrimitiveType [1807]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReadOnly [1808]</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsRecursive [1808]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReference [1809]</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsStatic [1809]</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>NormalizedName [1810]</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>NotificationSettings [1810]</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Parent [1811]</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Size [1811]</td>
<td>Gets the size of the IInstance [2052] in bytes. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>SubSymbols [1812]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TypeName [1812]</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueEncoding [1813]</td>
<td>Gets the value encoding. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

Reference

DynamicAliasInstance Class [1687]

TwinCAT.TypeSystem Namespace [1622]

6.11.12.2 DynamicAliasInstance Methods

The DynamicAliasInstance [1687] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1816]</td>
<td>Equals (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>GetDynamicMemberNames [1694]</td>
<td>Returns the enumeration of all dynamic member names. (Overrides DynamicSymbol.GetDynamicMemberNames [1816].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>ReadAnyValue</td>
<td>Reads the value of this Value into a new created instance of the managed type (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ReadRawValue</td>
<td>Reads the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ReadRawValueAsync</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Reads the value of this DynamicSymbol. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the value of this DynamicSymbol. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the IValueSymbol asynchronously. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>TryGetIndex</td>
<td>Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrider DynamicSymbol.TryGetMember(GetMemberBinder, Object.)</td>
</tr>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>TrySetIndex</td>
<td>Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.</td>
</tr>
<tr>
<td>TrySetMember</td>
<td>Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.</td>
</tr>
<tr>
<td>TryWriteValue</td>
<td>Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>UpdateAnyValue</td>
<td>Reads the value of this Value into the specified managed value. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteAnyValue</td>
<td>Writes the value represented by the managed value to this Value (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteRawValue(Byte)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WriteRawValue</strong></td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td><strong>WriteRawValueAsync</strong></td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td><strong>WriteValue</strong></td>
<td>Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td><strong>WriteValueAsync</strong></td>
<td>Writes the Value of the IValueSymbol (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>

#### Reference

- DynamicAliasInstance Class
- TwinCAT.TypeSystem Namespace

### 6.11.12.2.1 DynamicAliasInstance.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#  
```c#  
public override IEnumerable<string> GetDynamicMemberNames()  
```

**Return Value**

Type: `IEnumerable<String>`

A sequence that contains dynamic member names.

**Reference**

- DynamicAliasInstance Class
- TwinCAT.TypeSystem Namespace

### 6.11.12.2.2 DynamicAliasInstance.TryGetIndex Method

Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
## TwinCAT.Ads Namespaces

### 6.11.12.2.3 DynamicAliasInstance.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

#### C#

```csharp
public override bool TryGetMember(
    GetMemberBinder binder,
    out Object result
)
```

### Parameters

- **binder**
  - Type: `System.Dynamic.GetIndexBinder`
  - Provides information about the operation.

- **indexes**
  - Type: `System.Object`
  - The indexes that are used in the operation. For example, for the sampleObject[3] operation in C# (sampleObject(3) in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[0] is equal to 3.

- **result**
  - Type: `System.Object`
  - The result of the index operation.

### Return Value

**Type:** `Boolean`

true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

### Reference

- DynamicAliasInstance Class [1687]
- TwinCAT.TypeSystem Namespace [1622]
Parameters

binder Type: System.Dynamic.GetMemberBinder
Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the Console.WriteLine(sampleObject.SampleProperty) statement, where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.

result Type: System.Object
The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicAliasInstance Class [1687]
TwinCAT.TypeSystem Namespace [1622]

6.11.12.2.4 DynamicAliasInstance.TrySetIndex Method

Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public virtual bool TrySetIndex(
    SetIndexBinder binder,
    Object[] indexes,
    Object value
)

Parameters

binder Type: System.Dynamic.SetIndexBinder
Provides information about the operation.

indexes Type: System.Object
The indexes that are used in the operation. For example, for the sampleObject[3] = 10 operation in C# (sampleObject(3) = 10 in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[0] is equal to 3.

value Type: System.Object
The value to set to the object that has the specified index. For example, for the sampleObject[3] = 10 operation in C# (sampleObject(3) = 10 in Visual Basic), where sampleObject is derived from the DynamicObject class, value is equal to 10.
**Return Value**

Type: Boolean  
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

**Reference**

DynamicAliasInstance Class [1687]  
TwinCAT.TypeSystem Namespace [1622]

### 6.11.12.2.5 DynamicAliasInstance.TrySetMember Method

Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public virtual bool TrySetMember(
    SetMemberBinder binder,
    Object value
)
```

**Parameters**

**binder**  
Type: System.Dynamic.SetMemberBinder  
Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member to which the value is being assigned. For example, for the statement sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.

**value**  
Type: System.Object  
The value to set to the member. For example, for sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, the value is "Test".

**Return Value**

Type: Boolean  
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

**Reference**

DynamicAliasInstance Class [1687]  
TwinCAT.TypeSystem Namespace [1622]
6.11.12.3  DynamicAliasInstance Events

The DynamicAliasInstance [1687] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol [2254] has changed.  (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol [2254] has changed. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

Reference

DynamicAliasInstance Class [1687]

TwinCAT.TypeSystem Namespace [1622]

6.11.13  DynamicArrayInstance Class

Dynamic Array Instance

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbol [1791]
    TwinCAT.TypeSystem.DynamicArrayInstance
      TwinCAT.TypeSystem.DynamicOversamplingArrayInstance [1712]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdcdca3e72bc0e15da1c14

Syntax

C#

public class DynamicArrayInstance : DynamicSymbol,
  IArrayInstance, ISymbol, IAttributedInstance, IInstance, IBitSize

The DynamicArrayInstance type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerSymbol</td>
<td>Gets the inner symbol of this DynamicSymbol [1791] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>AccessRights</td>
<td>Gets the access rights. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>AllowGIOLAccess</td>
<td>Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Category [1801]</td>
<td>Gets the category. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Comment [1801]</td>
<td>Gets the comment of the IInstance [2052] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Connection [1802]</td>
<td>Gets the connection bound to this DynamicSymbol [1791] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>DataType [1802]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Dimensions [1704]</td>
<td>Gets the dimensions as read only collection.</td>
</tr>
<tr>
<td>Elements [1704]</td>
<td>Gets the contained Array Elements as read only collection.</td>
</tr>
<tr>
<td>ElementType [1705]</td>
<td>Gets the type of the contained elements.</td>
</tr>
<tr>
<td>HasValue [1803]</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstanceName [1803]</td>
<td>Gets the name of the instance (without periods .) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstancePath [1804]</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsBitType [1805]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsByteAligned [1805]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsContainerType [1806]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPersistent [1806]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPointer [1807]</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPrimitiveType [1807]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReadOnly [1808]</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsRecursive [1808]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReference [1809]</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsStatic [1809]</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Item [1705]</td>
<td>Gets the ISymbol [2176] with the specified indices.</td>
</tr>
<tr>
<td>NormalizedName [1810]</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters). (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>NotificationSettings [1810]</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Parent [1811]</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Size [ Morse 1811 ]</td>
<td>Gets the size of the Instance [ Morse 2052 ] in bytes. (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>SubSymbols [ Morse 1812 ]</td>
<td>Gets the SubSymbols of the ISymbol [ Morse 2176 ] (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>TypeName [ Morse 1812 ]</td>
<td>Gets the name of the DataType [ Morse 1986 ] that is used for this Instance [ Morse 2052 ]. (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>ValueEncoding [ Morse 1813 ]</td>
<td>Gets the value encoding. (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [ Morse 1816 ]</td>
<td>Equals (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>GetDynamicMemberNames [ Morse 1816 ]</td>
<td>Returns the enumeration of all dynamic member names. (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>GetHashCode [ Morse 1817 ]</td>
<td>Gets the HashCode of the Address (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>OnReadAnyValue [ Morse 1817 ]</td>
<td>Handler function for reading ADS 'Any' Values. (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>OnReadRawValue [ Morse 1818 ]</td>
<td>Handler function for reading Raw symbol value. (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>OnReadRawValueAsync [ Morse 1818 ]</td>
<td>Handler function reading the raw value of the DynamicSymbol [ Morse 1791 ]. (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>OnReadValue [ Morse 1819 ]</td>
<td>Handler function for the (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>OnReadValueAsync [ Morse 1820 ]</td>
<td>Handler function reading the DynamicSymbols [ Morse 1791 ] value asynchronously. (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>OnSetInstanceName [ Morse 1820 ]</td>
<td>Sets a new InstanceName InstancePath (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>OnTryReadValue [ Morse 1821 ]</td>
<td>Handler function for the (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>OnTryWriteValue [ Morse 1821 ]</td>
<td>Handler Function for writing value. (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>OnWriteRawValue [ Morse 1822 ]</td>
<td>Handler function for reading symbols raw value. (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>OnWriteRawValueAsync [ Morse 1823 ]</td>
<td>Handler function for writing the raw DynamicSymbol [ Morse 1791 ] value. (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>OnWriteValue [ Morse 1823 ]</td>
<td>Handler Function for writing value. (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>OnWriteValueAsync [ Morse 1824 ]</td>
<td>Handler Function for writing value. (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>ReadAnyValue [ Morse 1825 ]</td>
<td>Reads the value of this Value [ Morse 2254 ] into a new created instance of the managed type (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>ReadRawValue [ Morse 1826 ]</td>
<td>Reads the raw value of the IValueSymbol [ Morse 2254 ] (Ads Read / Write) (Inherited from DynamicSymbol [ Morse 1791 ].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValueAsyc</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValue()</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the IValueSymbol [2254] asynchronously. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TryGetElement(IList, Int32, ISymbol.)</td>
<td>Tries to get the array element with the specified indices (jagged array support).</td>
</tr>
<tr>
<td>TryGetElement(IList, Int32, ISymbol.)</td>
<td>Tries to get the array element</td>
</tr>
<tr>
<td>TryGetIndex</td>
<td>Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TrySetIndex</td>
<td>Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.</td>
</tr>
<tr>
<td>TryWriteValue</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>UpdateAnyValue</td>
<td>Reads the value of this Value [2254] into the specified managed value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteAnyValue</td>
<td>Writes the value represented by the managed value to this Value [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte)</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte, Int32)</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValueAsyc</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### TwinCAT.Ads DynamicArrayInstance Properties

The `DynamicArrayInstance` type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>_InnerSymbol</code></td>
<td>Gets the inner symbol of this DynamicSymbol (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td><code>AccessRights</code></td>
<td>Gets the access rights. (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td><code>AllowGIOAccess</code></td>
<td>Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol) (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td><code>Attributes</code></td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td><code>BitSize</code></td>
<td>Gets the size of the <code>IDataType</code> in bits. (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td><code>ByteSize</code></td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td><code>Category</code></td>
<td>Gets the category. (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td><code>Comment</code></td>
<td>Gets the comment of the <code>Instance</code> (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td><code>Connection</code></td>
<td>Gets the connection bound to this DynamicSymbol.</td>
</tr>
</tbody>
</table>

#### Reference

_TwinCAT.TypeSystem Namespace_
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DataType [1802]</strong></td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>Dimensions [1704]</strong></td>
<td>Gets the dimensions as read only collection.</td>
</tr>
<tr>
<td><strong>Elements [1704]</strong></td>
<td>Gets the contained Array Elements as read only collection.</td>
</tr>
<tr>
<td><strong>ElementType [1705]</strong></td>
<td>Gets the type of the contained elements.</td>
</tr>
<tr>
<td><strong>HasValue [1803]</strong></td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>InstanceName [1803]</strong></td>
<td>Gets the name of the instance (without periods (.)) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>InstancePath [1804]</strong></td>
<td>Gets the relative / absolute access path to the instance (with periods (.) ) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>IsBitType [1805]</strong></td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>IsByteAligned [1805]</strong></td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>IsContainerType [1806]</strong></td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>IsPersistent [1806]</strong></td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>IsPointer [1807]</strong></td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>IsPrimitiveType [1807]</strong></td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>IsReadOnly [1808]</strong></td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>IsRecursive [1808]</strong></td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>IsReference [1809]</strong></td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>IsStatic [1809]</strong></td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>Item [1705]</strong></td>
<td>Gets the ISymbol [2176] with the specified indices.</td>
</tr>
<tr>
<td><strong>NormalizedName [1810]</strong></td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>NotificationSettings [1810]</strong></td>
<td>Gets the notification settings. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>Parent [1811]</strong></td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>Size [1811]</strong></td>
<td>Gets the size of the IInstance [2052] in bytes. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>SubSymbols [1812]</strong></td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>TypeName [1812]</strong></td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>
Name | Description
--- | ---
ValueEncoding | Gets the value encoding. (Inherited from DynamicSymbol.)

Reference

DynamicArrayInstance Class [1698]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.13.1.1 DynamicArrayInstance.Dimensions Property

Gets the dimensions as read only collection.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public IDimensionCollection Dimensions { get; }
```

**Property Value**

**Type:** IDimensionCollection [2000]

The dimensions.

**Implements**

IArrayInstance.Dimensions [1967]

Reference

DynamicArrayInstance Class [1698]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.13.1.2 DynamicArrayInstance.Elements Property

Gets the contained Array Elements as read only collection.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public ISymbolCollection<ISymbol> Elements { get; }
```

**Property Value**

**Type:** ISymbolCollection [2185] ISymbol [2176].

The elements.
Implements

IArrayInstance.Elements [1968]

Reference

DynamicArrayInstance Class [1698]

TwinCAT.TypeSystem Namespace [1622]

6.11.13.1.3 DynamicArrayInstance.ElementType Property

Gets the type of the contained elements.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public IDataType ElementType { get; }

Property Value

Type: IDataType [1986]
The type of the element.

Implements

IArrayInstance.ElementType [1968]

Reference

DynamicArrayInstance Class [1698]

TwinCAT.TypeSystem Namespace [1622]

6.11.13.1.4 DynamicArrayInstance.Item Property

Gets the ISymbol with the specified indices.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public ISymbol this[int[] indices] { get; }
TwinCAT.Ads Namespaces

Parameters

indices

Type: System.Int32
The indices.

Return Value

Type: ISymbol [2176]
ISymbol.

Implements

IArrayInstance.Item.Int32.. [1969]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td>indices</td>
</tr>
</tbody>
</table>

Reference

DynamicArrayInstance Class [1698]
TwinCAT.TypeSystem Namespace [1622]

6.11.13.2 DynamicArrayInstance Methods

The DynamicArrayInstance [1698] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1816]</td>
<td>Equals (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>GetDynamicMemberNames [1816]</td>
<td>Returns the enumeration of all dynamic member names. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>GetHashCode [1817]</td>
<td>Gets the HashCode of the Address (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnReadAnyValue [1817]</td>
<td>Handler function for reading ADS 'Any' Values. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnReadRawValue [1818]</td>
<td>Handler function for reading Raw symbol value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnReadRawValueAsync [1818]</td>
<td>Handler function reading the raw value of the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnReadValue [1819]</td>
<td>Handler function for the (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnReadValueAsync [1820]</td>
<td>Handler function reading the DynamicSymbols [1791] value asynchronously. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnSetInstanceName [1820]</td>
<td>Sets a new InstanceName InstancePath (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>OnTryReadValue</strong></td>
<td>Handler function for the (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td><strong>OnTryWriteValue</strong></td>
<td>Handler function for reading symbols raw value. (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td><strong>OnWriteRawValue</strong></td>
<td>Handler function for writing the raw DynamicSymbol[1791] value. (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td><strong>OnWriteRawValueAsync</strong></td>
<td>Handler function for reading the raw DynamicSymbol[1791] value. (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td><strong>OnWriteValue</strong></td>
<td>Handler function for writing value. (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td><strong>OnWriteValueAsync</strong></td>
<td>Handler function for writing value. (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td><strong>ReadAnyValue</strong></td>
<td>Reads the value of this Value[2254] into a new created instance of the managed type (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td><strong>ReadRawValue</strong></td>
<td>Reads the raw value of the IValueSymbol[2254] (Ads Read / Write) (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td><strong>ReadRawValue(Int32)</strong></td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td><strong>ReadRawValueAsync</strong></td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td><strong>ReadValue(Int32)</strong></td>
<td>Reads the value of this DynamicSymbol[1791]. (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td><strong>ReadValueAsync</strong></td>
<td>Reads the Value of the IValueSymbol[2254] asynchronously. (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td><strong>TryGetElement(IList..Int32.., ISymbol.)</strong></td>
<td>Tries to get the array element with the specified indices (jagged array support).</td>
</tr>
<tr>
<td><strong>TryGetElement(Int32., ISymbol.)</strong></td>
<td>Tries to get the array element</td>
</tr>
<tr>
<td><strong>TryGetIndex</strong></td>
<td>Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.</td>
</tr>
<tr>
<td><strong>TryGetMember</strong></td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td><strong>TryReadValue</strong></td>
<td>Reads the Value of the IValueSymbol[2254] (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TrySetIndex [1711]</td>
<td>Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.</td>
</tr>
<tr>
<td>TryWriteValue [1834]</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>UpdateAnyValue [1837]</td>
<td>Reads the value of this Value [2254] into the specified managed value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteAnyValue [1838]</td>
<td>Writes the value represented by the managed value to this Value [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue( byte ) [1839]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue( byte, Int32 ) [1839]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValueAsync [1840]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValue(Object) [1841]</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValue(Object, Int32) [1844]</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValueAsync [1847]</td>
<td>Writes the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

Reference

DynamicArrayInstance Class [1698]

TwinCAT.TypeSystem Namespace [1622]

6.11.13.2.1 DynamicArrayInstance.TryGetElement Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryGetElement([List, Int32, ISymbol]) [1709]</td>
<td>Tries to get the array element with the specified indices (jagged array support).</td>
</tr>
<tr>
<td>TryGetElement([Int32, ISymbol]) [1709]</td>
<td>Tries to get the array element</td>
</tr>
</tbody>
</table>

Reference

DynamicArrayInstance Class [1698]

TwinCAT.TypeSystem Namespace [1622]
DynamicArrayInstance.TryGetElement Method (IList..Int32.., ISymbol.)

Tries to get the array element with the specified indices (jagged array support).

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public bool TryGetElement(
    IList<int[]> jaggedIndices,
    out ISymbol symbol
)
```

### Parameters

- **jaggedIndices**  
  - Type: `System.Collections.Generic.IList<int[]>`  
  - The jagged indices list.

- **symbol**  
  - Type: `TwinCAT.TypeSystem.ISymbol`  
  - The symbol.

### Return Value

- **Type:** `Boolean`  
  - `true` if found, `false` if the jagged indices specifiers is out-of-range.

### Implements

- `IArrayInstance.TryGetElement(IList..Int32.., ISymbol.)`

### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>jaggedIndices</td>
</tr>
<tr>
<td>ArgumentOutOfRangeException</td>
<td>jaggedIndices</td>
</tr>
</tbody>
</table>

### Reference

- DynamicArrayInstance Class [1698]
- TryGetElement Overload [1708]
- TwinCAT.TypeSystem Namespace [1622]

---

**DynamicArrayInstance.TryGetElement Method (.Int32., ISymbol.)**

Tries to get the array element

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public bool TryGetElement(
    int[] indices,
    out ISymbol symbol
)
```

Parameters

- `symbol`: Type: TwinCAT.TypeSystem.ISymbol. The found Array element symbol (out-parameter).

Return Value

Type: Boolean
true if found, false if the indices specifiers is out-of-range.

Implements

IArrayInstance.TryGetElement(Int32., ISymbol.)

Reference

DynamicArrayInstance Class
TryGetElement Overload
TwinCAT.TypeSystem Namespace

6.11.13.2.2 DynamicArrayInstance.TryGetIndex Method

Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public virtual bool TryGetIndex(
    GetIndexBinder binder,
    Object[] indexes,
    out Object result
)
```

Parameters

- `indexes`: Type: System.Object. The indexes that are used in the operation. For example, for the sampleObject[3] operation in C# (sampleObject(3) in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[0] is equal to 3.
TwinCAT.Ads Namespaces

result

Type: System.Object
The result of the index operation.

Return Value

Type: Boolean
ture if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicArrayInstance Class [1698]
TwinCAT.TypeSystem Namespace [1622]

6.11.13.2.3 DynamicArrayInstance.TrySetIndex Method

Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public virtual bool TrySetIndex(
    SetIndexBinder binder,
    Object[] indexes,
    Object value)

Parameters

binder
Type: System.Dynamic.SetIndexBinder
Provides information about the operation.

indexes
Type: System.Object
The indexes that are used in the operation. For example, for the sampleObject[3] = 10 operation in C# (sampleObject(3) = 10 in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[[ ]] is equal to 3.

value
Type: System.Object
The value to set to the object that has the specified index. For example, for the sampleObject[3] = 10 operation in C# (sampleObject(3) = 10 in Visual Basic), where sampleObject is derived from the DynamicObject class, value is equal to 10.

Return Value

Type: Boolean
ture if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicArrayInstance Class [1698]
TwinCAT.TypeSystem Namespace [1622]
6.11.13.3 DynamicArrayInstance Events

The DynamicArrayInstance type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol has changed. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol has changed. (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>

Reference

DynamicArrayInstance Class

TwinCAT.TypeSystem Namespace

6.11.13.4 DynamicArrayInstance Fields

The DynamicArrayInstance type exposes the following members.

Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>syncObject</td>
<td>Synchronization object (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>

Reference

DynamicArrayInstance Class

TwinCAT.TypeSystem Namespace

6.11.14 DynamicOversamplingArrayInstance Class

Dynamic Array Instance

Inheritance Hierarchy

System.Dynamic.DynamicObject
   TwinCAT.TypeSystem.DynamicSymbol
      TwinCAT.TypeSystem.DynamicArrayInstance
         TwinCAT.TypeSystem.DynamicOversamplingArrayInstance

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public sealed class DynamicOversamplingArrayInstance : DynamicArrayInstance, IOversamplingArrayInstance, IArrayInstance, ISymbol, IAttributedInstance, IInstance, IBitSize
```

The DynamicOversamplingArrayInstance type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerSymbol</td>
<td>Gets the inner symbol of this DynamicSymbol [1791] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>AccessRights</td>
<td>Gets the access rights. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the category. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the Instance [2052] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the connection bound to this DynamicSymbol [1791] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType [1986] of the Instance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Dimensions</td>
<td>Gets the dimensions as read only collection. (Inherited from DynamicArrayInstance [1698].)</td>
</tr>
<tr>
<td>Elements</td>
<td>Gets the contained Array Elements as read only collection. (Inherited from DynamicArrayInstance [1698].)</td>
</tr>
<tr>
<td>ElementType</td>
<td>Gets the type of the contained elements. (Inherited from DynamicArrayInstance [1698].)</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the Instance [2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

### Footnotes

- **[1791]** TwinCAT.Ads Namespaces
- **[1792]** Version: 1.1
- **[1793]** TC1000
- **[1794]** 1713
### Names

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsReference [1809]</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsStatic [1809]</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Item [1705]</td>
<td>Gets the ISymbol [2176] with the specified indices. (Inherited from DynamicArrayInstance [1698].)</td>
</tr>
<tr>
<td>NormalizedName [1810]</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>NotificationSettings [1810]</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OversamplingElement [1717]</td>
<td>Gets the oversampling element.</td>
</tr>
<tr>
<td>Parent [1811]</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Size [1811]</td>
<td>Gets the size of the IInstance [2052] in bytes. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>SubSymbols [1812]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TypeName [1812]</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueEncoding [1813]</td>
<td>Gets the value encoding. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1816]</td>
<td>Equals (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>GetDynamicMemberNames [1816]</td>
<td>Returns the enumeration of all dynamic member names. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>GetHashCode [1717]</td>
<td>Gets the HashCode of the Address (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadAnyValue [1825]</td>
<td>Reads the value of this Value [2254] into a new created instance of the managed type (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue [1826]</td>
<td>Reads the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue(Int32) [1826]</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValueAsync [1827]</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValue [1828]</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValue(Int32) [1831]</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValueAsync [1832]</td>
<td>Reads the Value of the IValueSymbol [2254] asynchronously. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TryGetElement(IList, Int32, ISymbol)</td>
<td>Tries to get the array element with the specified indices (jagged array support). (Inherited from DynamicArrayInstance [1698].)</td>
</tr>
<tr>
<td>TryGetElement(Int32, ISymbol)</td>
<td>Tries to get the array element (Inherited from DynamicArrayInstance [1698].)</td>
</tr>
<tr>
<td>TryGetIndex</td>
<td>Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations. (Inherited from DynamicArrayInstance [1698].)</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TrySetIndex</td>
<td>Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index. (Inherited from DynamicArrayInstance [1698].)</td>
</tr>
<tr>
<td>TryWriteValue</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>UpdateAnyValue</td>
<td>Reads the value of this Value [2254] into the specified managed value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteAnyValue</td>
<td>Writes the value represented by the managed value to this Value [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte)</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte, Int32)</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValueAsync</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValue/Object</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValue(Object, Int32)</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValueAsync</td>
<td>Writes the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>
TwinCAT.Ads Namespaces

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol has changed. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol has changed. (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace

6.11.14.1 DynamicOversamplingArrayInstance Properties

The DynamicOversamplingArrayInstance type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerSymbol</td>
<td>Gets the inner symbol of this DynamicSymbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>AccessRights</td>
<td>Gets the access rights. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the category. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the Instance (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the connection bound to this DynamicSymbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType of the Instance (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Dimensions</td>
<td>Gets the dimensions as read only collection. (Inherited from DynamicArrayInstance.)</td>
</tr>
<tr>
<td>Elements</td>
<td>Gets the contained Array Elements as read only collection. (Inherited from DynamicArrayInstance.)</td>
</tr>
<tr>
<td>ElementType</td>
<td>Gets the type of the contained elements. (Inherited from DynamicArrayInstance.)</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this instance has value (Inherited from DynamicArrayInstance.)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsByteAligned [1805]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsContainerType [1806]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPersistent [1806]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPointer [1807]</td>
<td>Indicates that the Instance [2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPrimitiveType [1807]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReadOnly [1808]</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsRecursive [1808]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReference [1809]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsStatic [1809]</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Item [1705]</td>
<td>Gets the ISymbol [2176] with the specified indices. (Inherited from DynamicArrayInstance [1698].)</td>
</tr>
<tr>
<td>NormalizedName [1810]</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>NotificationSettings [1810]</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OversamplingElement [1717]</td>
<td>Gets the oversampling element.</td>
</tr>
<tr>
<td>Parent [1811]</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Size [1811]</td>
<td>Gets the size of the Instance [2052] in bytes. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>SubSymbols [1812]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TypeName [1812]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueEncoding [1813]</td>
<td>Gets the value encoding. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

Reference

DynamicOversamplingArrayInstance Class [1712]
TwinCAT.TypeSystem Namespace [1622]

6.11.14.1.1 DynamicOversamplingArrayInstance.OversamplingElement Property

Gets the oversampling element.
TwinCAT.Ads Namespaces

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public ISymbol OversamplingElement { get; }
```

Property Value

Type: ISymbol [2176]
The oversampling element.

Implements

I OversamplingArrayInstance.OversamplingElement [2082]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

DynamicOversamplingArrayInstance Class [1712]
TwinCAT.TypeSystem Namespace [1622]

6.11.14.2 DynamicOversamplingArrayInstance Methods

The DynamicOversamplingArrayInstance [1712] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1816]</td>
<td>Equals (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>GetDynamicMemberNames [1816]</td>
<td>Returns the enumeration of all dynamic member names. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>GetHashCode [1817]</td>
<td>Gets the HashCode of the Address (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadAnyValue [1825]</td>
<td>Reads the value of this Value [2254] into a new created instance of the managed type (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue [1826]</td>
<td>Reads the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue(Int32) [1826]</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadRawValueAsyn</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Reads the value of this DynamicSymbol. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the value of this DynamicSymbol. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the IValueSymbol asynchronously. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>TryGetElement([List, Int32, , ISymbol])</td>
<td>Tries to get the array element with the specified indices (jagged array support). (Inherited from DynamicArrayInstance)</td>
</tr>
<tr>
<td>TryGetIndex</td>
<td>Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations. (Inherited from DynamicArrayInstance)</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>TrySetIndex</td>
<td>Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index. (Inherited from DynamicArrayInstance)</td>
</tr>
<tr>
<td>TryWriteValue</td>
<td>Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>UpdateAnyValue</td>
<td>Reads the value of this Value into the specified managed value. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>WriteAnyValue</td>
<td>Writes the value represented by the managed value to this Value. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>WriteRawValue(Byte)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>WriteRawValue(Byte, Int32)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>WriteRawValueAsyn</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol)</td>
</tr>
</tbody>
</table>
**TwinCAT.Ads Namespaces**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValue(Object)</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValue(Object, Int32)</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValueAsync</td>
<td>Writes the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

**Reference**

DynamicOversamplingArrayInstance Class [1712]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.14.3 DynamicOversamplingArrayInstance Events

The DynamicOversamplingArrayInstance [1712] type exposes the following members.

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol [2254] has changed. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol [2254] has changed. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

**Reference**

DynamicOversamplingArrayInstance Class [1712]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.15 DynamicPointerInstance Class

Dynamic Pointer Instance

**Inheritance Hierarchy**

System.Dynamic.DynamicObject  
TwinCAT.TypeSystem.DynamicSymbol [1791]  
TwinCAT.TypeSystem.DynamicPointerInstance

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```c#
public sealed class DynamicPointerInstance : DynamicSymbol,  
  IPointerInstance, ISymbol, IAttributedInstance, IInstance, IBitSize
```

The DynamicPointerInstance type exposes the following members.
### TwinCAT.Ads Namespaces

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerSymbol</td>
<td>Gets the inner symbol of this DynamicSymbol (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>AccessRights</td>
<td>Gets the access rights. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the category. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the Instance (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the connection bound to this DynamicSymbol (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType of the Instance. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this ISymbol is Persistent. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the Instance represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the Instance represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>NormalizedName</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotificationSettings</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Reference</td>
<td>Gets the resolved reference of Pointer / Reference (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the IInstance [2052] in bytes. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>SubSymbols</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Equals (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>GetDynamicMemberNames</td>
<td>Returns the enumeration of all dynamic member names. (Overides DynamicSymbol.GetDynamicMemberNames [1816].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadAnyValue</td>
<td>Reads the value of this Value [2254] into a new created instance of the managed type (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue</td>
<td>Reads the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValueAsync</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the IValueSymbol [2254] asynchronously. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overides DynamicSymbol.TryGetMember(GetMemberBinder, Object.) [1833].)</td>
</tr>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### TwinCAT.TypeSystem Namespace

**6.11.15.1 DynamicPointerInstance Properties**

The `DynamicPointerInstance` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>InnerSymbol</code></td>
<td>Gets the inner symbol of this <code>DynamicSymbol</code> (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td><code>AccessRights</code></td>
<td>Gets the access rights. (Inherited from <code>DynamicSymbol</code>.</td>
</tr>
<tr>
<td><code>Attributes</code></td>
<td>Gets the Symbol Attributes (Inherited from <code>DynamicSymbol</code>.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BitSize [1800]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ByteSize [1800]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Category [1801]</td>
<td>Gets the category. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Comment [1801]</td>
<td>Gets the comment of the Instance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Connection [1802]</td>
<td>Gets the connection bound to this DynamicSymbol [1791] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>DataType [1802]</td>
<td>Gets the IDataType [1986] of the Instance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>HasValue [1803]</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstanceName [1803]</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstancePath [1804]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsBitType [1805]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsByteAligned [1805]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsContainerType [1806]</td>
<td>Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPersistent [1806]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPointer [1807]</td>
<td>Indicates that the Instance [2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPrimitiveType [1807]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReadOnly [1808]</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsRecursive [1808]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReference [1808]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsStatic [1809]</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>NormalizedName [1810]</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>NotificationSettings [1810]</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Parent [1811]</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Reference [1825]</td>
<td>Gets the resolved reference of Pointer / Reference</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### Reference

**DynamicPointerInstance Class [1720]**

**TwinCAT.TypeSystem Namespace [1622]**

### 6.11.15.1.1 DynamicPointerInstance.Reference Property

Gets the resolved reference of Pointer / Reference

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public ISymbol Reference { get; }
```

**Property Value**

Type: ISymbol [2176]  
The reference symbol or NULL if PVOID Pointer.

**Implements**

IPointerInstance.Reference [2086]

### Reference

**DynamicPointerInstance Class [1720]**

**TwinCAT.TypeSystem Namespace [1622]**

### 6.11.15.2 DynamicPointerInstance Methods

The DynamicPointerInstance [1720] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1816]</td>
<td>Equals (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>GetDynamicMemberNames</td>
<td>Returns the enumeration of all dynamic member names. (Overrides DynamicSymbol.GetDynamicMemberNames. [1816].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadAnyValue</td>
<td>Reads the value of this Value [2254] into a new created instance of the managed type (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue</td>
<td>Reads the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValueAsync</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the IValueSymbol [2254] asynchronously. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrids DynamicSymbol.TryGetMember(GetMemberBinder, Object.) [1833].)</td>
</tr>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TryWriteValue</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>UpdateAnyValue</td>
<td>Reads the value of this Value [2254] into the specified managed value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteAnyValue</td>
<td>Writes the value represented by the managed value to this Value [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte)</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte, Int32)</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValueAsync</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValue(Object)</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteValue(Object, Int32)</td>
<td>Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteValueAsync</td>
<td>Writes the Value of the IValueSymbol. (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>

**Reference**

DynamicPointerInstance Class [1720]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.15.2.1 DynamicPointerInstance.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public override IEnumerable<string> GetDynamicMemberNames()
```

**Return Value**

Type: IEnumerable<String>

A sequence that contains dynamic member names.

**Reference**

DynamicPointerInstance Class [1720]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.15.2.2 DynamicPointerInstance.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public override bool TryGetMember(GetMemberBinder binder, out Object result)
```
Parameters

binder
Type: System.Dynamic.GetMemberBinder
Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the Console.WriteLine(sampleObject.SampleProperty) statement, where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.

result
Type: System.Object
The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicPointerInstance Class [1720]
TwinCAT.TypeSystem Namespace [1622]

6.11.15.3 DynamicPointerInstance Events

The DynamicPointerInstance [1720] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol [2254] has changed. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol [2254] has changed. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

Reference

DynamicPointerInstance Class [1720]
TwinCAT.TypeSystem Namespace [1622]

6.11.16 DynamicPointerValue Class

Class DynamicPointerValue.

Inheritance Hierarchy

System.Dynamic.DynamicObject
   TwinCAT.TypeSystem.DynamicValue [1869]
   TwinCAT.TypeSystem.DynamicPointerValue

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ddca3e72bc0ea15da1c14
The DynamicPointerValue type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Gets the age of the value (last successful read of the value) (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>CachedRaw</td>
<td>Gets the cached Raw internal Data. (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the data type bound to this Value (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this Value is a primitive value. (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>ResolvedType</td>
<td>Gets the resolved type. (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol that is bound to this value. (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Gets the Time stamp of the last successful read of the Value. (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>UpdateMode</td>
<td>Gets / Sets the update mode (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>ValueFactory</td>
<td>The value factory (Inherited from DynamicValue.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDynamicMemberNames</td>
<td>Returns the enumeration of all dynamic member names. (Overrides DynamicValue.GetDynamicMemberNames.)</td>
</tr>
<tr>
<td>Read</td>
<td>Reads the value (via ADS) (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>ReadAsync</td>
<td>read as an asynchronous operation. (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>ReadMember</td>
<td>Reads the specified member element. (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>ResolveValue</td>
<td>Resolves the Value object to its primitive value. (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>TryConvert</td>
<td>Provides implementation for type conversion operations. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that convert an object from one type to another. (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>TryGetArrayElementValues</td>
<td>Returns Array Element values. (Inherited from DynamicValue.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryGetIndex <img src="1883" alt="1883" /></td>
<td>Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations. (Inherited from DynamicValue <img src="1869" alt="1869" />.)</td>
</tr>
<tr>
<td>TryGetIndexValue <img src="1884" alt="1884" /></td>
<td>Reads the specified array element. (Inherited from DynamicValue <img src="1869" alt="1869" />.)</td>
</tr>
<tr>
<td>TryGetMember <img src="1886" alt="1886" /></td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicValue <img src="1869" alt="1869" />.)</td>
</tr>
<tr>
<td>TryGetMemberValue <img src="1733" alt="1733" /></td>
<td>Tries the get member value. (Overides DynamicValue.TryGetMemberValue(String, Object.) <img src="1887" alt="1887" />.)</td>
</tr>
<tr>
<td>TryInvoke <img src="1887" alt="1887" /></td>
<td>Provides the implementation for operations that invoke an object. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as invoking an object or a delegate. (Inherited from DynamicValue <img src="1869" alt="1869" />.)</td>
</tr>
<tr>
<td>TryInvokeMember <img src="1888" alt="1888" /></td>
<td>Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method. (Inherited from DynamicValue <img src="1869" alt="1869" />.)</td>
</tr>
<tr>
<td>TryResolveValue <img src="1889" alt="1889" /></td>
<td>Tries to resolves the Value object to its primitive value. (Inherited from DynamicValue <img src="1869" alt="1869" />.)</td>
</tr>
<tr>
<td>TrySetIndex <img src="1890" alt="1890" /></td>
<td>Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index. (Inherited from DynamicValue <img src="1869" alt="1869" />.)</td>
</tr>
<tr>
<td>TrySetIndexValue <img src="1890" alt="1890" /></td>
<td>Tries to set the indexed value on Arrays (Inherited from DynamicValue <img src="1869" alt="1869" />.)</td>
</tr>
<tr>
<td>TrySetMember <img src="1891" alt="1891" /></td>
<td>Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Inherited from DynamicValue <img src="1869" alt="1869" />.)</td>
</tr>
<tr>
<td>TrySetMemberValue <img src="1892" alt="1892" /></td>
<td>Tries to Set a Member/Property Value (Inherited from DynamicValue <img src="1869" alt="1869" />.)</td>
</tr>
<tr>
<td>Write <img src="1893" alt="1893" /></td>
<td>Writes the value (via ADS) (Inherited from DynamicValue <img src="1869" alt="1869" />.)</td>
</tr>
<tr>
<td>WriteAsync <img src="1893" alt="1893" /></td>
<td>write as an asynchronous operation. (Inherited from DynamicValue <img src="1869" alt="1869" />.)</td>
</tr>
<tr>
<td>WriteMember <img src="1894" alt="1894" /></td>
<td>Writes the specified member element. (Inherited from DynamicValue <img src="1869" alt="1869" />.)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace ![1622](1622)

TwinCAT.TypeSystem.DynamicValue ![1869](1869)
### 6.11.16.1 DynamicPointerValue Properties

The `DynamicPointerValue` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Gets the age of the value (last successful read of the value) (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>CachedRaw</td>
<td>Gets the cached Raw internal Data. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the data type bound to this <code>IValue</code> (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this <code>IValue</code> is a primitive value. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>ResolvedType</td>
<td>Gets the resolved type. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol that is bound to this value. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Gets the Time stamp of the last successful read of the Value. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>UpdateMode</td>
<td>Gets / Sets the update mode (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>ValueFactory</td>
<td>The value factory (Inherited from DynamicValue)</td>
</tr>
</tbody>
</table>

**Reference**

- DynamicPointerValue Class
- TwinCAT.TypeSystem Namespace

### 6.11.16.2 DynamicPointerValue Methods

The `DynamicPointerValue` type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDynamicMemberNames</td>
<td>Returns the enumeration of all dynamic member names. (Overrides DynamicValue.GetDynamicMemberNames)</td>
</tr>
<tr>
<td>Read</td>
<td>Reads the value (via ADS) (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>ReadAsync</td>
<td>read as an asynchronous operation. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>ReadMember</td>
<td>Reads the specified member element. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>ResolveValue</td>
<td>Resolves the Value object to its primitive value. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents this instance. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryConvert (<a href="#">1732</a>)</td>
<td>Provides implementation for type conversion operations. Classes derived from the <code>DynamicObject</code> class can override this method to specify dynamic behavior for operations that convert an object from one type to another. (Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
<tr>
<td>TryGetArrayElement Values (<a href="#">1883</a>)</td>
<td>Returns Array Element values. (Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
<tr>
<td>TryGetIndex (<a href="#">1883</a>)</td>
<td>Provides the implementation for operations that get a value by index. Classes derived from the <code>DynamicObject</code> class can override this method to specify dynamic behavior for indexing operations. (Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
<tr>
<td>TryGetIndexValue(Int32, Object) (<a href="#">1884</a>)</td>
<td>Reads the specified array element. (Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
<tr>
<td>TryGetIndexValue(Object, Object) (<a href="#">1885</a>)</td>
<td>(Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
<tr>
<td>TryGetMember (<a href="#">1886</a>)</td>
<td>Provides the implementation for operations that get member values. Classes derived from the <code>DynamicObject</code> class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
<tr>
<td>TryGetMemberValue(<a href="#">1733</a>)</td>
<td>Tries the get member value. (Overides <code>DynamicValue.TryGetMemberValue(String, Object) ([1887](#))</code>.</td>
</tr>
<tr>
<td>TryInvoke (<a href="#">1887</a>)</td>
<td>Provides the implementation for operations that invoke an object. Classes derived from the <code>DynamicObject</code> class can override this method to specify dynamic behavior for operations such as invoking an object or a delegate. (Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
<tr>
<td>TryInvokeMember (<a href="#">1888</a>)</td>
<td>Provides the implementation for operations that invoke a member. Classes derived from the <code>DynamicObject</code> class can override this method to specify dynamic behavior for operations such as calling a method. (Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
<tr>
<td>TryResolveValue (<a href="#">1889</a>)</td>
<td>Tries to resolves the Value object to its primitive value. (Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
<tr>
<td>TrySetIndex (<a href="#">1890</a>)</td>
<td>Provides the implementation for operations that set a value by index. Classes derived from the <code>DynamicObject</code> class can override this method to specify dynamic behavior for operations that access objects by a specified index. (Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
<tr>
<td>TrySetIndexValue(<a href="#">1890</a>)</td>
<td>Tries to set the indexed value on Arrays (Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
<tr>
<td>TrySetMember (<a href="#">1891</a>)</td>
<td>Provides the implementation for operations that set member values. Classes derived from the <code>DynamicObject</code> class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
<tr>
<td>TrySetMemberValue(<a href="#">1892</a>)</td>
<td>Tries to Set a Member/Property Value (Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
<tr>
<td>Write (<a href="#">1893</a>)</td>
<td>Writes the value (via ADS) (Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
<tr>
<td>WriteAsync (<a href="#">1893</a>)</td>
<td>write as an asynchronous operation. (Inherited from <code>DynamicValue ([1869](#))</code>.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteMember</td>
<td>Writes the specified member element. (Inherited from DynamicValue)</td>
</tr>
</tbody>
</table>

**Reference**

DynamicPointerValue Class [1728]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.16.2.1 DynamicPointerValue.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public override IEnumerable<string> GetDynamicMemberNames()
```

**Return Value**

Type: `IEnumerable<String>`

A sequence that contains dynamic member names.

**Reference**

DynamicPointerValue Class [1728]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.16.2.2 DynamicPointerValue.TryGetMemberValue Method

Tries the get member value.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public override bool TryGetMemberValue(string name, out object result)
```

**Parameters**

- `name`  
  Type: `System.String`  
  The name.

- `result`  
  Type: `System.Object`  
  The result.
Return Value
Type: Boolean
true if XXXX, false otherwise.

Implements
[IStructValue.TryGetMemberValue(String, Object)]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolException</td>
<td></td>
</tr>
</tbody>
</table>

Reference
DynamicPointerValue Class
TwinCAT.TypeSystem Namespace

6.11.17 DynamicReferenceInstance Class
Dynamic Reference Instance

Inheritance Hierarchy
System.Dynamic.DynamicObject
   TwinCAT.TypeSystem.DynamicSymbol
   TwinCAT.TypeSystem.DynamicReferenceInstance

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class DynamicReferenceInstance : DynamicSymbol, IReferenceInstance, ISymbol, IAttributedInstance, IInstance, IBitSize

The DynamicReferenceInstance type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerSymbol</td>
<td>Gets the inner symbol of this DynamicSymbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>AccessRights</td>
<td>Gets the access rights. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>AllowIGIOAccess</td>
<td>Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ByteSize [¶ 1800]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>Category [¶ 1801]</td>
<td>Gets the category. (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>Comment [¶ 1801]</td>
<td>Gets the comment of the [Instance [¶ 2052] (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>Connection [¶ 1802]</td>
<td>Gets the connection bound to this DynamicSymbol [¶ 1791] (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>DataType [¶ 1802]</td>
<td>Gets the IDirectType [¶ 1986] of the IInstance [¶ 2052]. (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>HasValue [¶ 1803]</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>InstanceName [¶ 1803]</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>InstancePath [¶ 1804]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>IsBitType [¶ 1805]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>IsByteAligned [¶ 1805]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>IsContainerType [¶ 1806]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>IsPersistent [¶ 1806]</td>
<td>Gets a value indicating whether this I[Symbol [¶ 2176] is persistent. (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>IsPointer [¶ 1807]</td>
<td>Indicates that the IInstance [¶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>IsPrimitiveType [¶ 1807]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>IsReadOnly [¶ 1808]</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>IsRecursive [¶ 1808]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>IsReference [¶ 1809]</td>
<td>Indicates that the IInstance [¶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>IsStatic [¶ 1809]</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>NormalizedName [¶ 1810]</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>NotificationSettings [¶ 1810]</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>Parent [¶ 1811]</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol [¶ 1791].)</td>
</tr>
<tr>
<td>ReferencedType [¶ 1740]</td>
<td>Gets the referenced type of the IReferenceInstance [¶ 2094]</td>
</tr>
<tr>
<td>ResolvedByteSize [¶ 1740]</td>
<td>Gets the resolved byte size of the IReferenceInstance [¶ 2094].</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### Name | Description
--- | ---
ResolvedCategory [1741] | Gets the Category of the Referenced Symbol.
ResolvedType [1741] | Gets the resolved type of the IReferenceInstance [2094].
Size [1811] | Gets the size of the IInstance [2052] in bytes. (Inherited from DynamicSymbol [1791].)
SubSymbols [1812] | Gets the SubSymbols of the ISymbol [2176] (Inherited from DynamicSymbol [1791].)
TypeName [1812] | Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from DynamicSymbol [1791].)
ValueEncoding [1813] | Gets the value encoding. (Inherited from DynamicSymbol [1791].)

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1816]</td>
<td>Equals (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>GetDynamicMemberNames [1816]</td>
<td>Returns the enumeration of all dynamic member names. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>GetHashCode [1817]</td>
<td>Gets the HashCode of the Address (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnReadAnyValue [1817]</td>
<td>Handler function for reading ADS 'Any' Values. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnReadRawValue [1818]</td>
<td>Handler function for reading Raw symbol value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnReadRawValueAsync [1818]</td>
<td>Handler function reading the raw value of the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnReadValue [1819]</td>
<td>Handler function for the (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnReadValueAsync [1820]</td>
<td>Handler function reading the DynamicSymbols [1791] value asynchronously. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnSetInstanceName [1820]</td>
<td>Sets a new InstanceName InstancePath (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnTryReadValue [1821]</td>
<td>Handler function for the (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnTryWriteValue [1821]</td>
<td>Handler Function for writing value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnWriteRawValue [1822]</td>
<td>Handler function for reading symbols raw value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnWriteRawValueAsync [1823]</td>
<td>Handler function for writing the raw DynamicSymbol [1791] value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnWriteValue [1823]</td>
<td>Handler Function for writing value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnWriteValueAsync [1824]</td>
<td>Handler Function for writing value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReadAnyValue</td>
<td>Reads the value of this Value into a new created instance of the managed type (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>ReadRawValue</td>
<td>Reads the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>ReadRawValueAsync</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Reads the value of this DynamicSymbol. (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the value of this DynamicSymbol. (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the IValueSymbol asynchronously. (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>TryGetIndex</td>
<td>Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>TrySetIndex</td>
<td>Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.</td>
</tr>
<tr>
<td>TryWriteValue</td>
<td>Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>UpdateAnyValue</td>
<td>Reads the value of this Value into the specified managed value. (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>WriteAnyValue</td>
<td>Writes the value represented by the managed value to this Value (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>WriteRawValue(Byte)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>WriteRawValue(Byte, Int32)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>WriteRawValueAsync</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol).</td>
</tr>
<tr>
<td>WriteValue(Object)</td>
<td>Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol).</td>
</tr>
</tbody>
</table>
TwinCAT.Ads Namespaces

### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValue(Object, Int32)</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValueAsync</td>
<td>Writes the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol [2254] has changed. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol [2254] has changed. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>normalizedDict</td>
<td>Dictionary of normalized Instance Names</td>
</tr>
<tr>
<td>resolvedReferenceType</td>
<td>The resolved alias type</td>
</tr>
<tr>
<td>syncObject</td>
<td>Synchronization object (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.17.1 DynamicReferenceInstance Properties

The DynamicReferenceInstance [1734] type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_InnerSymbol</td>
<td>Gets the inner symbol of this DynamicSymbol [1791] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>AccessRights</td>
<td>Gets the access rights. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>AllowIGIOAccess</td>
<td>Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the category. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the Instance [2052] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Connection [1802]</td>
<td>Gets the connection bound to this DynamicSymbol [1791] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>DataType [1802]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>HasValue [1803]</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstanceName [1803]</td>
<td>Gets the name of the instance (without periods .) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstancePath [1804]</td>
<td>Gets the relative / absolute access path to the instance (with periods .)) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsBitType [1805]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsByteAligned [1805]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsContainerType [1806]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPersistent [1806]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPointer [1807]</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPrimitiveType [1807]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReadOnly [1808]</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsRecursive [1808]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReference [1809]</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsStatic [1809]</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>NormalizedName [1810]</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>NotificationSettings [1810]</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Parent [1811]</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReferencedType [1740]</td>
<td>Gets the referenced type of the IReferenceInstance [2094]</td>
</tr>
<tr>
<td>ResolvedByteSize [1740]</td>
<td>Gets the resolved byte size of the IReferenceInstance [2094].</td>
</tr>
<tr>
<td>ResolvedCategory [1741]</td>
<td>Gets the Category of the Referenced Symbol.</td>
</tr>
<tr>
<td>ResolvedType [1741]</td>
<td>Gets the resolved type of the IReferenceInstance [2094].</td>
</tr>
<tr>
<td>Size [1811]</td>
<td>Gets the size of the IInstance [2052] in bytes. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>
### DynamicReferenceInstance.ReferencedType Property

Gets the referenced type of the IReferenceInstance.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public IDataType ReferencedType { get; }
```

**Property Value**

Type: IDataType

The referenced type

**Implements**

IReferenceInstance.ReferencedType

**Reference**

DynamicReferenceInstance Class

### DynamicReferenceInstance.ResolvedByteSize Property

Gets the resolved byte size of the IReferenceInstance.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public int ResolvedByteSize { get; }
```


**Property Value**

Type: `Int32`
The resolved byte size.

**Implements**

`IReferenceInstance.ResolvedByteSize` [2098]

**Reference**

`DynamicReferenceInstance Class` [1734]

`TwinCAT.TypeSystem Namespace` [1622]

### 6.11.17.1.3 DynamicReferenceInstance.ResolvedCategory Property

Gets the Category of the Referenced Symbol.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```
public DataTypeCategory ResolvedCategory { get; }
```

**Property Value**

Type: `DataTypeCategory` [1649]
The resolved category.

**Implements**

`IReferenceInstance.ResolvedCategory` [2098]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>NotImplementedException</code></td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

`DynamicReferenceInstance Class` [1734]

`TwinCAT.TypeSystem Namespace` [1622]

### 6.11.17.1.4 DynamicReferenceInstance.ResolvedType Property

Gets the resolved type of the `IReferenceInstance` [2094].
TwinCAT.Ads Namespaces

**Namespace:** TwinCAT.TypeSystem [► 1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

*C#*

```csharp
public IDataType ResolvedType { get; }
```

**Property Value**

Type: [IDataType ► 1986]
The resolved type.

**Implements**

IRelativeReferenceInstance.ResolvedType [► 2099]

**Reference**

DynamicReferenceInstance Class [► 1734]

TwinCAT.TypeSystem Namespace [► 1622]

### 6.11.17.2 DynamicReferenceInstance Methods

The `DynamicReferenceInstance [► 1734]` type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| ≔ Equals [► 1816] | Equals (Inherited from `DynamicSymbol [► 1791].`)
| ≔ GetDynamicMemberNames [► 1816] | Returns the enumeration of all dynamic member names. (Inherited from `DynamicSymbol [► 1791].`)
| ≔ OnReadAnyValue [► 1817] | Handler function for reading ADS 'Any' Values. (Inherited from `DynamicSymbol [► 1791].`)
| ≔ OnReadRawValue [► 1818] | Handler function for reading Raw symbol value. (Inherited from `DynamicSymbol [► 1791].`)
| ≔ OnReadValue [► 1819] | Handler function for the (Inherited from `DynamicSymbol [► 1791].`)
| ≔ OnSetInstanceName [► 1820] | Sets a new InstanceName InstancePath (Inherited from `DynamicSymbol [► 1791].`)
| ≔ OnTryReadValue [► 1821] | Handler function for the (Inherited from `DynamicSymbol [► 1791].`)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OnTryWriteValue</td>
<td>Handler Function for writing value. (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>OnWriteRawValue</td>
<td>Handler function for reading symbols raw value. (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>OnWriteRawValueAsync</td>
<td>Handler function for writing the raw DynamicSymbol [1791] value. (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>OnWriteValue</td>
<td>Handler Function for writing value. (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>OnWriteValueAsync</td>
<td>Handler Function for writing value. (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>ReadAnyValue</td>
<td>Reads the value of this Value [2254] into a new created instance of the managed type (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>ReadRawValue</td>
<td>Reads the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>ReadRawValueAsync</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the IValueSymbol [2254] asynchronously. (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>TryGetIndex</td>
<td>Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>TrySetIndex</td>
<td>Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.</td>
</tr>
<tr>
<td>TryWriteValue</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>UpdateAnyValue</td>
<td>Reads the value of this Value [2254] into the specified managed value. (Inherited from DynamicSymbol [1791])</td>
</tr>
<tr>
<td>WriteAnyValue</td>
<td>Writes the value represented by the managed value to this Value [2254] (Inherited from DynamicSymbol [1791])</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteRawValue(Byte)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteRawValue(Byte, Int32)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteRawValueAsync</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteValue(Object)</td>
<td>Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteValue(Object, Int32)</td>
<td>Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteValueAsync</td>
<td>Writes the Value of the IValueSymbol (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>

### Reference

DynamicReferenceInstance Class [1734]
TwinCAT.TypeSystem Namespace [1622]

#### 6.11.17.2.1 DynamicReferenceInstance.TryGetIndex Method

Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public virtual bool TryGetIndex(
    GetIndexBinder binder,
    Object[] indexes,
    out Object result
)
```

**Parameters**

- **binder**
  - Type: System.Dynamic.GetIndexBinder
  - Provides information about the operation.

- **indexes**
  - Type: System.Object
  - The indexes that are used in the operation. For example, for the sampleObject[3] operation in C# (sampleObject(3) in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[0] is equal to 3.

- **result**
  - Type: System.Object
  - The result of the index operation.
Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicReferenceInstance Class [1734]
TwinCAT.TypeSystem Namespace [1622]

6.11.17.2.2 DynamicReferenceInstance.TrySetIndex Method

Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd2ca3e72bc0ea15da1c14

Syntax

C#

public virtual bool TrySetIndex(
    SetIndexBinder binder,
    Object[] indexes,
    Object value
)

Parameters

binder
Type: System.Dynamic.SetIndexBinder
Provides information about the operation.

indexes
Type: System.Object
The indexes that are used in the operation. For example, for the sampleObject[3] = 10 operation in C# (sampleObject(3) = 10 in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[0] is equal to 3.

value
Type: System.Object
The value to set to the object that has the specified index. For example, for the sampleObject[3] = 10 operation in C# (sampleObject(3) = 10 in Visual Basic), where sampleObject is derived from the DynamicObject class, value is equal to 10.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicReferenceInstance Class [1734]
TwinCAT.TypeSystem Namespace [1622]
6.11.17.3 DynamicReferenceInstance Events

The `DynamicReferenceInstance` type exposes the following members.

**Events**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>RawValueChanged</code></td>
<td>Occurs when the RawValue of the <code>IValueSymbol</code> has changed. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>ValueChanged</code></td>
<td>Occurs when the (Primitive) value of the <code>IValueSymbol</code> has changed. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
</tbody>
</table>

**Reference**

`DynamicReferenceInstance Class`  
`TwinCAT.TypeSystem Namespace`

6.11.17.4 DynamicReferenceInstance Fields

The `DynamicReferenceInstance` type exposes the following members.

**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>normalizedDict</code></td>
<td>Dictionary of normalized Instance Names</td>
</tr>
<tr>
<td><code>resolvedReferenceType</code></td>
<td>The resolved alias type</td>
</tr>
<tr>
<td><code>syncObject</code></td>
<td>Synchronization object (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
</tbody>
</table>

**Reference**

`DynamicReferenceInstance Class`  
`TwinCAT.TypeSystem Namespace`

6.11.17.4.1 DynamicReferenceInstance.normalizedDict Field

Dictionary of normalized Instance Names

**Namespace:** `TwinCAT.TypeSystem`  
**Assembly:** `TwinCAT.Ads (in TwinCAT.Ads.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected Dictionary<string, ISymbol> normalizedDict
```

**Field Value**

Type: `Dictionary<String, ISymbol>`
6.11.17.4.2 DynamicReferenceInstance.resolvedReferenceType Field

The resolved alias type

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

protected IDataType resolvedReferenceType

Field Value

Type: IDataType

Reference

DynamicReferenceInstance Class [1734]
TwinCAT.TypeSystem Namespace [1622]

6.11.18 DynamicReferenceValue Class

Class DynamicReferenceValue.

Inheritance Hierarchy

System.Dynamic.DynamicObject
   TwinCAT.TypeSystem.DynamicValue [1869]
   TwinCAT.TypeSystem.DynamicReferenceValue

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class DynamicReferenceValue : DynamicValue

The DynamicReferenceValue type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age [1872]</td>
<td>Gets the age of the value (last successful read of the value) (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>CachedRaw</td>
<td>Gets the cached Raw internal Data. (Inherited from DynamicValue [1869].)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### Name | Description
--- | ---
DataType | Gets the data type bound to this IValue (Inherited from DynamicValue)
IsPrimitive | Gets a value indicating whether this IValue is a primitive value. (Inherited from DynamicValue)
ResolvedType | Gets the resolved type. (Inherited from DynamicValue)
Symbol | Gets the symbol that is bound to this value. (Inherited from DynamicValue)
TimeStamp | Gets the Time stamp of the last successful read of the Value. (Inherited from DynamicValue)
UpdateMode | Gets / Sets the update mode (Inherited from DynamicValue)
ValueFactory | The value factory (Inherited from DynamicValue)

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDynamicMemberNames</td>
<td>Returns the enumeration of all dynamic member names. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>Read</td>
<td>Reads the value (via ADS) (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>ReadAsync</td>
<td>read as an asynchronous operation. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>ReadMember</td>
<td>Reads the specified member element. (Overrides DynamicValue.ReadMember(ISymbol))</td>
</tr>
<tr>
<td>ResolveValue</td>
<td>Resolves the Value object to its primitive value. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>TryConvert</td>
<td>Provides implementation for type conversion operations. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that convert an object from one type to another. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>TryGetArrayElementValues</td>
<td>Returns Array Element values. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>TryGetIndex</td>
<td>Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>TryGetIndexValue( int32, Object )</td>
<td>Reads the specified array element. (Inherited from DynamicValue)</td>
</tr>
<tr>
<td>TryGetIndexValue( Object, Object )</td>
<td>Tries the get index value. (Inherited from DynamicValue)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryGetMember[1886]</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicValue[1869].)</td>
</tr>
<tr>
<td>TryGetMemberValue[1887]</td>
<td>Tries the get member value. (Inherited from DynamicValue[1869].)</td>
</tr>
<tr>
<td>TryInvoke[1887]</td>
<td>Provides the implementation for operations that invoke an object. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as invoking an object or a delegate. (Inherited from DynamicValue[1869].)</td>
</tr>
<tr>
<td>TryInvokeMember[1888]</td>
<td>Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method. (Inherited from DynamicValue[1869].)</td>
</tr>
<tr>
<td>TryResolveValue[1889]</td>
<td>Tries to resolves the Value object to its primitive value. (Inherited from DynamicValue[1869].)</td>
</tr>
<tr>
<td>TrySetIndex[1890]</td>
<td>Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index. (Inherited from DynamicValue[1869].)</td>
</tr>
<tr>
<td>TrySetIndexValue[1890]</td>
<td>Tries to set the indexed value on Arrays (Inherited from DynamicValue[1869].)</td>
</tr>
<tr>
<td>TrySetMember[1891]</td>
<td>Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Inherited from DynamicValue[1869].)</td>
</tr>
<tr>
<td>TrySetMemberValue[1892]</td>
<td>Tries to Set a Member/Property Value (Inherited from DynamicValue[1869].)</td>
</tr>
<tr>
<td>Write[1893]</td>
<td>Writes the value (via ADS) (Inherited from DynamicValue[1869].)</td>
</tr>
<tr>
<td>WriteAsync[1893]</td>
<td>write as an asynchronous operation. (Inherited from DynamicValue[1869].)</td>
</tr>
<tr>
<td>WriteMember[1894]</td>
<td>Writes the specified member element. (Inherited from DynamicValue[1869].)</td>
</tr>
</tbody>
</table>

#### Reference

TwinCAT.TypeSystem Namespace [1622]

TwinCAT.TypeSystem.DynamicValue [1869]

### 6.11.18.1 DynamicReferenceValue Properties

The DynamicReferenceValue[1747] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age[1872]</td>
<td>Gets the age of the value (last successful read of the value) (Inherited from DynamicValue[1869].)</td>
</tr>
<tr>
<td>CachedRaw[1873]</td>
<td>Gets the cached Raw internal Data. (Inherited from DynamicValue[1869].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DataType [1874]</td>
<td>Gets the data type bound to this IValue [2226] (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>IsPrimitive [1874]</td>
<td>Gets a value indicating whether this IValue [2226] is a primitive value. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>ResolvedType [1875]</td>
<td>Gets the resolved type. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>Symbol [1875]</td>
<td>Gets the symbol that is bound to this value. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>TimeStamp [1876]</td>
<td>Gets the Time stamp of the last successful read of the Value. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>UpdateMode [1876]</td>
<td>Gets / Sets the update mode (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>ValueFactory [1877]</td>
<td>The value factory (Inherited from DynamicValue [1869].)</td>
</tr>
</tbody>
</table>

Reference

DynamicReferenceValue Class [1747]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.18.2 DynamicReferenceValue Methods

The DynamicReferenceValue [1747] type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDynamicMemberNames [1879]</td>
<td>Returns the enumeration of all dynamic member names. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>Read [1879]</td>
<td>Reads the value (via ADS) (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>ReadAsync [1880]</td>
<td>read as an asynchronous operation. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>ReadMember [1751]</td>
<td>Reads the specified member element. (Overrdes DynamicValue.ReadMember(ISymbol) [1880].)</td>
</tr>
<tr>
<td>ResolveValue [1881]</td>
<td>Resolves the Value object to its primitive value. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>ToString [1882]</td>
<td>Returns a String that represents this instance. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>TryConvert [1882]</td>
<td>Provides implementation for type conversion operations. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that convert an object from one type to another. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>TryGetArrayElementValues [1883]</td>
<td>Returns Array Element values. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>TryGetIndex [1883]</td>
<td>Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryGetIndexValue(Int32, Object) [1884]</td>
<td>Reads the specified array element. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>TryGetIndexValue(Object, Object) [1885]</td>
<td>Tries the get index value. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>TryGetMember [1886]</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>TryGetMemberValue [1887]</td>
<td>Tries the get member value. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>TryInvoke [1887]</td>
<td>Provides the implementation for operations that invoke an object. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as invoking an object or a delegate. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>TryInvokeMember [1888]</td>
<td>Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>TryResolveValue [1889]</td>
<td>Tries to resolves the Value object to its primitive value. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>TrySetIndex [1890]</td>
<td>Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>TrySetIndexValue [1890]</td>
<td>Tries to set the indexed value on Arrays (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>TrySetMember [1891]</td>
<td>Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>TrySetMemberValue [1892]</td>
<td>Tries to Set a Member/Property Value (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>Write [1893]</td>
<td>Writes the value (via ADS) (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>WriteAsync [1893]</td>
<td>write as an asynchronous operation. (Inherited from DynamicValue [1869].)</td>
</tr>
<tr>
<td>WriteMember [1894]</td>
<td>Writes the specified member element. (Inherited from DynamicValue [1869].)</td>
</tr>
</tbody>
</table>

**Reference**

DynamicReferenceValue Class [1747]

TwinCAT.TypeSystem Namespace [1622]

**6.11.18.2.1 DynamicReferenceValue.ReadMember Method**

Reads the specified member element.
Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

protected override Object ReadMember(
  ISymbol memberInstance
)

Parameters

memberInstance Type: TwinCAT.TypeSystem.ISymbol

The member instance.

Return Value

Type: Object

Reference

DynamicReferenceValue Class
TwinCAT.TypeSystem Namespace

6.11.19 DynamicRpcStructInstance Class

Dynamic struct instance with RPC Methods.

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbol
    TwinCAT.TypeSystem.DynamicStructInstance
      TwinCAT.TypeSystem.DynamicRpcStructInstance

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public sealed class DynamicRpcStructInstance : DynamicStructInstance,
  IRpcStructInstance, IStructInstance, IAttributedInstance, IInstance,
  IBitSize, IRpcCallableInstance

The DynamicRpcStructInstance type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_InnerSymbol</td>
<td>Gets the inner symbol of this DynamicSymbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>AccessRights</td>
<td>Gets the access rights. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Attributes [1799]</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>BitSize [1800]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ByteSize [1800]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Category [1801]</td>
<td>Gets the category. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Comment [1801]</td>
<td>Gets the comment of the Instance [2052] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Connection [1802]</td>
<td>Gets the connection bound to this DynamicSymbol [1791] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>DataType [1802]</td>
<td>Gets the IDataType [1986] of the Instance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>HasRpcMethods [1785]</td>
<td>Gets a value indicating whether this instance has RPC methods (Inherited from DynamicStructInstance [1779].)</td>
</tr>
<tr>
<td>HasValue [1803]</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstanceName [1803]</td>
<td>Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstancePath [1804]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsBitType [1805]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsByteAligned [1805]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsContainerType [1806]</td>
<td>Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPersistent [1806]</td>
<td>Gets a value indicating whether this Symbol [2176] is persistent. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPointer [1807]</td>
<td>Indicates that the Instance [2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPrimitiveType [1807]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReadOnly [1808]</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsRecursive [1808]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReference [1809]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsStatic [1809]</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>MemberInstances [1786]</td>
<td>Gets the member instances of the Struct Instance [2158]. (Inherited from DynamicStructInstance [1779].)</td>
</tr>
<tr>
<td>NormalizedName [1810]</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

**TwinCAT.Ads Namespaces**

**Version: 1.1**

**TC1000**
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotificationSettings</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>RpcMethods</td>
<td>Gets the Method descriptions for the IRpcCallableType [2121].</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the IInstance [2052] in bytes. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>SubSymbols</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Equals (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>GetDynamicMemberNames</td>
<td>Gets the dynamic member names. (Overrides DynamicStructInstance.GetDynamicMemberNames [1789].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, Object, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>ReadAnyValue</strong> [1825]</td>
<td>Reads the value of this Value [2254] into a new created instance of the managed type (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>ReadRawValue</strong> [1826]</td>
<td>Reads the raw value of the ValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>ReadRawValue(Int32)</strong> [1826]</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>ReadRawValueAsync</strong> [1827]</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>ReadValue</strong> [1828]</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>ReadValue(Int32)</strong> [1831]</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>ReadValueAsync</strong> [1832]</td>
<td>Reads the Value of the ValueSymbol [2254] asynchronously. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>ToString</strong> [1832]</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>TryGetMember</strong> [1771]</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicStructInstance.TryGetMember(GetMemberBinder, Object.) [1789].)</td>
</tr>
<tr>
<td><strong>TryInvokeMember</strong> [1772]</td>
<td>Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method.</td>
</tr>
<tr>
<td><strong>TryInvokeRpcMethod(String, Object, Object.)</strong> [1773]</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td><strong>TryInvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object., Object.)</strong> [1775]</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td><strong>TryInvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object., Object.)</strong> [1776]</td>
<td>Invokes the the specified RpcMethod of the RpcCallableInstance [2104].</td>
</tr>
<tr>
<td><strong>TryReadValue</strong> [1834]</td>
<td>Reads the Value of the ValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>TrySetMember</strong> [1778]</td>
<td>Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Overrrides DynamicStructInstance.TrySetMember(SetMemberBinder, Object.) [1790].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryWriteValue</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>UpdateAnyValue</td>
<td>Reads the value of this Value [2254] into the specified managed value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteAnyValue</td>
<td>Writes the value represented by the managed value to this Value [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte)</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte, Int32)</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValueAsync</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValue(Object)</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValue(Object, Int32)</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValueAsync</td>
<td>Writes the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol [2254] has changed. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol [2254] has changed. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

### Reference

- TwinCAT.TypeSystem Namespace [1622]
- TwinCAT.TypeSystem.DynamicSymbol [1791]
- TwinCAT.TypeSystem.IStructInstance [2158]
- TwinCAT.TypeSystem.IRpcStructInstance [2140]
- TwinCAT.TypeSystem.IRpcCallableInstance [2104]

### 6.11.19.1 DynamicRpcStructInstance Properties

The DynamicRpcStructInstance [1752] type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerSymbol</td>
<td>Gets the inner symbol of this DynamicSymbol</td>
</tr>
<tr>
<td>AccessRights</td>
<td>Gets the access rights. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the category. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the Instance (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the connection bound to this DynamicSymbol (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType of the Instance. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>HasRpcMethods</td>
<td>Gets a value indicating whether this instance has RPC methods (Inherited from DynamicStructInstance)</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this instance has value (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods .) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this Symbol is persistent. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the Instance represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the Instance represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemberInstances [1876]</td>
<td>Gets the member instances of the Struct Instance [2158]. (Inherited from DynamicStructInstance [1779].)</td>
</tr>
<tr>
<td>NormalizedName [1810]</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>NotificationSettings [1810]</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Parent [1811]</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>RpcMethods [1758]</td>
<td>Gets the Method descriptions for the IRpcCallableType [2121]</td>
</tr>
<tr>
<td>Size [1811]</td>
<td>Gets the size of the Instance [2052] in bytes. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>SubSymbols [1812]</td>
<td>Gets the SubSymbols of the Symbol [2176] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TypeName [1812]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052], (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueEncoding [1813]</td>
<td>Gets the value encoding. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

### Reference

DynamicRpcStructInstance Class [1752]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.19.1.1 DynamicRpcStructInstance.RpcMethods Property

Gets the Method descriptions for the IRpcCallableType [2121]

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public IRpcMethodCollection RpcMethods { get; }
```

#### Property Value

Type: IRpcMethodCollection [2127]

The methods.

**Implements**

IRpcCallableInstance.RpcMethods [2105]

### Reference

DynamicRpcStructInstance Class [1752]

TwinCAT.TypeSystem Namespace [1622]
## DynamicRpcStructInstance Methods

The `DynamicRpcStructInstance` type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Equals (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>GetDynamicMemberNames</code></td>
<td>Gets the dynamic member names. (Overrides <code>DynamicStructInstance.GetDynamicMemberNames</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Gets the HashCode of the Address (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>InvokeRpcMethod(String, Object)</code></td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td><code>InvokeRpcMethod(String, Object, Object)</code></td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td><code>InvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object)</code></td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td><code>InvokeRpcMethodAsync(String, Object, CancellationToken)</code></td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td><code>InvokeRpcMethodAsync(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</code></td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td><code>ReadAnyValue</code></td>
<td>Reads the value of this <code>Value</code> into a new created instance of the managed type (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>ReadRawValue</code></td>
<td>Reads the raw value of the <code>IValueSymbol</code> (Ads Read / Write) (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>ReadRawValue(Int32)</code></td>
<td>Reads the Symbol's raw value (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>ReadRawValueAsync</code></td>
<td>Read raw value as an asynchronous operation. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>ReadValue</code></td>
<td>Reads the value of this <code>DynamicSymbol</code> (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadValue(Int32) [1831]</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValueAsync [1832]</td>
<td>Reads the Value of the IValueSymbol [2254] asynchronously. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ToString [1832]</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TryGetMember [1771]</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicStructInstance.TryGetMember(GetMemberBinder, Object.) [1789].)</td>
</tr>
<tr>
<td>TryInvokeMember [1772]</td>
<td>Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method.</td>
</tr>
<tr>
<td>TryInvokeRpcMethod (String, Object, Object.) [1773]</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>TryInvokeRpcMethod (String, Object, Object, Object.., Object...) [1775]</td>
<td>Invokes the the specified RpcMethod of the IRpcCallableInstance [2104].</td>
</tr>
<tr>
<td>TryReadValue [1834]</td>
<td>Reads the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TrySetMember [1778]</td>
<td>Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Overrides DynamicStructInstance.TrySetMember(SetMemberBinder, Object). [1790].)</td>
</tr>
<tr>
<td>TryWriteValue [1834]</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>UpdateAnyValue [1837]</td>
<td>Reads the value of this Value [2254] into the specified managed value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteAnyValue [1838]</td>
<td>Writes the value represented by the managed value to this Value [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte) [1839]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte, Int32) [1839]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WriteRawValueAsyn</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteValue(Object)</td>
<td>Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteValue(Object, Int32)</td>
<td>Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteValueAsync</td>
<td>Writes the Value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>

Reference

DynamicRpcStructInstance Class [1752]
TwinCAT.TypeSystem Namespace [1622]

6.11.19.2.1 DynamicRpcStructInstance.GetDynamicMemberNames Method

Gets the dynamic member names.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public override IEnumerable<string> GetDynamicMemberNames()
```

**Return Value**

Type: `IEnumerable<string>

**Reference**

DynamicRpcStructInstance Class [1752]
TwinCAT.TypeSystem Namespace [1622]

6.11.19.2.2 DynamicRpcStructInstance.InvokeRpcMethod Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvokeRpcMethod(String, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
</tbody>
</table>
## DynamicRpcStructInstance.InvokeRpcMethod Method (String, .Object.)

Invokes the specified RPC Method

### Reference

**DynamicRpcStructInstance Class** [1752]

**TwinCAT.TypeSystem Namespace** [1622]

### Syntax

**C#**

```csharp
public Object InvokeRpcMethod(
    string methodName,
    Object[] inParameters
)
```

### Parameters

- **methodName**
  - Type: `System.String`
  - The method name.

- **inParameters**
  - Type: `System.Object`
  - The input parameters or NULL

### Return Value

- **Type:** `Object`
  - The return value of the Method (as object).

### Implements

- `IRpcCallableInstance.InvokeRpcMethod(String, .Object.)` [2107]

### Remarks

This method only supports primitive data types as inParameters. Any available outparameters will be ignored. Complex types will fall back to byte[] arrays.
Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolLoaderMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /* (attribute 'TcRpcEnable')
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
            END_VAR
            */
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});

            // Browsing RpcMethods
            foreach(IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;
                foreach(IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}

Reference

DynamicRpcStructInstance Class [1752]
InvokeRpcMethod Overload [1761]
TwinCAT.TypeSystem Namespace [1622]

DynamicRpcStructInstance.InvokeRpcMethod Method (String, .Object., .Object..)
Invokes the specified RPC Method
Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Object InvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    out Object[] outParameters
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>methodName</td>
<td>System.String</td>
<td>The method name.</td>
</tr>
<tr>
<td>inParameters</td>
<td>System.Object</td>
<td>The input parameters or NULL</td>
</tr>
<tr>
<td>outParameters</td>
<td>System.Object</td>
<td>The output parameters.</td>
</tr>
</tbody>
</table>

Return Value

Type: Object
The return value of the Method (as object).

Implements

IRpcCallableInstance.InvokeRpcMethod(String, Object, Object) [1633]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

    // Call a Method that has the following signature (within MAIN Program)
    /* (attribute 'TcRpcEnable')*/
    METHOD PUBLIC M_Add: INT
    VAR_INPUT
    i1: INT := 0;
    i2: INT := 0;
    END_VAR
    */

    short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short)3, (short)4});

    // Call a Method that has no parameter and returns VOID
    main.InvokeRpcMethod("M_Method1", new object[]{});

    // Browsing RpcMethods
    foreach (IRpcMethod method in main.RpcMethods)
    {
        string methodName = method.Name;
        foreach (IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
    }

Reference

DynamicRpcStructInstance Class [¶ 1752]

InvokeRpcMethod Overload [¶ 1761]

TwinCAT.TypeSystem Namespace [¶ 1622]
TwinCAT.Ads Namespaces

outSpecifiers
Type: TwinCAT.TypeSystem.AnyTypeSpecifier
The out specifiers (specifying the out types) or NULL.

retSpecifier
Type: TwinCAT.TypeSystem.AnyTypeSpecifier
The ret specifier (specifying the return value) or NULL.

outParameters
Type: System.Object
The out parameters.

Return Value
Type: Object
The return value of the Method (as object).

Implements
IRpcCallableInstance.InvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object)

Remarks
The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the
parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In
case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not
necessary and should not be set.

Examples
The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode
class RpcCall1VirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader>Symbols["MAIN"];// Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /*
             * [attribute 'TcRpcEnable']
             * METHOD PUBLIC M_Add : INT
             * VAR_INPUT
             * i1 : INT := 0;
             * i2 : INT := 0;
             * END_VAR
             */

            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

            // Call a Method that has no parameter and returns VOID
main.InvokeRpcMethod("M_Method1", new object[] {});

// Browsing RpcMethods
foreach(IRpcMethod method in main.RpcMethods)
{
    string methodName = method.Name;
    foreach(IRpcMethodParameter parameter in method.Parameters)
    {
        string parameterName = parameter.Name;
        string parameterType = parameter.TypeName;
    }
}

Reference

DynamicRpcStructInstance Class [¶ 1752]

InvokeRpcMethod Overload [¶ 1761]

TwinCAT.TypeSystem Namespace [¶ 1622]

6.11.19.2.3 DynamicRpcStructInstance.InvokeRpcMethodAsync Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="1767" alt="InvokeRpcMethodAsync(String, Object, CancellationToken)" /></td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td><img src="1769" alt="InvokeRpcMethodAsync(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)" /></td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
</tbody>
</table>

Reference

DynamicRpcStructInstance Class [¶ 1752]

TwinCAT.TypeSystem Namespace [¶ 1622]

DynamicRpcStructInstance.InvokeRpcMethodAsync Method (String, .Object, CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.TypeSystem [¶ 1622]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
### Syntax

**C#**

```csharp
public Task<ResultRpcMethodAccess> InvokeRpcMethodAsync(
    string methodName,
    Object[] inParameters,
    CancellationToken cancel
)
```

### Parameters

- **methodName**
  - Type: System.String
  - The method name.

- **inParameters**
  - Type: System.Object
  - The parameters.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token

### Return Value

- **Type:** Task<ResultRpcMethodAccess>
- A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethodAccess results contain the return value (ReturnValue) together with the output parameters. The succeeded communication is indicated by the ErrorCode property (ErrorCode) after the communication.

### Implements

- IRpcCallableInstance.InvokeRpcMethodAsync(String, Object, CancellationToken)

### Remarks

Because this overload doesn't provide any AnyTypeSpecifier specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

### Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

#### Dynamic Tree Mode

```csharp
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program
        }
    }
}
```
// Call a Method that has the following signature (within MAIN Program)
/* (attribute 'TcRpcEnable')*/
METHOD PUBLIC M_Add : INT
  VAR_INPUT
  i1 : INT := 0;
  i2 : INT := 0;
  END_VAR
*/
short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short) 3, (short) 4});

// Call a Method that has no parameter and returns VOID
main.InvokeRpcMethod("M_Method1", new object[]{});

// Browsing RpcMethods
foreach(IRpcMethod method in main.RpcMethods)
{
  string methodName = method.Name;
  foreach(IRpcMethodParameter parameter in method.Parameters)
  {
    string parameterName = parameter.Name;
    string parameterType = parameter.TypeName;
  }
}

Reference

DynamicRpcStructInstance Class [1752]
InvokeRpcMethodAsync Overload [1767]
TwinCAT.TypeSystem Namespace [1622]

DynamicRpcStructInstance.InvokeRpcMethodAsync Method
(String, .Object, .AnyTypeSpecifier, .AnyTypeSpecifier, CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public Task<ResultRpcMethodAccess> InvokeRpcMethodAsync(
  string methodName,
  Object[] inParameters,
  AnyTypeSpecifier[] outSpecifiers,
  AnyTypeSpecifier retSpecifier,
  CancellationToken cancel)

Parameters

methodName Type: System.String
  The method name.

inParameters Type: System.Object
  The parameters.

outSpecifiers Type: TwinCAT.TypeSystem.AnyTypeSpecifier [1633]
  The out specifiers (specifying the out types) or NULL.
retSpecifier  
Type: TwinCAT.TypeSystem.AnyTypeSpecifier
The ret specifier (specifying the return value) or NULL.

cancel  
Type: System.Threading.CancellationToken
The cancellation token

Return Value
Type: Task.ResultRpcMethodAccess
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethodAccess results contains the return value (ReturnValue) together with the output parameters. The succeeded communication is indicated by the ErrorCode property after the communication.

Implements
IRpcCallableInstance.InvokeRpcMethodAsync(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)

Remarks
The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples
The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /*{attribute 'TcRpcEnable'}
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
                i1 : INT := 0;
                i2 : INT := 0;
            END_VAR
            */
            short result = (short)main.InvokeRpcMethod("M_Add", new object() { (short) 3, (short) 4});

            // Call a Method that has no parameter and returns VOID
        }
    }
}
main.InvokeRpcMethod("M_Method1", new object[]{})

//Browsing RpcMethods
foreach(IRpcMethod method in main.RpcMethods)
{
    string methodName = method.Name;

    foreach(IRpcMethodParameter parameter in method.Parameters)
    {
        string parameterName = parameter.Name;
        string parameterType = parameter.TypeName;
    }
}

Reference

DynamicRpcStructInstance Class [1752]
InvokeRpcMethodAsync Overload [1767]
TwinCAT.TypeSystem Namespace [1622]

6.11.19.2.4 DynamicRpcStructInstance.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public override bool TryGetMember(
    GetMemberBinder binder,
    out Object result)

Parameters

binder

Type: System.Dynamic.GetMemberBinder
Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the Console.WriteLine(sampleObject.SampleProperty) statement, where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.

result

Type: System.Object
The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)
6.11.19.2.5 DynamicRpcStructInstance.TryInvokeMember Method

Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public virtual bool TryInvokeMember(
    InvokeMemberBinder binder,
    Object[] args,
    out Object result
)
```

Parameters

- **binder**
  Type: System.Dynamic.InvokeMemberBinder
  Provides information about the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the statement sampleObject.SampleMethod(100), where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleMethod". The binder.IgnoreCase property specifies whether the member name is case-sensitive.

- **args**
  Type: System.Object
  The arguments that are passed to the object member during the invoke operation. For example, for the statement sampleObject.SampleMethod(100), where sampleObject is derived from the DynamicObject class, args][] is equal to 100.

- **result**
  Type: System.Object
  The result of the member invocation.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicRpcStructInstance Class [➤ 1752]
TwinCAT.TypeSystem Namespace [➤ 1622]
6.11.19.2.6 DynamicRpcStructInstance.TryInvokeRpcMethod Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryInvokeRpcMethod(String, Object, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, Object, Object, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object)</td>
<td>Invokes the the specified RpcMethod of the IRpcCallableInstance.</td>
</tr>
</tbody>
</table>

Reference

DynamicRpcStructInstance Class [1752]
TwinCAT.TypeSystem Namespace [1622]

DynamicRpcStructInstance.TryInvokeRpcMethod Method (String, .Object., Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int TryInvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    out Object retValue
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>methodName</td>
<td>System.String</td>
<td>The method name.</td>
</tr>
<tr>
<td>inParameters</td>
<td>System.Object</td>
<td>The parameters.</td>
</tr>
<tr>
<td>retValue</td>
<td>System.Object</td>
<td>The return value of the RPC method as object.</td>
</tr>
</tbody>
</table>
Return Value

Type: Int32
The return value of the Method (as object).

Implements

IRpcCallableInstance.TryInvokeRpcMethod(String, Object, Object) [2116]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program
            
            // Call a Method that has the following signature (within MAIN Program)
            /* {attribute 'TcRpcEnable'}
               METHOD PUBLIC M_Add : INT
               VAR_INPUT
               l1 : INT := 0;
               l2 : INT := 0;
               END_VAR
            */
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});
            
            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});
            
            // Browsing RpcMethods
            foreach(IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;
                foreach(IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}
**DynamicRpcStructInstance.TryInvokeRpcMethod Method (String, .Object., .Object.., Object.)**

Invokes the specified RPC Method

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# ```
public int TryInvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    out Object[] outParameters,
    out Object retValue
)
```

**Parameters**

| methodName | Type: System.String |
| inParameters | Type: System.Object |
| outParameters | Type: System.Object |
| retValue | Type: System.Object |

**Return Value**

Type: Int32

The ADS Error Code.

**Implements**

IRpcCallableInstance.TryInvokeRpcMethod(String, .Object., .Object.., Object.) [2118]

**Remarks**

Because this overload doesn't provide any AnyTypeSpecifier [1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

**Examples**

The following sample shows how to call (Remote Procedures / Methods) within the PLC.
Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /* {attribute 'TcRpcEnable'}
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
                i1 : INT := 0;
                i2 : INT := 0;
            END_VAR
            */
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});

            //Browsing RpcMethods
            foreach (IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;
                foreach (IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter TypeName;
                }
            }
        }
    }
}

Reference

DynamicRpcStructInstance Class [1752]
TryInvokeRpcMethod Overload [1773]
TwinCAT.TypeSystem Namespace [1622]


Invokes the the specified RpcMethod of the IRpcCallableInstance [2104].

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

```csharp
public int TryInvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters,
    out Object retValue)
```

Parameters

- **methodName**
  - Type: `System.String`
  - Name of the method.

- **inParameters**
  - Type: `System.Object`
  - The parameters.

- **outSpecifiers**
  - Type: `TwinCAT.TypeSystem.AnyTypeSpecifier[]`
  - The out specifiers (specifying the out types) or NULL.

- **retSpecifier**
  - Type: `TwinCAT.TypeSystem.AnyTypeSpecifier[]`
  - The ret specifier (specifying the return value) or NULL.

- **outParameters**
  - Type: `System.Object`
  - The out parameters.

- **retValue**
  - Type: `System.Object`
  - The return value of the RPC method.

Return Value

- Type: `Int32
  - AdsErrorCode.

Implements

`IRpcCallableInstance.TryInvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object) [2120]`

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters `inParameters`, `outParameters`, `outSpecifiers`, `retSpecifier` are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

**Dynamic Tree Mode**

```csharp
class RpcCall1VirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AdsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            // ...
        }
    }
}
```
{  
    //client.Synchronize = false;
    // Connect to the target device
    client.Connect(address);
    SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolLoadMode.VirtualTree);
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
    // Get the Symbols (Dynamic Symbols)
    IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program
    // Call a Method that has the following signature (within MAIN Program)
    /*  
        {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
          VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
          END_VAR
        */
    short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});
    // Call a Method that has no parameter and returns VOID
    main.InvokeRpcMethod("M_Method1", new object[] {});
    // Browsing RpcMethods
    foreach(IRpcMethod method in main.RpcMethods)
    {
        string methodName = method.Name;
        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
    }
}

Reference
DynamicRpcStructInstance Class [▶1752]
TryInvokeRpcMethod Overload [▶1773]
TwinCAT.TypeSystem Namespace [▶1622]

6.11.19.2.7 DynamicRpcStructInstance.TrySetMember Method

Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.

Namespace: TwinCAT.TypeSystem [▶1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public override bool TrySetMember(
    SetMemberBinder binder,
    Object value
)
Parameters

binder

Type: System.Dynamic.SetMemberBinder
Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member to which the value is being assigned. For example, for the statement sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.

value

Type: System.Object
The value to set to the member. For example, for sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, the value is "Test".

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicRpcStructInstance Class [1752]
TwinCAT.TypeSystem Namespace [1622]

6.11.19.3 DynamicRpcStructInstance Events

The DynamicRpcStructInstance [1752] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✨ RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol [2254] has changed. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>✨ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol [2254] has changed. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

Reference

DynamicRpcStructInstance Class [1752]
TwinCAT.TypeSystem Namespace [1622]

6.11.20 DynamicStructInstance Class

Dynamic struct instance

Inheritance Hierarchy

System.Dynamic.DynamicObject
TwinCAT.TypeSystem.DynamicSymbol [1791]
TwinCAT.TypeSystem.DynamicStructInstance
  TwinCAT.TypeSystem.DynamicRpcStructInstance [1752]
  TwinCAT.TypeSystem.DynamicVirtualStructInstance [1894]
The DynamicStructInstance type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerSymbol [1798]</td>
<td>Gets the inner symbol of this DynamicSymbol [1791] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>AccessRights [1798]</td>
<td>Gets the access rights. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>AllowIGPIOAccess [1799]</td>
<td>Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Attributes [1799]</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>BitSize [1800]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ByteSize [1800]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Category [1801]</td>
<td>Gets the category. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Comment [1801]</td>
<td>Gets the comment of the IInstance [2052] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Connection [1802]</td>
<td>Gets the connection bound to this DynamicSymbol [1791] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>DataType [1802]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>HasRpcMethods [1785]</td>
<td>Gets a value indicating whether this instance has RPC methods</td>
</tr>
<tr>
<td>HasValue [1803]</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstanceName [1803]</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstancePath [1804]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsBitType [1805]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsByteAligned [1805]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsContainerType [1806]</td>
<td>Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPersistent [1806]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsPointer</td>
<td>Indicates that the Instance represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the Instance represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>MemberInstances</td>
<td>Gets the member instances of the Struct Instance.</td>
</tr>
<tr>
<td>NormalizedDict</td>
<td>Dictionary of normalized Instance Names</td>
</tr>
<tr>
<td>NormalizedName</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>NotificationSettings</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the Instance in bytes. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>SubSymbols</td>
<td>Gets the SubSymbols of the Symbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType that is used for this Instance. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Equals (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>GetDynamicMemberNames</td>
<td>Returns the enumeration of all dynamic member names. (Overrides DynamicSymbol.GetDynamicMemberNames.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>OnReadAnyValue</td>
<td>Handler function for reading ADS 'Any' Values. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>OnReadRawValue</td>
<td>Handler function for reading Raw symbol value. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>OnReadRawValueAsync</td>
<td>Handler function reading the raw value of the DynamicSymbol. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OnReadValue</td>
<td>Handler function for the (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td>OnReadValueAsync</td>
<td>Handler function reading the DynamicSymbols [1791] value asynchronously. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnSetInstanceName</td>
<td>Sets a new InstanceName InstancePath (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnTryReadValue</td>
<td>Handler function for the (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td>OnTryWriteValue</td>
<td>Handler Function for writing value. (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td>OnWriteRawValue</td>
<td>Handler function for reading symbols raw value. (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td>OnWriteRawValueAsync</td>
<td>Handler function for writing the raw DynamicSymbol [1791] value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>OnWriteValue</td>
<td>Handler Function for writing value. (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td>OnWriteValueAsync</td>
<td>Handler Function for writing value. (Inherited from DynamicSymbol[1791].)</td>
</tr>
<tr>
<td>ReadAnyValue</td>
<td>Reads the value of this Value [2254] into a new created instance of the managed type (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue</td>
<td>Reads the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValueAsync</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the IValueSymbol [2254] asynchronously. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overides DynamicSymbol.TryGetMember(GetMemberBinder, Object.) [1833].)</td>
</tr>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TrySetMember</td>
<td>Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryWriteValue</td>
<td>Writes the specified value to the DynamicSymbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>UpdateAnyValue</td>
<td>Reads the value of this Value into the specified managed value. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteAnyValue</td>
<td>Writes the value represented by the managed value to this Value (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteRawValue( Byte )</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteRawValue( Byte, Int32 )</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteRawValue Async</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteValue(Object)</td>
<td>Writes the specified value to the DynamicSymbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteValue(Object, Int32)</td>
<td>Writes the specified value to the DynamicSymbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>WriteValueAsync</td>
<td>Writes the Value of the IValueSymbol (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>

**Events**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol has changed. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol has changed. (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>

**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>syncObject</td>
<td>Synchronization object (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace

**6.11.20.1 DynamicStructInstance Properties**

The DynamicStructInstance type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_InnerSymbol</td>
<td>Gets the inner symbol of this DynamicSymbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AccessRights [1799]</td>
<td>Gets the access rights. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>AllowGIOAccess [1799]</td>
<td>Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Attributes [1799]</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>BitSize [1800]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ByteSize [1800]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Category [1801]</td>
<td>Gets the category. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Comment [1801]</td>
<td>Gets the comment of the Instance [2052] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Connection [1802]</td>
<td>Gets the connection bound to this DynamicSymbol [1791] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>DataType [1802]</td>
<td>Gets the IDataType [1986] of the Instance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>HasRpcMethods [1785]</td>
<td>Gets a value indicating whether this instance has RPC methods</td>
</tr>
<tr>
<td>HasValue [1803]</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstanceName [1803]</td>
<td>Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstancePath [1804]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsBitType [1805]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsByteAligned [1805]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsContainerType [1806]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPersistent [1806]</td>
<td>Gets a value indicating whether this Symbol [2176] is persistent. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPointer [1807]</td>
<td>Indicates that the Instance [2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPrimitiveType [1807]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReadOnly [1808]</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsRecursive [1808]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReference [1809]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsStatic [1809]</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>MemberInstances [1786]</td>
<td>Gets the member instances of the Struct Instance [2158].</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NormalizedDict</td>
<td>Dictionary of normalized Instance Names</td>
</tr>
<tr>
<td>NormalizedName</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>NotificationSettings</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the IInstance in bytes. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>SubSymbols</td>
<td>Gets the SubSymbols of the ISymbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType that is used for this IInstance.</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>

Reference

DynamicStructInstance Class

TwinCAT.TypeSystem Namespace

6.11.20.1.1 DynamicStructInstance.HasRpcMethods Property

Gets a value indicating whether this instance has RPC methods

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool HasRpcMethods { get; }
```

Property Value

Type: Boolean
ture if this instance has RPC methods; otherwise, false.

Implements

IStructInstance.HasRpcMethods

Remarks

If the struct instance supports RPC Methods, then the instance class is also supporting IRpcStructInstance.
Reference
DynamicStructInstance Class [1779]
TwinCAT.TypeSystem Namespace [1622]
TwinCAT.TypeSystem.IRpcStructInstance [2140]
TwinCAT.TypeSystem.IRpcMethod [2123]
TwinCAT.TypeSystem.IRpcMethodParameter [2133]

6.11.20.1.2 DynamicStructInstance.MemberInstances Property

Gets the member instances of the Struct Instance [2158].

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public ISymbolCollection<ISymbol> MemberInstances { get; }

Property Value

Type: ISymbolCollection [2185], ISymbol [2176].
The member instances.

Implements

IStructInstance.MemberInstances [2162]

Reference
DynamicStructInstance Class [1779]
TwinCAT.TypeSystem Namespace [1622]

6.11.20.1.3 DynamicStructInstance.NormalizedDict Property

Dictionary of normalized Instance Names

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected IDictionary<string, ISymbol> NormalizedDict { get; }

Property Value

Type: IDictionary<String, ISymbol> [2176].
# DynamicStructInstance Methods

The `DynamicStructInstance` type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Equals (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>GetDynamicMemberNames</code></td>
<td>Returns the enumeration of all dynamic member names. (Overrides <code>DynamicSymbol.GetDynamicMemberNames</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Gets the HashCode of the Address (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>OnReadAnyValue</code></td>
<td>Handler function for reading ADS 'Any' Values. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>OnReadRawValue</code></td>
<td>Handler function for reading Raw symbol value. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>OnReadRawValueAsync</code></td>
<td>Handler function reading the raw value of the <code>DynamicSymbol</code>. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>OnReadValue</code></td>
<td>Handler function for the (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>OnReadValueAsync</code></td>
<td>Handler function reading the <code>DynamicSymbols</code> value asynchronously. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>OnSetInstanceName</code></td>
<td>Sets a new InstanceName InstancePath (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>OnTryReadValue</code></td>
<td>Handler function for the (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>OnTryWriteValue</code></td>
<td>Handler Function for writing value. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>OnWriteRawValue</code></td>
<td>Handler function for reading symbols raw value. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>OnWriteRawValueAsync</code></td>
<td>Handler function for writing the raw <code>DynamicSymbol</code> value. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>OnWriteValue</code></td>
<td>Handler Function for writing value. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>OnWriteValueAsync</code></td>
<td>Handler Function for writing value. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>ReadAnyValue</code></td>
<td>Reads the value of this <code>Value</code> into a new created instance of the managed type (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td><code>ReadRawValue</code></td>
<td>Reads the raw value of the <code>IValueSymbol</code> (Ads Read / Write) (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>ReadRawValue(Int32)</strong> [1826]</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol [1791]).</td>
</tr>
<tr>
<td><strong>ReadRawValueAsyc</strong> [1827]</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [1791]).</td>
</tr>
<tr>
<td><strong>ReadValue</strong> [1828]</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>ReadValue(Int32)</strong> [1831]</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>ReadValue Async</strong> [1832]</td>
<td>Reads the Value of the IValueSymbol [2254] asynchronously. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>ToString</strong> [1833]</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>TryGetMember</strong> [1789]</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicSymbol.TryGetMember(GetMemberBinder, Object.). [1833].)</td>
</tr>
<tr>
<td><strong>TryReadValue</strong> [1834]</td>
<td>Reads the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>TrySetMember</strong> [1790]</td>
<td>Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.</td>
</tr>
<tr>
<td><strong>TryWriteValue</strong> [1834]</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>UpdateAnyValue</strong> [1837]</td>
<td>Reads the value of this Value [2254] into the specified managed value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>WriteAnyValue</strong> [1838]</td>
<td>Writes the value represented by the managed value to this Value [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>WriteRawValue(Byte)</strong> [1839]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>WriteRawValue(Byte, Int32)</strong> [1839]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>WriteRawValueAsync</strong> [1840]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>WriteValue(Object)</strong> [1841]</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>WriteValue(Object, Int32)</strong> [1844]</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td><strong>WriteValueAsync</strong> [1847]</td>
<td>Writes the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>
6.11.20.2.1 DynamicStructInstance.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

**Namespace:** TwinCAT.TypeSystem [⇒ 1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public override IEnumerable<string> GetDynamicMemberNames()
```

**Return Value**

Type: `IEnumerable<String>`

A sequence that contains dynamic member names.

Reference

DynamicStructInstance Class [⇒ 1779]

TwinCAT.TypeSystem Namespace [⇒ 1622]

6.11.20.2.2 DynamicStructInstance.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the `DynamicObject` class can override this method to specify dynamic behavior for operations such as getting a value for a property.

**Namespace:** TwinCAT.TypeSystem [⇒ 1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public override bool TryGetMember(
    GetMemberBinder binder,
    out Object result
)
```

**Parameters**

`binder`

Type: `System.Dynamic.GetMemberBinder`

Provides information about the object that called the dynamic operation. The `binder.Name` property provides the name of the member on which the dynamic operation is performed. For example, for the `Console.WriteLine(sampleObject.SampleProperty)` statement, where `sampleObject` is an instance of the class derived from the `DynamicObject` class, `binder.Name` returns "SampleProperty". The `binder.IgnoreCase` property specifies whether the member name is case-sensitive.
result

Type: System.Object

The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

Return Value

Type: Boolean

true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicStructInstance Class[› 1779]

TwinCAT.TypeSystem Namespace [› 1622]

6.11.20.2.3 DynamicStructInstance.TrySetMember Method

Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.

Namespace: TwinCAT.TypeSystem [› 1622]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public virtual bool TrySetMember(
    SetMemberBinder binder,
    Object value
)
```

Parameters

binder

Type: System.Dynamic.SetMemberBinder

Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member to which the value is being assigned. For example, for the statement sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.

value

Type: System.Object

The value to set to the member. For example, for sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, the value is "Test".

Return Value

Type: Boolean

true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicStructInstance Class[› 1779]
TwinCAT.Ads Namespaces

6.11.20.3 DynamicStructInstance Events

The `DynamicStructInstance` type exposes the following members.

**Events**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the <code>IValueSymbol</code> has changed. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the <code>IValueSymbol</code> has changed. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
</tbody>
</table>

**Reference**

`DynamicStructInstance Class`[1779]

TwinCAT.TypeSystem Namespace[1622]

6.11.20.4 DynamicStructInstance Fields

The `DynamicStructInstance` type exposes the following members.

**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>syncObject</td>
<td>Synchronization object (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
</tbody>
</table>

**Reference**

`DynamicStructInstance Class`[1779]

TwinCAT.TypeSystem Namespace[1622]

6.11.21 DynamicSymbol Class

Dynamic `Symbol` object.

**Inheritance Hierarchy**

`System.Dynamic.DynamicObject`
`TwinCAT.TypeSystem.DynamicSymbol`

**Namespace:** TwinCAT.TypeSystem[1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
## TwinCAT.Ads Namespaces

### Syntax

**C#**

```csharp
public class DynamicSymbol : DynamicObject,
    IDynamicSymbol, ISymbol, IAttributedInstance, IInstance, IBitSize,
    ISymbolFactoryServicesProvider, IValueSymbol, IValueRawSymbol, IHierarchicalSymbol, IValueAccessProvider,
    IContextMaskProvider
```

The `DynamicSymbol` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>_InnerSymbol</code></td>
<td>Gets the inner symbol of this DynamicSymbol</td>
</tr>
<tr>
<td>AccessRights</td>
<td>Gets the access rights.</td>
</tr>
<tr>
<td>AllowIGIOAccess</td>
<td>Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the Symbol Attributes</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>IDataType</code> in bits.</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the category.</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the <code>IInstance</code></td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the connection bound to this DynamicSymbol</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the <code>IDataType</code> of the <code>IInstance</code>.</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations).</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.))</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full <code>DataType</code> but instead of some sort of bit mapping</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type.</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this <code>ISymbol</code> is persistent.</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the <code>IInstance</code> represents a Pointer type (Pointer TO)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type.</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only.</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsReference [1809]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE TO)</td>
</tr>
<tr>
<td>IsStatic [1809]</td>
<td>Gets a value indicating whether this instance is static.</td>
</tr>
<tr>
<td>NormalizedName [1810]</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters),</td>
</tr>
<tr>
<td>NotificationSettings [1810]</td>
<td>Gets the notification settings.</td>
</tr>
<tr>
<td>Parent [1811]</td>
<td>Gets the parent Symbol</td>
</tr>
<tr>
<td>SubSymbols [1812]</td>
<td>Gets the SubSymbols of the Symbol [2176]</td>
</tr>
<tr>
<td>TypeName [1812]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052].</td>
</tr>
<tr>
<td>ValueEncoding [1813]</td>
<td>Gets the value encoding.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1816]</td>
<td>Equals</td>
</tr>
<tr>
<td>GetDynamicMemberNames [1816]</td>
<td>Returns the enumeration of all dynamic member names.</td>
</tr>
<tr>
<td>OnReadAnyValue [1817]</td>
<td>Handler function for reading ADS 'Any' Values.</td>
</tr>
<tr>
<td>OnReadRawValue [1818]</td>
<td>Handler function for reading Raw symbol value.</td>
</tr>
<tr>
<td>OnReadValue [1819]</td>
<td>Handler function for the</td>
</tr>
<tr>
<td>OnReadValueAsync [1820]</td>
<td>Handler function reading the DynamicSymbols value asynchronously.</td>
</tr>
<tr>
<td>OnSetInstanceName [1820]</td>
<td>Sets a new InstanceName InstancePath</td>
</tr>
<tr>
<td>OnTryReadValue [1821]</td>
<td>Handler function for the</td>
</tr>
<tr>
<td>OnTryWriteValue [1821]</td>
<td>Handler Function for writing value.</td>
</tr>
<tr>
<td>OnWriteRawValue [1822]</td>
<td>Handler function for reading symbols raw value.</td>
</tr>
<tr>
<td>OnWriteRawValueAsync [1823]</td>
<td>Handler function for writing the raw DynamicSymbol value.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OnWriteValue</td>
<td>Handler Function for writing value.</td>
</tr>
<tr>
<td>OnWriteValueAsync</td>
<td>Handler Function for writing value.</td>
</tr>
<tr>
<td>ReadAnyValue</td>
<td>Reads the value of this Value [2254] into a new created instance of the</td>
</tr>
<tr>
<td>ReadRawValue</td>
<td>Reads the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the Symbols raw value</td>
</tr>
<tr>
<td>ReadRawValueAsync</td>
<td>Read raw value as an asynchronous operation.</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Reads the value of this DynamicSymbol.</td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the value of this DynamicSymbol.</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the IValueSymbol [2254] asynchronously.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance.</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values.</td>
</tr>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol [2254]</td>
</tr>
<tr>
<td>TryWriteValue</td>
<td>Writes the specified value to the DynamicSymbol.</td>
</tr>
<tr>
<td>UpdateAnyValue</td>
<td>Reads the value of this Value [2254] into the specified managed value.</td>
</tr>
<tr>
<td>WriteAnyValue</td>
<td>Writes the value represented by the managed value to this Value [2254]</td>
</tr>
<tr>
<td>WriteRawValue</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteRawValueBytes</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteRawValueBytesInt32</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteRawValueAsync</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteValue(Object)</td>
<td>Writes the specified value to the DynamicSymbol.</td>
</tr>
<tr>
<td>WriteValue(Object, Int32)</td>
<td>Writes the specified value to the DynamicSymbol.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### WriteValueAsync

- **Name**: WriteValueAsync
- **Description**: Writes the Value of the IValueSymbol

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol has changed.</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol has changed.</td>
</tr>
</tbody>
</table>

#### Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equality</td>
<td>Operator==</td>
</tr>
<tr>
<td>Inequality</td>
<td>Implements the != operator.</td>
</tr>
</tbody>
</table>

#### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>syncObject</td>
<td>Synchronization object</td>
</tr>
</tbody>
</table>

#### Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollValuesAnnotated d(Observable.Unit)</td>
<td>Overloaded. Polls the values as ValueChangedEventArgs sequence annotated value on trigger sequence (Defined by ValueSymbolExtensions).</td>
</tr>
<tr>
<td>PollValuesAnnotated d(TimeSpan)</td>
<td>Overloaded. Polls the values as ValueChangedEventArgs sequence with a specified period time. (Defined by ValueSymbolExtensions).</td>
</tr>
<tr>
<td>WhenValueChanged</td>
<td>Gets an observable sequence when the value of the IValueSymbol has changed. (Defined by ValueSymbolExtensions).</td>
</tr>
<tr>
<td>WriteValues(Observable.Object)</td>
<td>Overloaded. Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol. (Defined by ValueSymbolExtensions).</td>
</tr>
<tr>
<td>WriteValues(Observable.Object, Action.Exception)</td>
<td>Overloaded. Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol. (Defined by ValueSymbolExtensions).</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td>WriteValues(IObservable.Object, CancellationToken)</td>
<td>Overloaded. Subscribes the <a href="#2254">ValueSymbol</a> to an observable sequence of values and writes them to the <a href="#2254">ValueSymbol</a>. (Defined by ValueSymbolExtensions[1106].)</td>
</tr>
<tr>
<td>WriteValues(IObservable.Object, Action.Exception, CancellationToken)</td>
<td>Overloaded. Subscribes the <a href="#2254">ValueSymbol</a> to an observable sequence of values and writes them to the <a href="#2254">ValueSymbol</a>. (Defined by ValueSymbolExtensions[1106].)</td>
</tr>
</tbody>
</table>

**Remarks**

The DynamicSymbol adds dynamic run time behaviour to the [ISymbol](#2176)/[IValueSymbol](#2254). That means e.g. for StructSymbols that .NET Properties are defined and dispatched at runtime to the structs fields like they are defined in TwinCAT / ADS Types. Indexed access to Array Symbols is another example where the dynamic runtime support takes place.

**Examples**

Sample for the dynamic resolution of Symbols:

**Dynamic Symbol access**

```csharp
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args) {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);
    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;
    using (AdsClient client = new AdsClient())
    {
        // Connect to the target device
        client.Connect(address);

        // Usage of "dynamic" Type and Symbols (>= .NET4 only)
        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
        IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);
    }
}
```

**Reference**

- TwinCAT.TypeSystem Namespace [1622]
- TwinCAT.TypeSystem.IDynamicSymbol [2004]
- TwinCAT.TypeSystem.IValueSymbol [2254]
- TwinCAT.TypeSystem.ISymbol [2176]
- System.Dynamic.DynamicObject

**Inheritance Hierarchy**

- System.Dynamic.DynamicObject
- TwinCAT.TypeSystem.DynamicSymbol
  - TwinCAT.TypeSystem.DynamicAliasInstance [1687]
6.11.21.1 DynamicSymbol Properties

The `DynamicSymbol` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_InnerSymbol [1798]</td>
<td>Gets the inner symbol of this <code>DynamicSymbol</code> [1791]</td>
</tr>
<tr>
<td>AccessRights [1798]</td>
<td>Gets the access rights.</td>
</tr>
<tr>
<td>AllowIGIOAccess [1799]</td>
<td>Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol)</td>
</tr>
<tr>
<td>Attributes [1799]</td>
<td>Gets the Symbol Attributes</td>
</tr>
<tr>
<td>BitSize [1800]</td>
<td>Gets the size of the <code>IDataType</code> [1986] in bits.</td>
</tr>
<tr>
<td>ByteSize [1800]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes</td>
</tr>
<tr>
<td>Category [1801]</td>
<td>Gets the category.</td>
</tr>
<tr>
<td>Comment [1801]</td>
<td>Gets the comment of the <code>IInstance</code> [2052]</td>
</tr>
<tr>
<td>Connection [1802]</td>
<td>Gets the connection bound to this <code>DynamicSymbol</code> [1791]</td>
</tr>
<tr>
<td>DataType [1802]</td>
<td>Gets the <code>IDataType</code> [1986] of the <code>IInstance</code> [2052].</td>
</tr>
<tr>
<td>HasValue [1803]</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations).</td>
</tr>
<tr>
<td>InstanceName [1803]</td>
<td>Gets the name of the instance (without periods (.))</td>
</tr>
<tr>
<td>InstancePath [1804]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.))</td>
</tr>
<tr>
<td>IsBitType [1805]</td>
<td>Gets a value indicating whether this instance is not basing on a full <code>DataType</code> but instead of some sort of bit mapping</td>
</tr>
<tr>
<td>IsByteAligned [1805]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td>IsContainerType [1806]</td>
<td>Gets a value indicating whether this Symbol is a container type.</td>
</tr>
<tr>
<td>IsPersistent [1806]</td>
<td>Gets a value indicating whether this <code>Symbol</code> [2176] is persistent.</td>
</tr>
<tr>
<td>IsPointer [1807]</td>
<td>Indicates that the <code>IInstance</code> [2052] represents a Pointer type (Pointer TO)</td>
</tr>
<tr>
<td>IsPrimitiveType [1807]</td>
<td>Gets a value indicating whether this instance is a primitive type.</td>
</tr>
<tr>
<td>IsReadOnly [1808]</td>
<td>Indicates that this instance is read only.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsRecursive [1808]</td>
<td>Gets a value indicating whether this instance is recursive.</td>
</tr>
<tr>
<td>IsReference [1809]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE TO)</td>
</tr>
<tr>
<td>IsStatic [1809]</td>
<td>Gets a value indicating whether this instance is static.</td>
</tr>
<tr>
<td>NormalizedName [1810]</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters),</td>
</tr>
<tr>
<td>NotificationSettings [1810]</td>
<td>Gets the notification settings.</td>
</tr>
<tr>
<td>Parent [1811]</td>
<td>Gets the parent Symbol</td>
</tr>
<tr>
<td>SubSymbols [1812]</td>
<td>Gets the SubSymbols of the Symbol [2176]</td>
</tr>
<tr>
<td>TypeName [1812]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052].</td>
</tr>
<tr>
<td>ValueEncoding [1813]</td>
<td>Gets the value encoding.</td>
</tr>
</tbody>
</table>

**Reference**

DynamicSymbol Class [1791]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.21.1.1 DynamicSymbol._InnerSymbol Property

Gets the inner symbol of this DynamicSymbol [1791]

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```
public IValueSymbol _InnerSymbol { get; }
```

**Property Value**

Type: IValueSymbol [2254]  
The inner symbol.

**Reference**

DynamicSymbol Class [1791]  
TwinCAT.TypeSystem Namespace [1622]

### 6.11.21.1.2 DynamicSymbol.AccessRights Property

Gets the access rights.

**Reference**

DynamicSymbol Class [1791]  
TwinCAT.TypeSystem Namespace [1622]
**TwinCAT.Ads Namespaces**

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**
```csharp
public SymbolAccessRights AccessRights { get; }
```

**Property Value**

Type: SymbolAccessRights

The access rights.

**Implements**

IValueSymbol.AccessRights

**Reference**

DynamicSymbol Class

TwinCAT.TypeSystem Namespace

---

**6.11.21.1.3 DynamicSymbol.AllowIGIOAccess Property**

Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol)

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**
```csharp
protected bool AllowIGIOAccess { get; }
```

**Property Value**

Type: Boolean

**Reference**

DynamicSymbol Class

TwinCAT.TypeSystem Namespace

---

**6.11.21.1.4 DynamicSymbol.Attributes Property**

Gets the Symbol Attributes

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#
```csharp
public ITypeAttributeCollection Attributes { get; }
```

Property Value

Type: ITypeAttributeCollection
The attributes.

Implements

|AttributedInstance.Attributes

Reference

DynamicSymbol Class
TwinCAT.TypeSystem Namespace

6.11.21.1.5 DynamicSymbol.BitSize Property

Gets the size of the IDatatype in bits.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
```csharp
public int BitSize { get; }
```

Property Value

Type: Int32
The size of the bit.

Implements

IBitSize.BitSize

Reference

DynamicSymbol Class
TwinCAT.TypeSystem Namespace

6.11.21.1.6 DynamicSymbol.ByteSize Property

Gets the (aligned) size of of the Type/Instance in Bytes

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
### TwinCAT.Ads Namespaces

#### Syntax

**C#**

```csharp
public int ByteSize { get; }
```

#### Property Value

- **Type:** `Int32`
The size of the byte.

#### Implements

- `IBitSize.ByteSize`  [1984]

#### Reference

- `DynamicSymbol Class`  [1791]
- `TwinCAT.TypeSystem Namespace`  [1622]

---

#### 6.11.21.1.7 DynamicSymbol.Category Property

Gets the category.

**Namespace:** `TwinCAT.TypeSystem`  [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public DataTypeCategory Category { get; }
```

#### Property Value

- **Type:** `DataTypeCategory`  [1649]
The category.

#### Implements

- `ISymbol.Category`  [2179]

#### Reference

- `DynamicSymbol Class`  [1791]
- `TwinCAT.TypeSystem Namespace`  [1622]

---

#### 6.11.21.1.8 DynamicSymbol.Comment Property

Gets the comment of the `IInstance`  [2052]

**Namespace:** `TwinCAT.TypeSystem`  [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
6.11.21.1.9 **DynamicSymbol.Connection Property**

Gets the connection bound to this DynamicSymbol. 

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public IConnection Connection { get; }
```

### Property Value

**Type:** IConnection

The connection.

**Implements**

IValueSymbol.Connection

### Reference

DynamicSymbol Class

TwinCAT.TypeSystem Namespace

6.11.21.1.10 **DynamicSymbol.DataType Property**

Gets the IDataType of the IInstance.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public IDataType DataType { get; }
```

### Property Value

**Type:** IDataType

The data type.

**Implements**

IValueSymbol.DataType

### Reference

DynamicSymbol Class

TwinCAT.TypeSystem Namespace
Syntax

C#

```csharp
public IDataType DataType { get; }
```

**Property Value**

Type: `IDataType` [1986]
The type of the data.

**Implements**

`IInstance.DataType` [2054]

**Reference**

`DynamicSymbol Class` [1791]
`TwinCAT.TypeSystem Namespace` [1622]

6.11.21.1.11 DynamicSymbol.HasValue Property

Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations).

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool HasValue { get; }
```

**Property Value**

Type: `Boolean`
true if this instance has value; otherwise, false.

**Implements**

`IValueRawSymbol.HasValue` [2248]

**Reference**

`DynamicSymbol Class` [1791]
`TwinCAT.TypeSystem Namespace` [1622]

6.11.21.1.12 DynamicSymbol.InstanceName Property

Gets the name of the instance (without periods (.)

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public string InstanceName { get; }
```

Property Value

Type: String
The name of the instance.

Implements

IInstance.InstanceName [2054]

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.1.13 DynamicSymbol.InstancePath Property

Gets the relative / absolute access path to the instance (with periods ( . ))

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public string InstancePath { get; }
```

Property Value

Type: String
The instance path.

Implements

IInstance.InstancePath [2055]

Remarks

If this path is relative or absolute depends on the context. IMember [2065] are using relative paths, ISymbol [2176]s are using absolute ones.

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]
6.11.21.1.14 DynamicSymbol.IsBitType Property

Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```
public bool IsBitType { get; }
```

**Property Value**

Type: Boolean  
true if this instance is bit mapping; otherwise, false.

**Implements**

IBitSize.IsBitType

**Reference**

DynamicSymbol Class  
TwinCAT.TypeSystem Namespace

6.11.21.1.15 DynamicSymbol.IsByteAligned Property

Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```
public bool IsByteAligned { get; }
```

**Property Value**

Type: Boolean  
true if this instance is byte aligned; otherwise, false.

**Implements**

IBitSize.IsByteAligned

**Reference**

DynamicSymbol Class  
TwinCAT.TypeSystem Namespace
6.11.21.1.16 DynamicSymbol.IsContainerType Property

Gets a value indicating whether this Symbol is a container type.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public bool IsContainerType { get; }
```

**Property Value**

Type: `Boolean`
true if this instance is container type; otherwise, false.

**Implements**

`ISymbol.IsContainerType`

**Reference**

DynamicSymbol Class

TwinCAT.TypeSystem Namespace

6.11.21.1.17 DynamicSymbol.IsPersistent Property

Gets a value indicating whether this `ISymbol` is persistent.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public bool IsPersistent { get; }
```

**Property Value**

Type: `Boolean`
true if this instance is persistent; otherwise, false.

**Implements**

`ISymbol.IsPersistent`

**Reference**

DynamicSymbol Class

TwinCAT.TypeSystem Namespace
6.11.21.1.18 DynamicSymbol.IsPointer Property

Indicates that the Instance[2052] represents a Pointer type (Pointer TO)

Namespace: TwinCAT.TypeSystem[1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool IsPointer { get; }

Property Value

Type: Boolean
true if is ReferenceTo, otherwise false.

Implements

IInstance.IsPointer[2055]

Reference

DynamicSymbol Class[1791]
TwinCAT.TypeSystem Namespace[1622]

6.11.21.1.19 DynamicSymbol.IsPrimitiveType Property

Gets a value indicating whether this instance is a primitive type.

Namespace: TwinCAT.TypeSystem[1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool IsPrimitiveType { get; }

Property Value

Type: Boolean
true if this instance is primitive type; otherwise, false.

Implements

ISymbol.IsPrimitiveType[2180]

Reference

DynamicSymbol Class[1791]
TwinCAT.TypeSystem Namespace[1622]
6.11.21.1.20 DynamicSymbol.IsReadOnly Property

Indicates that this instance is read only.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool IsReadOnly { get; }
```

**Property Value**

Type: Boolean

**Implements**

ISymbol.IsReadOnly

**Remarks**

Actually, this Flag is restricted to TcCOM-Objects readonly Parameters. Within the PLC this is used for the ApplicationName and ProjectName of PLC instances. Write-Access on these Modules will create an DeviceAccessDenied error.

**Reference**

DynamicSymbol Class

TwinCAT.TypeSystem Namespace

6.11.21.1.21 DynamicSymbol.IsRecursive Property

Gets a value indicating whether this instance is recursive.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool IsRecursive { get; }
```

**Property Value**

Type: Boolean

true if this instance is recursive; otherwise, false.

**Implements**

ISymbol.IsRecursive
6.11.21.1.22 DynamicSymbol.IsReference Property

Indicates that the IInstance represents a Reference type (REFERENCE TO)

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool IsReference { get; }
```

**Property Value**

Type: Boolean

true if is ReferenceTo, otherwise false.

**Implements**

IInstance.IsReference

6.11.21.1.23 DynamicSymbol.IsStatic Property

Gets a value indicating whether this instance is static.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool IsStatic { get; }
```

**Property Value**

Type: Boolean

true if this instance is static; otherwise, false.

**Implements**

IInstance.IsStatic
6.11.21.1.24 DynamicSymbol.NormalizedName Property

Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters),

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public string NormalizedName { get; }

Property Value

Type: String
The normalized instance name (can be the same like InstanceName [2054])

Implements

IDynamicSymbol.NormalizedName [2007]

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.1.25 DynamicSymbol.NotificationSettings Property

Gets the notification settings.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public INotificationSettings NotificationSettings { get; set; }

Property Value

Type: INotificationSettings [972]
The notification settings.
**Implements**

IValueSymbol.NotificationSettings [2260]

**Reference**

DynamicSymbol Class [1791]

TwinCAT.TypeSystem Namespace [1622]

---

### 6.11.21.1.26 DynamicSymbol.Parent Property

Gets the parent Symbol

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ISymbol Parent { get; }
```

**Property Value**

Type: ISymbol [2176]

The parent.

**Implements**

ISymbol.Parent [2182]

---

**Reference**

DynamicSymbol Class [1791]

TwinCAT.TypeSystem Namespace [1622]

---

### 6.11.21.1.27 DynamicSymbol.Size Property

Gets the size of the IInstance [2052] in bytes.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int Size { get; }
```

**Property Value**

Type: Int32

The size of the IInstance [2052] in bytes.
6.11.21.1.28 DynamicSymbol.SubSymbols Property

Gets the SubSymbols of the ISymbol.[2176]

Namespace: TwinCAT.TypeSystem.[1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public ISymbolCollection<ISymbol> SubSymbols { get; }

Property Value

Type: ISymbolCollection.[2185] ISymbol.[2176].

Implements

ISymbol.SubSymbols.[2182]

Remarks

Only used for Array and Struct instances. Otherwise empty

Reference

DynamicSymbol Class.[1791]
TwinCAT.TypeSystem Namespace.[1622]

6.11.21.1.29 DynamicSymbol.TypeName Property

Gets the name of the DataType.[1986] that is used for this IInstance.[2052].

Namespace: TwinCAT.TypeSystem.[1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public string TypeName { get; }
Property Value

Type: String
The name of the type.

Implements

IInstance.TypeName [2056]

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.30 DynamicSymbol.ValueEncoding Property

Gets the value encoding.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public Encoding ValueEncoding { get; }

Property Value

Type: Encoding
The value encoding.

Implements

IAtributedInstance.ValueEncoding [1982]

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.2 DynamicSymbol Methods

The DynamicSymbol [1791] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals</td>
<td>Returns the enumeration of all dynamic member names.</td>
</tr>
<tr>
<td>equals</td>
<td>Returns the enumeration of all dynamic member names.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals</td>
<td>Returns the enumeration of all dynamic member names.</td>
</tr>
<tr>
<td>equals</td>
<td>Returns the enumeration of all dynamic member names.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address</td>
</tr>
<tr>
<td>OnReadAnyValue</td>
<td>Handler function for reading ADS 'Any' Values.</td>
</tr>
<tr>
<td>OnReadRawValue</td>
<td>Handler function for reading Raw symbol value.</td>
</tr>
<tr>
<td>OnReadRawValueAsync</td>
<td>Handler function reading the raw value of the <code>DynamicSymbol</code> [1791].</td>
</tr>
<tr>
<td>OnReadValue</td>
<td>Handler function for the</td>
</tr>
<tr>
<td>OnReadValueAsync</td>
<td>Handler function reading the <code>DynamicSymbols</code> [1791] value asynchronously.</td>
</tr>
<tr>
<td>OnSetInstanceName</td>
<td>Sets a new InstanceName InstancePath</td>
</tr>
<tr>
<td>OnTryReadValue</td>
<td>Handler function for the</td>
</tr>
<tr>
<td>OnTryWriteValue</td>
<td>Handler Function for writing value.</td>
</tr>
<tr>
<td>OnWriteRawValue</td>
<td>Handler function for reading symbols raw value.</td>
</tr>
<tr>
<td>OnWriteRawValueAsync</td>
<td>Handler function for writing the raw <code>DynamicSymbol</code> [1791] value.</td>
</tr>
<tr>
<td>OnWriteValue</td>
<td>Handler Function for writing value.</td>
</tr>
<tr>
<td>OnWriteValueAsync</td>
<td>Handler Function for writing value.</td>
</tr>
<tr>
<td>ReadAnyValue</td>
<td>Reads the value of this <code>Value</code> [2254] into a new created instance of the managed type</td>
</tr>
<tr>
<td>ReadRawValue</td>
<td>Reads the raw value of the <code>IValueSymbol</code> [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the Symbols raw value</td>
</tr>
<tr>
<td>ReadRawValueAsync</td>
<td>Read raw value as an asynchronous operation.</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Reads the value of this <code>DynamicSymbol</code> [1791].</td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the value of this <code>DynamicSymbol</code> [1791].</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the <code>IValueSymbol</code> [2254] asynchronously.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance.</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values.</td>
</tr>
<tr>
<td></td>
<td>Classes derived from the <code>DynamicObject</code> class can override this method to specify dynamic behavior for operations such as getting a value for a property.</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol [2254]</td>
</tr>
<tr>
<td>TryWriteValue [1834]</td>
<td>Writes the specified value to the DynamicSymbol [1791].</td>
</tr>
<tr>
<td>UpdateAnyValue [1837]</td>
<td>Reads the value of this Value [2254] into the specified managed value.</td>
</tr>
<tr>
<td>WriteAnyValue [1838]</td>
<td>Writes the value represented by the managed value to this Value [2254].</td>
</tr>
<tr>
<td>WriteRawValue(Byte) [1839]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteRawValue(Byte, Int32) [1839]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteRawValueAsync [1840]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteValue(Object) [1841]</td>
<td>Writes the specified value to the DynamicSymbol [1791].</td>
</tr>
<tr>
<td>WriteValue(Object, Int32) [1844]</td>
<td>Writes the specified value to the DynamicSymbol [1791].</td>
</tr>
<tr>
<td>WriteValueAsync [1847]</td>
<td>Writes the Value of the IValueSymbol [2254]</td>
</tr>
</tbody>
</table>

### Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollValuesAnnotate d(IObservable.Unit.) [1111]</td>
<td>Overloaded. Polls the values as ValueChangedEventArgs [2439] sequence annotated value on trigger sequence (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td>PollValuesAnnotate d(TimeSpan) [1112]</td>
<td>Overloaded. Polls the values as ValueChangedEventArgs [2439] sequence with a specified period time. (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td>WhenValueChanged [1113]</td>
<td>Gets an observable sequence when the value of the IValueSymbol [2254] has changed. (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td>WriteValues(IObservable.Object) [1117]</td>
<td>Overloaded. Subscribes the IValueSymbol [2254] to an observable sequence of values and writes them to the IValueSymbol [2254]. (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td>WriteValues(IObservable.Object, Action.Exception.) [1118]</td>
<td>Overloaded. Subscribes the IValueSymbol [2254] to an observable sequence of values and writes them to the IValueSymbol [2254]. (Defined by ValueSymbolExtensions [1106].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WriteValues(IObservable.Object, CancellationToken)</td>
<td>Overloaded. Subscribes the [ValueSymbol] to an observable sequence of values and writes them to the [ValueSymbol]. (Defined by ValueSymbolExtensions.)</td>
</tr>
<tr>
<td>WriteValues(IObservable.Object, Action.Exception, CancellationToken)</td>
<td>Overloaded. Subscribes the [ValueSymbol] to an observable sequence of values and writes them to the [ValueSymbol]. (Defined by ValueSymbolExtensions.)</td>
</tr>
</tbody>
</table>

**Reference**

DynamicSymbol Class [1791]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.21.2.1 DynamicSymbol.Equals Method

Equals

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public virtual bool Equals(
    Object obj
)
```

**Parameters**

- **obj**
  
  Type: System.Object
  
  The object to compare with the current object.

**Return Value**

Type: Boolean

true if the specified Object is equal to this instance; otherwise, false.

**Reference**

DynamicSymbol Class [1791]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.21.2.2 DynamicSymbol.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
6.11.21.2.3 DynamicSymbol.GetHashCode Method

Gets the HashCode of the Address

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#
```
public virtual int GetHashCode()
```

**Return Value**

Type: Int32
A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

**Reference**

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.2.4 DynamicSymbol.OnReadAnyValue Method

Handler function for reading ADS 'Any' Values.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#
```
protected virtual Object OnReadAnyValue(
    Type managedType)
```
Parameters
managedType

Type: System.Type
Managed type to read.

Return Value

Type: Object
System.Object.

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.2.5 DynamicSymbol.OnReadRawValue Method

Handler function for reading Raw symbol value.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

protected virtual byte[] OnReadRawValue(
    int timeout
)

Parameters

timeout

Type: System.Int32
The timeout.

Return Value

Type: Byte
System.Byte[].

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.2.6 DynamicSymbol.OnReadRawValueAsync Method

Handler function reading the raw value of the DynamicSymbol [1791].

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
protected virtual Task<ResultReadRawAccess> OnReadRawValueAsync(
    CancellationToken cancel
)
```

**Parameters**

- **cancel**  
  Type: `System.Threading.CancellationToken`  
  The cancellation token.

**Return Value**

Type: `Task<ResultReadRawAccess>`

**Reference**

- DynamicSymbol Class [1791]
- TwinCAT.TypeSystem Namespace [1622]

### 6.11.21.2.7 DynamicSymbol.OnReadValue Method

Handler function for the

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected virtual Object OnReadValue(
    int timeout
)
```

**Parameters**

- **timeout**  
  Type: `System.Int32`  
  The timeout.

**Return Value**

Type: `Object`  
System.Object.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotAccessVirtualSymbolException [1641]</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

- DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.21.2.8 DynamicSymbol.OnReadValueAsync Method

Handler function reading the DynamicSymbols [1791] value asynchronously.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```
protected virtual Task<ResultReadValueAccess> OnReadValueAsync(
    CancellationToken cancel)
```

**Parameters**

**cancel**

Type: System.Threading.CancellationToken

The cancellation token.

**Return Value**

Type: Task<ResultReadValueAccess> [2566].

An asynchronous task returning the ResultReadValueAccess [2566] as result.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotAccessVirtualSymbolException [1641]</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

DynamicSymbol Class [1791]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.21.2.9 DynamicSymbol.OnSetInstanceName Method

Sets a new InstanceName InstancePath

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```
protected virtual void OnSetInstanceName(
    string instanceName
)
```
Parameters

instanceName  Type: System.String
Instance name.

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.2.10 DynamicSymbol.OnTryReadValue Method

Handler function for the

Namespace:  TwinCAT.TypeSystem [1622]
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

protected virtual int OnTryReadValue(
    int timeout,
    out Object value
)

Parameters

timeout  Type: System.Int32
The timeout.

value  Type: System.Object
The value.

Return Value

Type: Int32
The error Code.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotAccessVirtualSymbolException [1641]</td>
<td></td>
</tr>
</tbody>
</table>

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.2.11 DynamicSymbol.OnTryWriteValue Method

Handler Function for writing value.
TwinCAT.Ads Namespaces

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected virtual int OnTryWriteValue(
    Object value,
    int timeout
)
```

**Parameters**

- **value**
  - Type: System.Object
  - The value.

- **timeout**
  - Type: System.Int32
  - The timeout.

**Return Value**

Type: Int32

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotAccessVirtualSymbolException [1641]</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

- DynamicSymbol Class [1791]
- TwinCAT.TypeSystem Namespace [1622]

### 6.11.21.2.12 DynamicSymbol.OnWriteRawValue Method

Handler function for reading symbols raw value.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected virtual void OnWriteRawValue(
    byte[] rawValue,
    int timeout
)
```

**Parameters**

- **rawValue**
  - Type: System.Byte
  - The value as byte array.

- **timeout**
  - Type: System.Int32
  - The timeout.
6.11.21.2.13 DynamicSymbol.OnWriteRawValueAsync Method

Handler function for writing the raw DynamicSymbol value.

**Namespace:** TwinCAT.TypeSystem
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
protected virtual Task<ResultWriteAccess> OnWriteRawValueAsync(
    byte[] rawValue,
    CancellationToken cancel
)
```

**Parameters**

- `rawValue`: Type: `System.Byte`
  The raw value to write.

- `cancel`: Type: `System.Threading.CancellationToken`
  The cancellation token.

**Return Value**

Type: `Task<ResultWriteAccess>`
An asynchronous task object that represents the 'OnWriteRawValue' operation and returns a `ResultWriteAccess` as result.

**Reference**

DynamicSymbol Class
TwinCAT.TypeSystem Namespace

6.11.21.2.14 DynamicSymbol.OnWriteValue Method

Handler Function for writing value.

**Namespace:** TwinCAT.TypeSystem
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
protected virtual void OnWriteValue(
    Object value,
    int timeout
)
```
Parameters

value Type: System.Object
The value.

timeout Type: System.Int32
The timeout.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotAccessVirtualSymbolException [1641]</td>
<td></td>
</tr>
</tbody>
</table>

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.2.15 DynamicSymbol.OnWriteValueAsync Method

Handler Function for writing value.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected virtual Task<ResultWriteAccess> OnWriteValueAsync(
    Object value,
    CancellationToken cancel
)
```

Parameters

value Type: System.Object
The value.

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultWriteAccess> [2575],
Task<WriteValueResult>.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotAccessVirtualSymbolException [1641]</td>
<td></td>
</tr>
</tbody>
</table>
6.11.21.2.16 DynamicSymbol.ReadAnyValue Method

Reads the value of this Value [2254] into a new created instance of the managed type

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public Object ReadAnyValue(
    TypemanagedType
)
```

**Parameters**

managedType

- **Type:** System.Type
- The tp.

**Return Value**

- **Type:** Object
- Read value (System.Object).

**Reference**

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

IValueAnySymbol.WriteAnyValue(Object) [2243]
IValueAnySymbol.UpdateAnyValue(Object) [2242]

6.11.21.2.17 DynamicSymbol.ReadRawValue Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="1826" alt="ReadRawValue" /></td>
<td>Reads the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td><img src="1826" alt="ReadRawValue" /></td>
<td>Reads the Symbols raw value</td>
</tr>
</tbody>
</table>

**Reference**

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]
DynamicSymbol.ReadRawValue Method

Reads the raw value of the `IValueSymbol` (Ads Read / Write)

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public byte[] ReadRawValue()
```

**Field Value**

Type: `Byte`
The raw value.

**Return Value**

Type: `Byte`
System.Byte[].

**Implements**

`IValueRawSymbol.ReadRawValue`. 

**Reference**

DynamicSymbol Class

ReadRawValue Overload

TwinCAT.TypeSystem Namespace

DynamicSymbol.ReadRawValue Method (Int32)

Reads the Symbols raw value

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public byte[] ReadRawValue(
    int timeout
)
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeout</td>
<td>System.Int32</td>
<td>The timeout in ms.</td>
</tr>
</tbody>
</table>
Field Value
Type: Byte
The raw value in bytes.

Return Value
Type: Byte
System.Byte[].

Implements
IValueRawSymbol.ReadRawValue(Int32) [2250]

Remarks
A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference
DynamicSymbol Class [1791]
ReadRawValue Overload [1825]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.2.18 DynamicSymbol.ReadRawValueAsync Method
Read raw value as an asynchronous operation.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fددca3e72bc0ea15da1c14

Syntax
C#
public Task<ResultReadRawAccess> ReadRawValueAsync(
    CancellationToken cancel
)

Parameters
cancel Type: System Threading CancellationToken
The cancellation token.

Field Value
Type: Task ResultReadRawAccess [2564].
The raw value.

Return Value
Type: Task ResultReadRawAccess [2564].
System.Byte[].
TwinCAT.Ads Namespaces

Implements

IValueRawSymbol.ReadRawValueAsync(CancellationToken) [2251]

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.2.19 DynamicSymbol.ReadValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>➡️ ReadValue [1828]</td>
<td>Reads the value of this DynamicSymbol [1791].</td>
</tr>
<tr>
<td>➡️ ReadValue(Int32)</td>
<td>[1831]</td>
</tr>
<tr>
<td>➡️ ReadValue(Int32)</td>
<td>Reads the value of this DynamicSymbol [1791].</td>
</tr>
</tbody>
</table>

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

DynamicSymbol.ReadValue Method

Reads the value of this DynamicSymbol [1791].

Namespace: TwinCAT.TypeSystem [1791]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public Object ReadValue()

Return Value

Type: Object
System.Object.

Implements

IValueSymbol.ReadValue [2263]
Remarks

Examples

Dynamic Read access

```csharp
namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT;
    using TwinCAT.Ads;
    using TwinCAT.Ads.TypeSystem;
    using TwinCAT.Ads.ValueAccess;
    using TwinCAT.TypeSystem;
    using TwinCAT.TypeSystem.Generic;
    using TwinCAT.ValueAccess;

    class SymbolBrowserProgramV2DynamicTree
    {
        #region CODE_SAMPLE_SIMPLEDYNAMIC
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static async void Main(string[] args)
        {
            // Get the AdsAddress from command-line arguments
            AmsAddress address = ArgParser.Parse(args);
            CancellationTokenSource cancelSource = new CancellationTokenSource();
            CancellationToken cancel = cancelSource.Token;
            using (AdsClient client = new AdsClient())
            {
                // Connect to the target device
                client.Connect(address);
                // Usage of "dynamic" Type and Symbols (>= .NET4 only)
                SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
                IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);
                #endregion

                // Set the Default setting for Notifications
                dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 200, 2000);

                // Get the Symbols (Dynamic Symbols)
                var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);
                dynamic dynamicSymbols = resultSymbols.Symbols;
                dynamic adsPort = dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.AdsPort;

                #region CODE_SAMPLE_SIMPLIFIEDYNAMIC
                // Access Main Symbol with Dynamic Language Runtime support (DLR)
                // Dynamically created property "Main"
                dynamic symMain = dynamicSymbols.Main;

                // Main is an 'VirtualSymbol' / Organizational unit that doesn't have a value
                // Calling ReadValue is not allowed
                // bool test = symMain.HasValue;

                //dynamic invalid = symMain.ReadValue();

                //Reading TaskInfo Value
                // With calling ReadValueAsync() a 'snapshot' of the Symbols Instance is taken (reading async)
                ResultReadValueAccess resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo.
                    ReadValueAsync(cancel);
                dynamic vTaskInfoArray = resultRead.Value;

                // Getting the Snapshot time in UTC format
                DateTimeOffset timeStamp1 = vTaskInfoArray.TimeStamp;
                // Getting TaskInfo Symbol for Task 1
        ```
dynamic symTaskInfo1 = dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo[1];

    // Getting CycleCount Symbol
    dynamic symCycleCount = symTaskInfo1.CycleCount;

    // Take Snapshot value of the ApplicationInfo struct
    resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.ReadValueAsync(cancel);
    dynamic vAppInfo = resultRead.Value;

    // Get the UTC Timestamp of the snapshot
    DateTimeOffset timeStamp2 = vAppInfo.TimeStamp;

    // Access the ProjectName of the ApplicationInfo Snapshot (type-safe!)
    string projectNameValue = vAppInfo.ProjectName;

    // Reading the CycleCount Value
    resultRead = await symTaskInfo1.CycleCount.ReadValueAsync(cancel); // Taking a Value Snapshot
    int cycleCountValue = (int)resultRead.Value;

    // Registering for dynamic "ValueChanged" events for the Values
    // Using Default Notification settings
    symCycleCountValueChanged += new EventHandler<ValueChangedEventArgs>(cycleCount_ValueChanged);

    // Override default notification settings
    symTaskInfo1.NotificationSettings = new NotificationSettings(AdsTransMode.Cyclic, 500, 0);

    // Register for ValueChanged event.
    symTaskInfo1ValueChanged += new EventHandler<ValueChangedEventArgs>(taskInfo1ValueChanged);

    Thread.Sleep(10000); // Sleep main thread for 10 Seconds
    }
}

static int _cycleCountEvents = 0;
/// <summary>
/// Handler function for the CycleCount ValueChanged event.
/// </summary>
/// <param name="sender">The sender.</param>
/// <param name="e">The e.</param>
static void cycleCount_ValueChanged(object sender, ValueChangedEventArgs e)
{
    lock(_notificationSynchronizer)
    {
        Interlocked.Increment(ref _cycleCountEvents);
        // val is a type safe value of int!
        dynamic val = e.Value;
        uint intVal = val;

        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert UTC to local time
        Console.WriteLine("CycleCount changed to: {0}, TimeStamp: {1}", intVal, changedTime.ToString("HH:mm:ss:fff");
    }
}

static int _taskInfo1Events = 0;
/// <summary>
/// Handler function for the TaskInfo ValueChanged event.
/// </summary>
/// <param name="sender">The sender.</param>
/// <param name="e">The e.</param>
static void taskInfo1ValueChanged(object sender, ValueChangedEventArgs e)
{
    lock (_notificationSynchronizer)
    {
        Interlocked.Increment(ref _taskInfo1Events);
        dynamic val = e.Value;
        uint intVal = val;

        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert UTC to local time
        Console.WriteLine("TaskInfo1 changed to: [0], TimeStamp: [1]", intVal, changedTime.ToString("HH:mm:ss:fff");
    }
}
DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert to local time

// Val is a during Runtime created struct type and contains
// the same Properties as related PLC object.
int cycleTime = val.CycleTime;

Console.WriteLine("TaskInfo1Value changed TimeStamp: {0}", changedTime.ToString("HH:mm:ss:fff");

Reference

DynamicSymbol Class [1791]

ReadValue Overload [1828]

TwinCAT.TypeSystem Namespace [1622]

DynamicSymbol.ReadValue Method (Int32)

Reads the value of this DynamicSymbol [1791].

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

public object ReadValue(int timeout)

Parameters

timeout Type: System.Int32
The timeout in ms.

Field Value

Type: Object
The value.

Return Value

Type: Object
System.Object.

Implements

IValueSymbol.ReadValue(Int32) [2263]

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures
and arrays) it depends on the ISymbolLoader [2200] settings what will happen. In non dynamic modes: the
raw byte Array will be returned, in dynamic mode: A Value will be created on the fly. A negative timeout
indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.
6.11.21.2.20  DynamicSymbol.ReadValueAsync Method

Reads the Value of the IValueSymbol [1254] asynchronously.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public Task<ResultReadValueAccess> ReadValueAsync(
    CancellationToken cancel
)

Parameters

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultReadValueAccess> [2566]
A tasks that represents the asynchronous 'ReadValue' operation. The read result is stored in the ResultReadValueAccess [2566] return value and contains the Value [2570] and the ErrorCode [2559].

Implements

IValueSymbol.ReadValueAsync(CancellationToken) [1264]

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly.

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.2.21  DynamicSymbol.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public virtual string ToString()
```

Return Value

Type: `String`
A `String` that represents this instance.

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.21.22 DynamicSymbol.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the `DynamicObject` class can override this method to specify dynamic behavior for operations such as getting a value for a property.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

C#

```csharp
public virtual bool TryGetMember(
    GetMemberBinder binder,
    out Object result
)
```

#### Parameters

- **binder**
  - Type: `System.Dynamic.GetMemberBinder`
  - Provides information about the object that called the dynamic operation. The `binder.Name` property provides the name of the member on which the dynamic operation is performed. For example, for the `Console.WriteLine(sampleObject.SampleProperty)` statement, where `sampleObject` is an instance of the class derived from the `DynamicObject` class, `binder.Name` returns "SampleProperty". The `binder.IgnoreCase` property specifies whether the member name is case-sensitive.

- **result**
  - Type: `System.Object`
  - The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

#### Return Value

Type: `Boolean`
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicSymbol Class [1791]
6.11.21.2.23 DynamicSymbol.TryReadValue Method

Reads the Value of the IValueSymbol.

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public int TryReadValue(
    int timeout,
    out Object value
)

Parameters

timeout 
Type: System.Int32
The timeout in ms.

value
Type: System.Object
The symbol value.

Return Value

Type: Int32
The error code.

Implements

IValueSymbol.TryReadValue(Int32, Object.)

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

DynamicSymbol Class

TwinCAT.TypeSystem Namespace

6.11.21.2.24 DynamicSymbol.TryWriteValue Method

Writes the specified value to the DynamicSymbol.

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public int TryWriteValue(
    Object value,
    int timeout
)
```

Parameters

- **value**
  - Type: `System.Object`
  - The value.

- **timeout**
  - Type: `System.Int32`
  - The timeout in ms.

Return Value

Type: `Int32`
- The error code.

Implements

`IValueSymbol.TryWriteValue(Object, Int32)` [2265]

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the `ISymbolLoader` [2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Examples

Dynamic Write access

```csharp
namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT;
    using TwinCAT.Ads;
    using TwinCAT.Ads.TypeSystem;
    using TwinCAT.Ads.ValueAccess;
    using TwinCAT.TypeSystem;
    using TwinCAT.TypeSystem.Generic;
    using TwinCAT.ValueAccess;

    class SymbolBrowserProgramV2DynamicTree
    {
        #region CODE_SAMPLE_SIMPLEDYNAMIC
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static async void Main(string[] args)
        {
            // Get the AdsAddress from command-line arguments
            AmsAddress address = ArgParser.Parse(args);

            CancellationTokenSource cancelSource = new CancellationTokenSource();
            CancellationToken cancel = cancelSource.Token;

            using (AdsClient client = new AdsClient())
```
// Connect to the target device
client.Connect(address);

// Usage of "dynamic" Type and Symbols (>= .NET4 only)
SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);

// Set the Default setting for Notifications
dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransModeOnChange, 200, 2000);

// Get the Symbols (Dynamic Symbols)
var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);
dynamic dynamicSymbols = resultSymbolsSymbols;

// Get the Symbols (Dynamic Symbols)
var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);
dynamic dynamicSymbols = resultSymbolsSymbols;

// Set the Default setting for Notifications
dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransModeOnChange, 200, 2000);

// Get the Symbols (Dynamic Symbols)
var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);
dynamic dynamicSymbols = resultSymbolsSymbols;

// Set the Default setting for Notifications
dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransModeOnChange, 200, 2000);

// Get the Symbols (Dynamic Symbols)
var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);
dynamic dynamicSymbols = resultSymbolsSymbols;

// Set the Default setting for Notifications
dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransModeOnChange, 200, 2000);

// Get the Symbols (Dynamic Symbols)
var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);
dynamic dynamicSymbols = resultSymbolsSymbols;
static object _notificationSynchronizer = new object();
static int _cycleCountEvents = 0;

/// <summary>
/// Handler function for the CycleCount ValueChanged event.
/// </summary>
/// <param name="sender">The sender.</param>
/// <param name="e">The e.</param>
static void cycleCount_ValueChanged(object sender, ValueChangedEventArgs e)
{
    lock(_notificationSynchronizer)
    {
        Interlocked.Increment(ref _cycleCountEvents);
        dynamic val = e.Value;
        uint intVal = val;
        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert UTC to local time
        Console.WriteLine("CycleCount changed to: {0}, TimeStamp: {1}", intVal, changedTime.ToString("HH:mm:ss:fff"));
    }
}

static int _taskInfo1Events = 0;

/// <summary>
/// Handler function for the TaskInfo ValueChanged event.
/// </summary>
/// <param name="sender">The sender.</param>
/// <param name="e">The e.</param>
static void taskInfo1Value_ValueChanged(object sender, ValueChangedEventArgs e)
{
    lock(_notificationSynchronizer)
    {
        Interlocked.Increment(ref _taskInfo1Events);
        dynamic val = e.Value;
        int cycleTime = val.CycleTime;
        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert to local time
        Console.WriteLine("TaskInfo1Value changed TimeStamp: {0}", changedTime.ToString("HH:mm:ss:fff"));
    }
}

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.2.25 DynamicSymbol.UpdateAnyValue Method

Reads the value of this Value [2254] into the specified managed value.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public void UpdateAnyValue(
    ref Object valueObject
)
```

**Parameters**

`valueObject`  
Type: `System.Object`  
The managed object.

**Return Value**

Type:  
Read value (System.Object).

**Reference**

DynamicSymbol Class  
TwinCAT.TypeSystem Namespace  
IValueAnySymbol.ReadAnyValue(Type)  
IValueAnySymbol.WriteAnyValue(Object)  

**6.11.21.2.26 DynamicSymbol.WriteAnyValue Method**

Writes the value represented by the managed value to this Value.

**Namespace:** TwinCAT.TypeSystem  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void WriteAnyValue(
    Object managedValue
)
```

**Parameters**

`managedValue`  
Type: `System.Object`  
The managed value.

**Reference**

DynamicSymbol Class  
TwinCAT.TypeSystem Namespace  
DynamicSymbol.ReadAnyValue(Type)  
DynamicSymbol.UpdateAnyValue(Object)  

1838  
Version: 1.1  
TC1000
6.11.21.2.27 DynamicSymbol.WriteRawValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteRawValue(byte[])</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteRawValue(byte, Int32)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write)</td>
</tr>
</tbody>
</table>

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

DynamicSymbol.WriteRawValue Method (.Byte.)

Wtites the raw value of the IValueSymbol (Ads Read / Write)

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void WriteRawValue(
    byte[] rawValue
)
```

Parameters

rawValue Type: System.Byte. The value as byte array.

Implements

IValueRawSymbol.WriteRawValue(Byte) [2252]

Reference

DynamicSymbol Class [1791]
WriteRawValue Overload [1839]
TwinCAT.TypeSystem Namespace [1622]

DynamicSymbol.WriteRawValue Method (.Byte., Int32)

Wtites the raw value of the IValueSymbol (Ads Read / Write)
Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void WriteRawValue(
    byte[] rawValue,
    int timeout
)
```

Parameters

- **(rawValue)**
  - Type: **System.Byte**
  - The value as byte array.

- **timeout**
  - Type: **System.Int32**
  - The timeout.

Field Value

Type:
The value.

Implements

Delegate **IValueRawSymbol.WriteRawValue(byte[], Int32)**

Remarks

A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

- **DynamicSymbol Class**
- **WriteRawValue Overload**
- **TwinCAT.TypeSystem Namespace**

**6.11.21.2.28 DynamicSymbol.WriteRawValueAsync Method**

Writes the raw value of the **IValueSymbol** (Ads Read / Write)

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultWriteAccess> WriteRawValueAsync(
    byte[] rawValue,
    CancellationToken cancel
)
```
Parameters

**rawValue**
- Type: `System.Byte`
- The value as byte array.

**cancel**
- Type: `System.Threading.CancellationToken`
- The cancellation token.

Return Value

**Type:** `Task<ResultWriteAccess>`
- A task that represents the asynchronous read operation. The `ResultRead` parameter contains the total number of bytes read into the buffer (`ReadBytes`) and the `ErrorCode` after execution.

Implements

- `IValueRawSymbol.WriteRawValueAsync(Byte, CancellationToken)`

Reference

- `DynamicSymbol Class` [1791]
- `TwinCAT.TypeSystem Namespace` [1622]

### 6.11.21.2.29 DynamicSymbol.WriteValue Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValue(Object)</td>
<td>Writes the specified value to the <code>DynamicSymbol</code> [1791].</td>
</tr>
<tr>
<td>WriteValue(Object, Int32)</td>
<td>Writes the specified value to the <code>DynamicSymbol</code> [1791].</td>
</tr>
</tbody>
</table>

Reference

- `DynamicSymbol Class` [1791]
- `TwinCAT.TypeSystem Namespace` [1622]

### DynamicSymbol.WriteValue Method (Object)

Writes the specified value to the `DynamicSymbol` [1791].

**Namespace:** `TwinCAT.TypeSystem` [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void WriteValue(
    Object value
)
```
Parameters
value  Type: System.Object
The value.

Implements
IValueSymbol.WriteValue(Object) [2266]

Remarks

Examples

Dynamic Write access

```csharp
namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT;
    using TwinCAT.Ads;
    using TwinCAT.Ads.TypeSystem;
    using TwinCAT.Ads.ValueAccess;
    using TwinCAT.TypeSystem.Generic;
    using TwinCAT.ValueAccess;

    class SymbolBrowserProgramV2DynamicTree
    {
        #region CODE_SAMPLE_SIMPLEDYNAMIC
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static async void Main(string[] args)
        {
            // Get the AdsAddress from command-line arguments
            AmsAddress address = ArgParser.Parse(args);
            CancellationTokenSource cancelSource = new CancellationTokenSource();
            CancellationToken cancel = cancelSource.Token;
            using (AdsClient client = new AdsClient())
            {
                // Connect to the target device
                client.Connect(address);
                // Usage of "dynamic" Type and Symbols (>= .NET4 only)
                SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
                IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);
                #endregion
                // Set the Default setting for Notifications
                dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransModeOnChange, 200, 2000);
                // Get the Symbols (Dynamic Symbols)
                var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);
                dynamic dynamicSymbols = resultSymbols.Symbols;
                dynamic adsPort = dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.AdsPort;
                #endregion
                // Access Main Symbol with Dynamic Language Runtime support (DLA)
                // Dynamically created property "Main"
                dynamic symMain = dynamicSymbols.Main;
                // Main is an 'VirtualSymbol' / Organizational unit that doesn't have a value
                // Calling ReadValue is not allowed
```
//bool test = symMain.HasValue;
//dynamic invalid = symMain.ReadValue();

//Reading TaskInfo Value
// With calling ReadValueAsync() a 'snapshot' of the Symbols Instance is taken (reading async)
ResultReadValueAccess resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo.ReadValueAsync(cancel);
dynamic vTaskInfoArray = resultRead.Value;

// Getting the Snapshot time in UTC format
DateTimeOffset timeStamp1 = vTaskInfoArray.TimeStamp;
// Getting TaskInfo Symbol for Task 1
dynamic symTaskInfo1 = dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo[1];
// Getting CycleCount Symbol
dynamic symCycleCount = symTaskInfo1.CycleCount;

// Take Snapshot value of the ApplicationInfo struct
resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.ReadValueAsync(cancel);
dynamic vAppInfo = resultRead.Value;

// Get the UTC Timestamp of the snapshot
DateTimeOffset timeStamp2 = vAppInfo.TimeStamp;
// Access the ProjectName of the ApplicationInfo Snapshot (type-safe!)
string projectNameValue = vAppInfo.ProjectName;

// Reading the CycleCount Value
resultRead = await symTaskInfo1.CycleCount.ReadValueAsync(cancel); // Taking a Value Snapshot
int cycleCountValue = (int)resultRead.Value;
#endregion

// Registering for dynamic "ValueChanged" events for the Values
// Using Default Notification settings
symCycleCount.ValueChanged += new EventHandler<ValueChangedEventArgs>(cycleCount_ValueChanged);

// Override default notification settings
symTaskInfo1.NotificationSettings = new NotificationSettings(AdsTransMode.Cyclic, 500, 0);

// Register for ValueChanged event.
symTaskInfo1.ValueChanged += new EventHandler<ValueChangedEventArgs>(taskInfo1ValueChanged);

Thread.Sleep(10000); // Sleep main thread for 10 Seconds
}
Console.WriteLine("CycleCount Changed events received: {0}", _cycleCountEvents);
Console.WriteLine("taskInfo1 Changed events received: {0}", _taskInfo1Events);
Console.WriteLine("Press [Enter] for leave:");
Console.ReadLine();
}

static object _notificationSynchronizer = new object();
static int _cycleCountEvents = 0;

/// <summary>
/// Handler function for the CycleCount ValueChanged event.
/// </summary>
/// <param name="sender">The sender.</param>
/// <param name="e">The e.</param>
static void cycleCount_ValueChanged(object sender, ValueChangedEventArgs e)
{
    lock (_notificationSynchronizer)
    {
        Interlocked.Increment(ref _cycleCountEvents);
        // val is a type safe value of int!
dynamic val = e.Value;
        uint intVal = val;

        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert UTC to local time
        Console.WriteLine("CycleCount changed to: {0}, TimeStamp: {1}". intVal, changedTime.ToString("HH:mm:ss:fff");
    }
}
static int _taskInfo1Events = 0;

/// <summary>
/// Handler function for the TaskInfo ValueChanged event.
/// </summary>
/// <param name="sender">The sender.</param>
/// <param name="e">The e.</param>
static void taskInfo1Value_ValueChanged(object sender, ValueChangedEventArgs e)
{
    lock (_notificationSynchronizer)
    {
        Interlocked.Increment(ref _taskInfo1Events);
        dynamic val = e.Value;
        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert to local time

        // Val is a during Runtime created struct type and contains
        // the same Properties as related PLC object.
        int cycleTime = val.CycleTime;
        Console.WriteLine("TaskInfo1Value changed TimeStamp: {0}", changedTime.ToString("HH:mm:ss:ff f\"\"));
    }
}

Reference

DynamicSymbol Class [1791]
WriteValue Overload [1841]
TwinCAT.TypeSystem Namespace [1622]

DynamicSymbol.WriteValue Method (Object, Int32)

Writes the specified value to the DynamicSymbol [1791].

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void WriteValue(
    Object value,
    int timeout
)
```

Parameters

value Type: System.Object
The value.

timeout Type: System.Int32
The timeout in ms.

Implements

IValueSymbol.WriteValue(Object, Int32) [2267]
Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the [SymbolLoader](#2200) settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Examples

Dynamic Write access

```csharp
namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT;
    using TwinCAT.Ads;
    using TwinCAT.Ads.TypeSystem;
    using TwinCAT.Ads.ValueAccess;
    using TwinCAT.TypeSystem;
    using TwinCAT.TypeSystem.Generic;
    using TwinCAT.ValueAccess;

    class SymbolBrowserProgramV2DynamicTree
    {
        #region CODE_SAMPLE_SIMPLEDYNAMIC
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static async void Main(string[] args)
        {
            // Get the AdsAddress from command-line arguments
            AmsAddress address = ArgParser.Parse(args);

            CancellationTokenSource cancelSource = new CancellationTokenSource();
            CancellationToken cancel = cancelSource.Token;

            using (AdsClient client = new AdsClient())
            {
                // Connect to the target device
                client.Connect(address);

                // Usage of "dynamic" Type and Symbols (>= .NET4 only)
                SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
                IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);

                // Set the Default setting for Notifications
                dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransModeOnChange, 200, 2000);

                // Get the Symbols (Dynamic Symbols)
                var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);

                dynamic dynamicSymbols = resultSymbols.Symbols;
                dynamic adsPort = dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.AdsPort;

                #endregion

                // Access Main Symbol with Dynamic Language Runtime support (DLR)
                // Dynamically created property "Main"
                //dynamic symMain = dynamicSymbols.Main;

                // Main is an 'VirtualSymbol' / Organizational unit that doesn't have a value
                // Calling ReadValue is not allowed
                //bool test = symMain.HasValue;

                // With calling ReadValueAsync() a 'snapshot' of the Symbols Instance is taken (reading async)
            }
        }
    }
}
```

```csharp
TwinCAT.Ads Namespaces

ResultReadValueAccess resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo.ReadValueAsync(cancel);
dynamic vTaskInfoArray = resultRead.Value;

// Getting the Snapshot time in UTC format
DateTimeOffset timeStamp1 = vTaskInfoArray.TimeStamp;

// Getting TaskInfo Symbol for Task 1
dynamic symTaskInfo1 = dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo[1];

// Getting CycleCount Symbol
dynamic symCycleCount = symTaskInfo1.CycleCount;

// Take Snapshot value of the ApplicationInfo struct
resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.ReadValueAsync(cancel);
dynamic vAppInfo = resultRead.Value;

// Get the UTC Timestamp of the snapshot
DateTimeOffset timeStamp2 = vAppInfo.TimeStamp;

// Access the ProjectName of the ApplicationInfo Snapshot (type-safe!)
string projectNameValue = vAppInfo.ProjectName;

// Reading the CycleCount Value
resultRead = await symTaskInfo1.CycleCount.ReadValueAsync(cancel);    // Taking a Value Snapshot
int cycleCountValue = (int)resultRead.Value;

// Registering for dynamic "ValueChanged" events for the Values
// Using Default Notification settings
symCycleCount.ValueChanged += new EventHandler<ValueChangedEventArgs>(cycleCount_ValueChanged);

// Override default notification settings
symTaskInfo1.NotificationSettings = new NotificationSettings(AdsTransMode.Cyclic, 500, 0);
// Register for ValueChanged event.
symTaskInfo1.ValueChanged += new EventHandler<ValueChangedEventArgs>(taskInfo1_ValueChanged);    // Struct Type
Thread.Sleep(10000);    // Sleep main thread for 10 Seconds
}
Console.WriteLine("CycleCount Changed events received: {0}", _cycleCountEvents);
Console.WriteLine("taskInfo1 Changed events received: {0}", _taskInfo1Events);
Console.WriteLine("Press [Enter] for leave:"");
Console.ReadLine();

static object _notificationSynchronizer = new object();
static int _cycleCountEvents = 0;

/// <summary>
/// Handler function for the CycleCount ValueChanged event.
/// </summary>
/// <param name="sender">The sender.</param>
/// <param name="e">The e.</param>
static void cycleCount_ValueChanged(object sender, ValueChangedEventArgs e)
{
    lock(_notificationSynchronizer)
    {
        Interlocked.Increment(ref _cycleCountEvents);
        // val is a type safe value of int!
dynamic val = e.Value;
        intVal = val;

        DateTimeOffset changedTime = e.DateTime.ToLocalTime();    // Convert UTC to local time
        Console.WriteLine("CycleCount changed to: [0], TimeStamp: [1]", intVal, changedTime.ToString("HH:mm:ss:fff"));
    }
}

static int _taskInfo1Events = 0;

/// <summary>
/// Handler function for the TaskInfo ValueChanged event.
/// </summary>
```
/// <param name="sender">The sender.</param>
/// <param name="e">The e.</param>
static void taskInfo1ValueChanged(object sender, ValueChangedEventArgs e)
{
    lock (_notificationSynchronizer)
    {
        Interlocked.Increment(ref _taskInfo1Events);
        dynamic val = e.Value;
        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert to local time
        // Val is a during Runtime created struct type and contains
        // the same Properties as related PLC object.
        int cycleTime = val.CycleTime;
        Console.WriteLine("TaskInfo1Value changed TimeStamp: {0}", changedTime.ToString("HH:mm:ss:ff"));
    }
}

Reference
DynamicSymbol Class [1791]
WriteValue Overload [1841]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.2.30 DynamicSymbol.WriteValueAsync Method

Writes the Value of the IValueSymbol [2254]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#

public Task<ResultWriteAccess> WriteValueAsync(
    Object value,
    CancellationToken cancel
)

Parameters
value
    Type: System.Object
    The value.

cancel
    Type: System.Threading.CancellationToken
    The cancellation token.

Return Value
Type: Task<ResultWriteAccess> [2575]
A tasks that represents the asynchronous ‘ReadValue’ operation. The read result is stored in the
ResultWriteAccess [2575] return value and contains the ErrorCode [2559].

Implements
IValueSymbol.WriteValueAsync(Object, CancellationToken) [2268]
Remarks
Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also.

Reference
DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.3 DynamicSymbol Events
The DynamicSymbol type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol has changed.</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol has changed.</td>
</tr>
</tbody>
</table>

Reference
DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.3.1 DynamicSymbol.RawValueChanged Event
Occurs when the RawValue of the IValueSymbol has changed.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public event EventHandler<RawValueChangedEventArgs> RawValueChanged
```

Value

Type: System.EventHandler<RawValueChangedEventArgs> [2289].

Implements

IValueRawSymbol.RawValueChanged [2254]

Reference

DynamicSymbol Class [1791]
6.11.21.3.2 DynamicSymbol.ValueChanged Event

Occurs when the (Primitive) value of the IValueSymbol has changed.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public event EventHandler<ValueChangedEventArgs> ValueChanged
```

Value

Type: System.EventHandler<ValueChangedEventArgs>

Implements

IValueSymbol.ValueChanged

Remarks

Examples

Use Dynamic Notifications

```csharp
using System;
using System.Diagnostics;
using System.Threading;
using TwinCAT;
using TwinCAT.Ads;
using TwinCAT.Ads.TypeSystem;
using TwinCAT.Ads.ValueAccess;
using TwinCAT.TypeSystem;
using TwinCAT.TypeSystem.Generic;

namespace Sample
{
    class SymbolBrowserV2Notifications
    {
        /// <summary>
        /// Defines the entry point of the application.
        /// </summary>
        /// <param name="args">The arguments.</param>
        static void Main(string[] args)
        {
            // Parse the Command Line Parameters.
            AmsAddress address = ArgParser.Parse(args);

            #region DEFAULTNOTIFICATON_SAMPLE
            // Create AdsClient object
            using (AdsClient client = new AdsClient())
            {
                // No automatic Synchronization (necessary for Console applications without message loop)
                client.Synchronize = false;

                // Connect to client
                client.Connect(address);

                // Usage of 'dynamic' type/symbol loader
                SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree, ValueAccessMode.IndexGroupOffsetPreferred);
            }
        }
    }
}
```
IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);

// Set the DefaultNotification Properties
dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransMode.ClientOnChange, 200, 2000);

// Determine the symbols
dynamic dynamicSymbols = ((IDynamicSymbolLoader)dynLoader).SymbolsDynamic;

// Task 1 Symbol (build in symbol)
dynamic task1Symbol = dynamicSymbols._TaskInfo[1];

// CycleCount Symbol
dynamic cycleCountSymbol = task1Symbol.CycleCount;

// Override Notification Setting for Cycle Count Symbol
cycleCountSymbol.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 250, 0);

// Register Dynamic Value Changed event.
cycleCountSymbol.ValueChanged += new EventHandler_ValueChanged;

#endregion

// Sleep main thread to receive notifications
Thread.Sleep(10000);

#endregion DEFAULTNOTIFICATION_SAMPLE

} // end region
Console.WriteLine("CycleCount changed events received: {0}", _cycleCountEvents);
Console.WriteLine("");
Console.WriteLine("Press [Enter] for leave:");
Console.ReadLine();

/// <summary>
/// The cycle count event counter.
/// </summary>
static int _cycleCountEvents = 0;

/// <summary>
/// Handler function for CycleCount changed events.
/// </summary>
/// <param name="sender">Event sender.</param>
/// <param name="args">Event arguments.</param>
static void cycleCount_ValueChanged(object sender, ValueChangedEventArgs args) {
    Interlocked.Increment(ref _cycleCountEvents);

    // Use Value as dynamic (type safe: INT) object.
    dynamic val = args.Value;
    int intVal = val;

    DateTimeOffset changedTime = args.DateTime.ToLocalTime(); // Convert UTC to local time
    Console.WriteLine("CycleCount changed to: {0}, TimeStamp: {1}", intVal, changedTime.ToString("HH:mm:ss:fff"));
}

Reference

DynamicSymbol Class [1791]

TwinCAT.TypeSystem Namespace [1622]

6.11.21.4 DynamicSymbol Operators

The DynamicSymbol [1791] type exposes the following members.
Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equality [1851]</td>
<td>Operator==</td>
</tr>
<tr>
<td>Inequality [1851]</td>
<td>Implements the != operator.</td>
</tr>
</tbody>
</table>

Reference

DynamicSymbol Class [1791]

TwinCAT.TypeSystem Namespace [1622]

6.11.21.4.1 DynamicSymbol.Equality Operator

Operator==

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static bool operator ==(
    DynamicSymbol o1,
    DynamicSymbol o2
)
```

Parameters

- o1: Type: TwinCAT.TypeSystem.DynamicSymbol [1791]
The o1.
- o2: Type: TwinCAT.TypeSystem.DynamicSymbol [1791]
The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

DynamicSymbol Class [1791]
TwinCAT.TypeSystem Namespace [1622]

6.11.21.4.2 DynamicSymbol.Inequality Operator

Implements the != operator.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public static bool operator !=(
    DynamicSymbol o1,
    DynamicSymbol o2
)
```

Parameters

- **o1**
  - Type: `TwinCAT.TypeSystem.DynamicSymbol`
  - The o1.

- **o2**
  - Type: `TwinCAT.TypeSystem.DynamicSymbol`
  - The o2.

Return Value

- Type: `Boolean`
  - The result of the operator.

Reference

- `DynamicSymbol Class`
- `TwinCAT.TypeSystem Namespace`

6.11.21.5 DynamicSymbol Fields

The `DynamicSymbol` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>syncObject</td>
<td>Synchronization object</td>
</tr>
</tbody>
</table>

Reference

- `DynamicSymbol Class`
- `TwinCAT.TypeSystem Namespace`

6.11.21.5.1 DynamicSymbol.syncObject Field

Synchronization object

Namespace: `TwinCAT.TypeSystem`
Assembly: `TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`

Syntax

C#

```csharp
protected Object syncObject
```
DynamicSymbolsCollection Class

Dynamic (Expandable) Symbols collection.

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbolsCollection

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public sealed class DynamicSymbolsCollection : DynamicObject,
  IDynamicSymbolsCollection, IDynamicMetaObjectProvider, IEnumerable<ISymbol>, IEnumerable

The DynamicSymbolsCollection type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DynamicSymbolsCollection(IEnumerable.ISymbol.)</td>
<td>Initializes a new instance of the DynamicSymbolsCollection class (for internal use only)</td>
</tr>
<tr>
<td>DynamicSymbolsCollection(SymbolCollection.ISymbol.)</td>
<td>Initializes a new instance of the DynamicSymbolsCollection class (for internal use only)</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Gets the DynamicSymbol with the specified name.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Gets an empty collection.</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDynamicMemberNames</td>
<td>Returns the enumeration of all dynamic member names.</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Gets the enumerator.</td>
</tr>
<tr>
<td>TryGetInstance</td>
<td>Tries to get the <code>IInstance</code> of the specified path.</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values. Classes derived from the <code>DynamicObject</code> class can override this method to specify dynamic behavior for operations such as getting a value for a property.</td>
</tr>
</tbody>
</table>

### Remarks

The `DynamicSymbolsCollection` collection adds dynamically its child `Symbols` as Members (for access like "Main.Symbol")

### Reference

[TwinCAT.TypeSystem Namespace](#)

### 6.11.22.1 DynamicSymbolsCollection Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DynamicSymbolsCollection(IEnumerable.ISymbol.)</code></td>
<td>Initializes a new instance of the <code>DynamicSymbolsCollection</code> class (for internal use only)</td>
</tr>
<tr>
<td><code>DynamicSymbolsCollection(SymbolCollection.ISymbol.)</code></td>
<td>Initializes a new instance of the <code>DynamicSymbolsCollection</code> class (for internal use only)</td>
</tr>
</tbody>
</table>

### Reference

[DynamicSymbolsCollection Class](#)

[TwinCAT.TypeSystem Namespace](#)

### 6.11.22.1.1 DynamicSymbolsCollection Constructor (IEnumerable.ISymbol.)

Initializes a new instance of the `DynamicSymbolsCollection` class (for internal use only)

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public DynamicSymbolsCollection(
    IEnumerable<ISymbol> symbols
)
```

Parameters

The symbols.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>symbols</td>
</tr>
</tbody>
</table>

Reference

DynamicSymbolsCollection Class [1853]
DynamicSymbolsCollection Overload [1854]
TwinCAT.TypeSystem Namespace [1622]

6.11.22.1.2 DynamicSymbolsCollection Constructor (SymbolCollection.ISymbol.)

Initializes a new instance of the `DynamicSymbolsCollection` class (for internal use only)

Namespace: `TwinCAT.TypeSystem` [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public DynamicSymbolsCollection(
    SymbolCollection<ISymbol> symbols
)
```

Parameters

symbols Type: `TwinCAT.TypeSystem.Generic.SymbolCollection<ISymbol>`.
The symbols.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>symbols</td>
</tr>
</tbody>
</table>

Reference

DynamicSymbolsCollection Class [1853]
DynamicSymbolsCollection Overload [1854]
TwinCAT.TypeSystem Namespace [1622]
6.11.22.2  DynamicSymbolsCollection Properties

The DynamicSymbolsCollection [1853] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item [1856]</td>
<td>Gets the DynamicSymbol [1791] with the specified name.</td>
</tr>
</tbody>
</table>

Reference

DynamicSymbolsCollection Class [1853]
TwinCAT.TypeSystem Namespace [1622]

6.11.22.2.1  DynamicSymbolsCollection.Item Property

Gets the DynamicSymbol [1791] with the specified name.

Namespace:  TwinCAT.TypeSystem [1622]
Assembly:  TwinCATAds.dll (in TwinCATAds.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public DynamicSymbol this[string name] { get; }
```

Parameters

name  
Type: System.String
The name.

Return Value

Type: DynamicSymbol [1791]
DynamicSymbol.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyNotFoundException</td>
<td>Symbol name not found in DynamicSymbols collection!</td>
</tr>
</tbody>
</table>

Reference

DynamicSymbolsCollection Class [1853]
TwinCAT.TypeSystem Namespace [1622]

6.11.22.3  DynamicSymbolsCollection Methods

The DynamicSymbolsCollection [1853] type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Empty</code> (<code>1857</code>)</td>
<td>Gets an empty collection.</td>
</tr>
<tr>
<td><code>GetDynamicMemberNames</code> (<code>1857</code>)</td>
<td>Returns the enumeration of all dynamic member names.</td>
</tr>
<tr>
<td><code>GetEnumerator</code> (<code>1858</code>)</td>
<td>Gets the enumerator.</td>
</tr>
<tr>
<td><code>TryGetInstance</code> (<code>1858</code>)</td>
<td>Tries to get the [Instance <code>2052</code>]. of the specified path.</td>
</tr>
<tr>
<td><code>TryGetMember</code> (<code>1859</code>)</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.</td>
</tr>
</tbody>
</table>

### Reference

- DynamicSymbolsCollection Class [1853]
- TwinCAT.TypeSystem Namespace [1622]

#### 6.11.22.3.1 DynamicSymbolsCollection.Empty Method

Gets an empty collection.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:

5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static DynamicSymbolsCollection Empty()
```

**Return Value**

Type: DynamicSymbolsCollection [1853]

DynamicSymbolsCollection.

**Reference**

- DynamicSymbolsCollection Class [1853]
- TwinCAT.TypeSystem Namespace [1622]

#### 6.11.22.3.2 DynamicSymbolsCollection.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:

5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public virtual IEnumerable<string> GetDynamicMemberNames()
```

**Return Value**

Type: `IEnumerable<String>`
A sequence that contains dynamic member names.

**Reference**

DynamicSymbolsCollection Class [1853]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.22.3.3 DynamicSymbolsCollection.GetEnumerator Method

Gets the enumerator.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdc3a3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public IEnumerator<ISymbol> GetEnumerator()
```

**Return Value**

Type: `IEnumerator<ISymbol>`
A `IEnumerator<T>` that can be used to iterate through the collection.

**Implements**

`IEnumerator<T>.GetEnumerator`.

**Reference**

DynamicSymbolsCollection Class [1853]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.22.3.4 DynamicSymbolsCollection.TryGetInstance Method

Tries to get the `IInstance` of the specified path.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdc3a3e72bc0ea15da1c14
Syntax

C#

```csharp
public bool TryGetInstance(
    string instanceSpecifier,
    out ISymbol symbol
)
```

Parameters

instanceSpecifier  Type: System.String
The instance path or Instance Name (dependent of Mode [2468] setting)
symbol  Type: TwinCAT.TypeSystem.ISymbol [2176].
The symbol.

Return Value

Type: Boolean
true if the Instance [2052] is found; otherwise, false

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>instancePath</td>
</tr>
<tr>
<td>ArgumentException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

DynamicSymbolsCollection Class [1853]
TwinCAT.TypeSystem Namespace [1622]

6.11.22.3.5  DynamicSymbolsCollection.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public virtual bool TryGetMember(
    GetMemberBinder binder,
    out Object result
)
```
Parameters

binder
Type: System.Dynamic.GetMemberBinder
Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the Console.WriteLine(sampleObject.SampleProperty) statement, where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.

result
Type: System.Object
The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

Return Value
Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference
DynamicSymbolsCollection Class [1853]
TwinCAT.TypeSystem Namespace [1622]

6.11.23 DynamicUnionInstance Class
Dynamic union instance

Inheritance Hierarchy
System.Dynamic.DynamicObject
TwinCAT.TypeSystem.DynamicSymbol [1791]
TwinCAT.TypeSystem.DynamicUnionInstance
Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#

```csharp
public sealed class DynamicUnionInstance : DynamicSymbol,
    IUnionInstance, ISymbol, IAttributedInstance, IInstance, IBitSize
```

The DynamicUnionInstance type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_InnerSymbol</td>
<td>Gets the inner symbol of this DynamicSymbol [1791] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>AccessRights</td>
<td>Gets the access rights. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType [1800] in bits. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the category. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the IInstance [2052] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the connection bound to this DynamicSymbol [1791] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType [1806] of the IInstance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>FieldInstances</td>
<td>Gets the member instances of the Struct Instance [2158].</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods .) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>NormalizedName</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>NotificationSettings</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>
### Name

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size [1811]</td>
<td>Gets the size of the Instance [2052] in bytes. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>SubSymbols [1812]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TypeName [1812]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueEncoding [1813]</td>
<td>Gets the value encoding. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [1816]</td>
<td>Equals (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>GetDynamicMemberNames [1867]</td>
<td>Returns the enumeration of all dynamic member names. (Overrides DynamicSymbol.GetDynamicMemberNames. [1816].)</td>
</tr>
<tr>
<td>GetHashCode [1817]</td>
<td>Gets the HashCode of the Address (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadAnyValue [1825]</td>
<td>Reads the value of this Value [2254] into a new created instance of the managed type (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue [1826]</td>
<td>Reads the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue(Int32) [1826]</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValueAsync [1827]</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValue [1828]</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValue(Int32) [1831]</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValueAsync [1832]</td>
<td>Reads the Value of the IValueSymbol [2254] asynchronously. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ToString [1832]</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TryGetMember [1867]</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicSymbol.TryGetMember(GetMemberBinder, Object.) [1833].)</td>
</tr>
<tr>
<td>TryReadValue [1834]</td>
<td>Reads the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TrySetMember [1868]</td>
<td>Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

**Version:** 1.1

#### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryWriteValue</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>UpdateAnyValue</td>
<td>Reads the value of this Value [2254] into the specified managed value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteAnyValue</td>
<td>Writes the value represented by the managed value to this Value [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte)</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte, Int32)</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValueAsync</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValue(Object)</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValue(Object, Int32)</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValueAsync</td>
<td>Writes the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol [2254] has changed. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol [2254] has changed. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

#### Reference

TwinCAT.TypeSystem Namespace [1622]

### 6.11.23.1 DynamicUnionInstance Properties

The DynamicUnionInstance [1860] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerSymbol</td>
<td>Gets the inner symbol of this DynamicSymbol [1791] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>AccessRights</td>
<td>Gets the access rights. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>IDataType</code> in bits. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the category. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the <code>IInstance</code>. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the connection bound to this <code>DynamicSymbol</code> (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the <code>IDataType</code> of the <code>IInstance</code>. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>FieldInstances</td>
<td>Gets the member instances of the <code>Struct Instance</code>.</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods(.) (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full <code>DataType</code> but instead of some sort of bit mapping (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this <code>Symbol</code> is persistent. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the <code>Instance</code> represents a Pointer type (Pointer TO) (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the <code>Instance</code> represents a Reference type (REFERENCE TO) (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this instance is static. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>NormalizedName</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn’t contain invalid characters), (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>NotificationSettings</td>
<td>Gets the notification settings. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
</tbody>
</table>

**TwinCAT.Ads Namespaces**
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>Gets the size of the IInstance in bytes. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>SubSymbols</td>
<td>Gets the SubSymbols of the ISymbol (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType that is used for this IInstance. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>

#### DynamicUnionInstance Class

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd6ca3e72bc0ea15da1c14

#### DynamicUnionInstance.FieldInstances Property

Gets the member instances of the Struct Instance.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd6ca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public ISymbolCollection<ISymbol> FieldInstances { get; }
```

**Property Value**

Type: ISymbolCollection<ISymbol>. The member instances.

**Implements**

IUnionInstance.FieldInstances

#### DynamicUnionInstance Methods

The DynamicUnionInstance type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Equals (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>GetDynamicMemberNames</td>
<td>Returns the enumeration of all dynamic member names. (Overrides DynamicSymbol.GetDynamicMemberNames.[1816].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>ReadAnyValue</td>
<td>Reads the value of this Value.[2254] into a new created instance of the managed type (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>ReadRawValue</td>
<td>Reads the raw value of the IValueSymbol.[2254] (Ads Read / Write) (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>ReadRawValueAsync</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Reads the value of this DynamicSymbol.[1791]. (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the value of this DynamicSymbol.[1791]. (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the IValueSymbol.[2254] asynchronously. (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicSymbol.TryGetMember(GetMemberBinder, Object).[1833].)</td>
</tr>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol.[2254] (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>TrySetMember</td>
<td>Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.</td>
</tr>
<tr>
<td>TryWriteValue</td>
<td>Writes the specified value to the DynamicSymbol.[1791]. (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>UpdateAnyValue</td>
<td>Reads the value of this Value.[2254] into the specified managed value. (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>WriteAnyValue</td>
<td>Writes the value represented by the managed value to this Value.[2254] (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte)</td>
<td>Writes the raw value of the IValueSymbol.[2254] (Ads Read / Write) (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte, Int32)</td>
<td>Writes the raw value of the IValueSymbol.[2254] (Ads Read / Write) (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>WriteRawValueAsync</td>
<td>Writes the raw value of the IValueSymbol.[2254] (Ads Read / Write) (Inherited from DynamicSymbol.[1791].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WriteValue(Object) [1841]</td>
<td>Writes the specified value to the <code>DynamicSymbol [1791]</code>. (Inherited from <code>DynamicSymbol [1791]</code>.)</td>
</tr>
<tr>
<td>WriteValue(Object, Int32) [1844]</td>
<td>Writes the specified value to the <code>DynamicSymbol [1791]</code>. (Inherited from <code>DynamicSymbol [1791]</code>.)</td>
</tr>
<tr>
<td>WriteValueAsync [1847]</td>
<td>Writes the Value of the <code>IValueSymbol [2254]</code> (Inherited from <code>DynamicSymbol [1791]</code>.)</td>
</tr>
</tbody>
</table>

### Reference

**DynamicUnionInstance Class [1860]**

**TwinCAT.TypeSystem Namespace [1622]**

#### 6.11.23.2.1 DynamicUnionInstance.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

**Namespace**: TwinCAT.TypeSystem [1622]

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public override IEnumerable<string> GetDynamicMemberNames()
```

**Return Value**

Type: `IEnumerable<String>`

A sequence that contains dynamic member names.

**Reference**

**DynamicUnionInstance Class [1860]**

**TwinCAT.TypeSystem Namespace [1622]**

#### 6.11.23.2.2 DynamicUnionInstance.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the `DynamicObject` class can override this method to specify dynamic behavior for operations such as getting a value for a property.

**Namespace**: TwinCAT.TypeSystem [1622]

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public override bool TryGetMember(
    GetMemberBinder binder,
    out Object result
)
```
**Parameters**

**binder**

| Type: System.Dynamic.GetMemberBinder |
| Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the Console.WriteLine(sampleObject.SampleProperty) statement, where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive. |

**result**

| Type: System.Object |
| The result of the get operation. For example, if the method is called for a property, you can assign the property value to result. |

**Return Value**

| Type: Boolean |
| true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.) |

**Reference**

DynamicUnionInstance Class [1860]

TwinCAT.TypeSystem Namespace [1622]

---

**6.11.23.2.3 DynamicUnionInstance.TrySetMember Method**

Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public virtual bool TrySetMember(
    SetMemberBinder binder,
    Object value
)
```

**Parameters**

**binder**

| Type: System.Dynamic.SetMemberBinder |
| Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member to which the value is being assigned. For example, for the statement sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive. |

**value**

| Type: System.Object |
| The value to set to the member. For example, for sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, the value is "Test". |
Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicUnionInstance Class [› 1860]
TwinCAT.TypeSystem Namespace [› 1622]

6.11.23.3 DynamicUnionInstance Events

The DynamicUnionInstance [› 1860] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol [› 2254] has changed.</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol [› 2254] has changed.</td>
</tr>
</tbody>
</table>

Reference

DynamicUnionInstance Class [› 1860]
TwinCAT.TypeSystem Namespace [› 1622]

6.11.24 DynamicValue Class

Dynamic value (uses RuntimeBinding for ISymbol [› 2176] value reading / writing).

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicValue
    TwinCAT.TypeSystem.DynamicPointerValue [› 1728]
    TwinCAT.TypeSystem.DynamicReferenceValue [› 1747]

Namespace: TwinCAT.TypeSystem [› 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DynamicValue : DynamicObject,
IDynamicValue, IDynamicMetaObjectProvider, IValue, IStructValue, IArrayValue
```

The DynamicValue type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Gets the age of the value (last successful read of the value)</td>
</tr>
<tr>
<td>CachedRaw</td>
<td>Gets the cached Raw internal Data.</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the data type bound to this IValue.</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IValue is a primitive value.</td>
</tr>
<tr>
<td>ResolvedType</td>
<td>Gets the resolved type.</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol that is bound to this value.</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Gets the Time stamp of the last successful read of the Value.</td>
</tr>
<tr>
<td>UpdateMode</td>
<td>Gets / Sets the update mode</td>
</tr>
<tr>
<td>ValueFactory</td>
<td>The value factory</td>
</tr>
</tbody>
</table>

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDynamicMemberNames</td>
<td>Returns the enumeration of all dynamic member names.</td>
</tr>
<tr>
<td>Read</td>
<td>Reads the value (via ADS)</td>
</tr>
<tr>
<td>ReadAsync</td>
<td>read as an asynchronous operation.</td>
</tr>
<tr>
<td>ReadMember</td>
<td>Reads the specified member element.</td>
</tr>
<tr>
<td>ResolveValue</td>
<td>Resolves the Value object to its primitive value.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance.</td>
</tr>
<tr>
<td>TryConvert</td>
<td>Provides implementation for type conversion operations. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that convert an object from one type to another.</td>
</tr>
<tr>
<td>TryGetArrayElementValues</td>
<td>Returns Array Element values.</td>
</tr>
<tr>
<td>TryGetIndex</td>
<td>Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.</td>
</tr>
<tr>
<td>TryGetIndexValue</td>
<td>Reads the specified array element.</td>
</tr>
<tr>
<td>TryGetIndexValueнт32, Object</td>
<td>Tries the get index value.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.</td>
</tr>
<tr>
<td>TryGetMemberValue</td>
<td>Tries the get member value.</td>
</tr>
<tr>
<td>TryInvoke</td>
<td>Provides the implementation for operations that invoke an object. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as invoking an object or a delegate.</td>
</tr>
<tr>
<td>TryInvokeMember</td>
<td>Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method.</td>
</tr>
<tr>
<td>TryResolveValue</td>
<td>Tries to resolves the Value object to its primitive value.</td>
</tr>
<tr>
<td>TrySetIndex</td>
<td>Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.</td>
</tr>
<tr>
<td>TrySetIndexValue</td>
<td>Tries to set the indexed value on Arrays</td>
</tr>
<tr>
<td>TrySetMember</td>
<td>Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.</td>
</tr>
<tr>
<td>TrySetMemberValue</td>
<td>Tries to Set a Member/Property Value</td>
</tr>
<tr>
<td>Write</td>
<td>Writes the value (via ADS)</td>
</tr>
<tr>
<td>WriteAsync</td>
<td>Write as an asynchronous operation.</td>
</tr>
<tr>
<td>WriteMember</td>
<td>Writes the specified member element.</td>
</tr>
</tbody>
</table>

**Remarks**

The DynamicValue adds dynamic run time behaviour to the IValue[Value][IValue]. That means e.g. for struct values that .NET Properties are on-the-fly defined and dispatched at runtime just like defined in the structs structs data type definition. Another example is the access of Array Element values through indexes.

**Examples**

Sample for the dynamic resolution of Symbols and reading values:

**Dynamic Symbol access**

```csharp
/// <summary>
/// Defines the entry point of the application.
/// </summary>
static async void Main(string[] args)
{
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);
    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;
    using (AdsClient client = new AdsClient())
```
{  
    // Connect to the target device
    client.Connect(address);

    // Usage of "dynamic" Type and Symbols (>= .NET4 only)
    SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
    IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);
}

Reference
TwinCAT.TypeSystem Namespace [1622]
TwinCAT.TypeSystem.DynamicSymbol [1791]
System.Dynamic.DynamicObject
TwinCAT.TypeSystem.IValue [2226]

6.11.24.1 DynamicValue Properties
The DynamicValue [1869] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age [1872]</td>
<td>Gets the age of the value (last successful read of the value)</td>
</tr>
<tr>
<td>CachedRaw [1873]</td>
<td>Gets the cached Raw internal Data.</td>
</tr>
<tr>
<td>DataType [1874]</td>
<td>Gets the data type bound to this IValue [2226]</td>
</tr>
<tr>
<td>IsPrimitive [1874]</td>
<td>Gets a value indicating whether this IValue [2226] is a primitive value.</td>
</tr>
<tr>
<td>ResolvedType [1875]</td>
<td>Gets the resolved type.</td>
</tr>
<tr>
<td>Symbol [1875]</td>
<td>Gets the symbol that is bound to this value.</td>
</tr>
<tr>
<td>TimeStamp [1876]</td>
<td>Gets the Time stamp of the last successful read of the Value.</td>
</tr>
<tr>
<td>UpdateMode [1876]</td>
<td>Gets / Sets the update mode</td>
</tr>
<tr>
<td>ValueFactory [1877]</td>
<td>The value factory</td>
</tr>
</tbody>
</table>

Reference
DynamicValue Class [1869]
TwinCAT.TypeSystem Namespace [1622]

6.11.24.1.1 DynamicValue.Age Property
Gets the age of the value (last successful read of the value)

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

```csharp
public TimeSpan Age { get; }
```

Property Value

Type: `TimeSpan`
The age.

Implements

`IValue.Age` [2228]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

`DynamicValue Class` [1869]

`TwinCAT.TypeSystem Namespace` [1622]

`DynamicValue.TimeStamp` [1876]

6.11.24.1.2 DynamicValue.CachedRaw Property

Gets the cached Raw internal Data.

**Namespace:** `TwinCAT.TypeSystem` [1622]

**Assembly:** `TwinCAT.Ads` (in `TwinCAT.Ads.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public byte[] CachedRaw { get; }
```

Property Value

Type: `byte[]`
The raw cached data.

Implements

`IValue.CachedRaw` [2228]

Reference

`DynamicValue Class` [1869]

`TwinCAT.TypeSystem Namespace` [1622]
6.11.24.1.3 DynamicValue.DataType Property

Gets the data type bound to this [Value][2226]

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public IDataType DataType { get; }
```

**Property Value**

Type: IDataType [1986]
The type of the data.

**Implements**

[Value.DataType][2228]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

DynamicValue Class [1869]

TwinCAT.TypeSystem Namespace [1622]

6.11.24.1.4 DynamicValue.IsPrimitive Property

Gets a value indicating whether this [Value][2226] is a primitive value.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool IsPrimitive { get; }
```

**Property Value**

Type: Boolean
true if this instance is primitive; otherwise, false.
6.11.24.1.5 DynamicValue.ResolvedType Property

Gets the resolved type.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  
protected IDataType ResolvedType { get; }  

Property Value

Type: IDataType
Resolved type.

Reference

DynamicValue Class
TwinCAT.TypeSystem Namespace

6.11.24.1.6 DynamicValue.Symbol Property

Gets the symbol that is bound to this value.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  
public ISymbol Symbol { get; }  

Property Value

Type: ISymbol
The symbol.

Reference

DynamicValue Class
TwinCAT.TypeSystem Namespace

Implements

IValue.IsPrimitive [2229]
6.11.24.1.7 DynamicValue.TimeStamp Property

Gets the Time stamp of the last successful read of the Value.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public DateTimeOffset TimeStamp { get; }

Property Value

Type: DateTimeOffset
The read time stamp.

Implements

IValue.TimeStamp

Reference

DynamicValue Class
TwinCAT.TypeSystem Namespace

6.11.24.1.8 DynamicValue.UpdateMode Property

Gets / Sets the update mode

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public ValueUpdateMode UpdateMode { get; set; }

Property Value

Type: ValueUpdateMode
The update mode.

Implements

IValue.UpdateMode
Remarks

The default value is initialized by the creating Value Factory.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immediately</td>
<td>Writes the values of this DynamicValue [1869]”/&gt; instantly when setting its</td>
</tr>
<tr>
<td></td>
<td>value or the value of its child members/elements.</td>
</tr>
<tr>
<td>Triggered</td>
<td>Caches internally the value of this DynamicValue [1869]”/&gt; until the Write.</td>
</tr>
<tr>
<td></td>
<td>[1893] method is called. This reduces ADS rountrips, if one or more member/</td>
</tr>
<tr>
<td></td>
<td>element values should be changed. Furthermore the write on the destination</td>
</tr>
<tr>
<td></td>
<td>system happens consistently in one ADS Write operation, which could be</td>
</tr>
<tr>
<td></td>
<td>important for dependent properties/members/elements.</td>
</tr>
</tbody>
</table>

Reference

DynamicValue Class [1869]

TwinCAT.TypeSystem Namespace [1622]

TwinCAT.ValueUpdateMode [150]

6.11.24.1.9 DynamicValue.ValueFactory Property

The value factory

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

protected IAccessorValueFactory ValueFactory { get; }

Property Value

Type: IAccessorValueFactory [2552]

Reference

DynamicValue Class [1869]

TwinCAT.TypeSystem Namespace [1622]

6.11.24.2 DynamicValue Methods

The DynamicValue [1869] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDynamicMemberNames</td>
<td>Returns the enumeration of all dynamic member names.</td>
</tr>
<tr>
<td>Read</td>
<td>Reads the value (via ADS)</td>
</tr>
<tr>
<td>ReadAsync</td>
<td>read as an asynchronous operation.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ReadMember</td>
<td>Reads the specified member element.</td>
</tr>
<tr>
<td>ResolveValue</td>
<td>Resolves the Value object to its primitive value.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance.</td>
</tr>
<tr>
<td>TryConvert</td>
<td>Provides implementation for type conversion operations. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that convert an object from one type to another.</td>
</tr>
<tr>
<td>TryGetArrayElementValues</td>
<td>Returns Array Element values.</td>
</tr>
<tr>
<td>TryGetIndex</td>
<td>Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.</td>
</tr>
<tr>
<td>TryGetIndexValue (&lt;int&gt;, Object)</td>
<td></td>
</tr>
<tr>
<td>TryGetIndexValue( Object, Object)</td>
<td></td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.</td>
</tr>
<tr>
<td>TryGetMemberValue</td>
<td>Tries the get member value.</td>
</tr>
<tr>
<td>TryInvoke</td>
<td>Provides the implementation for operations that invoke an object. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as invoking an object or a delegate.</td>
</tr>
<tr>
<td>TryInvokeMember</td>
<td>Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method.</td>
</tr>
<tr>
<td>TryResolveValue</td>
<td>Tries to resolves the Value object to its primitive value.</td>
</tr>
<tr>
<td>TrySetIndex</td>
<td>Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.</td>
</tr>
<tr>
<td>TrySetIndexValue</td>
<td>Tries to set the indexed value on Arrays</td>
</tr>
<tr>
<td>TrySetMember</td>
<td>Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.</td>
</tr>
<tr>
<td>TrySetMemberValue</td>
<td>Tries to Set a Member/Property Value</td>
</tr>
<tr>
<td>Write</td>
<td>Writes the value (via ADS)</td>
</tr>
<tr>
<td>WriteAsync</td>
<td>write as an asynchronous operation.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>WriteMember</td>
<td>Writes the specified member element.</td>
</tr>
</tbody>
</table>

**Reference**

DynamicValue Class [1869]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.24.2.1 DynamicValue.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll)  
**Version:** 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public virtual IEnumerable<string> GetDynamicMemberNames()
```

**Return Value**

Type: `IEnumerable<String>`  
A sequence that contains dynamic member names.

**Reference**

DynamicValue Class [1869]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.24.2.2 DynamicValue.Read Method

Reads the value (via ADS)

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll)  
**Version:** 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void Read()
```

**Implements**

`IValue.Read` [2231]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolException</td>
<td>[2401]</td>
</tr>
</tbody>
</table>
6.11.24.2.3 DynamicValue.ReadAsync Method

read as an asynchronous operation.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Task<ResultAccess> ReadAsync(
    CancellationToken cancel
)
```

Parameters

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultAccess>
Task<ReadValueResult>.

Implements

IValue.ReadAsync(CancellationToken)

Reference

DynamicValue Class
TwinCAT.TypeSystem Namespace

6.11.24.2.4 DynamicValue.ReadMember Method

Reads the specified member element.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected virtual Object ReadMember(
    ISymbol memberInstance
)
```
Parameters
memberInstance  Type: TwinCAT.TypeSystem.ISymbol
The member instance.

Return Value
Type: Object

Reference
DynamicValue Class [1869]
TwinCAT.TypeSystem Namespace [1622]

6.11.24.2.5 DynamicValue.ResolveValue Method
Resolves the Value object to its primitive value.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax
C#
public Object ResolveValue(
    bool resolveEnumToPrimitive
)

Parameters
resolveEnumToPrimitive  Type: System.Boolean
if set to true, EnumValue [2028]s are resolved to their primitives also.

Return Value
Type: Object
System.Object.

Implements
IValue.ResolveValue(Boolean) [2232]

Remarks
If the value is not primitive, this method returns the IValue [2226] itself.

Reference
DynamicValue Class [1869]
TwinCAT.TypeSystem Namespace [1622]
6.11.24.2.6 **DynamicValue.ToString Method**

Returns a `String` that represents this instance.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public virtual string ToString()
```

**Return Value**

Type: `String`

A `String` that represents this instance.

**Reference**

DynamicValue Class

TwinCAT.TypeSystem Namespace

6.11.24.2.7 **DynamicValue.TryConvert Method**

Provides implementation for type conversion operations. Classes derived from the `DynamicObject` class can override this method to specify dynamic behavior for operations that convert an object from one type to another.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public virtual bool TryConvert(
    ConvertBinder binder,
    out Object result
)
```

**Parameters**

- **binder**
  Type: `System.Dynamic.ConvertBinder`
  Provides information about the conversion operation. The binder.Type property provides the type to which the object must be converted. For example, for the statement `(String)sampleObject` in C# (CType(sampleObject, Type) in Visual Basic), where `sampleObject` is an instance of the class derived from the `DynamicObject` class, binder.Type returns the `String` type. The binder.Explicit property provides information about the kind of conversion that occurs. It returns true for explicit conversion and false for implicit conversion.

- **result**
  Type: `System.Object`
  The result of the type conversion operation.
Return Value

Type: Boolean
ture if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicValue Class [1869]
TwinCAT.TypeSystem Namespace [1622]

6.11.24.2.8 DynamicValue.TryGetArrayElementValues Method

Returns Array Element values.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool TryGetArrayElementValues(
    out IEnumerable<Object> elementValues
)
```

Parameters

- elementValues
  Type: System.Collections.Generic.IEnumerable<Object>
The element values.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IArrayValue.TryGetArrayElementValues(IEnumerable.Object..) [1978]

Reference

DynamicValue Class [1869]
TwinCAT.TypeSystem Namespace [1622]

6.11.24.2.9 DynamicValue.TryGetIndex Method

Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C# public virtual bool TryGetIndex(
    GetIndexBinder binder,
    Object[] indexes,
    out Object result
)

Parameters

binder Type: System.Dynamic.GetIndexBinder
Provides information about the operation.

indexes Type: System.Object
The indexes that are used in the operation. For example, for the sampleObject[3] operation in C# (sampleObject(3) in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[0] is equal to 3.

result Type: System.Object
The result of the index operation.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicValue Class [1869]
TwinCAT.TypeSystem Namespace [1622]

6.11.24.2.10 DynamicValue.TryGetIndexValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ] TryGetIndexValue(Int32, Object) [1884]</td>
<td>Reads the specified array element.</td>
</tr>
<tr>
<td>![ ] TryGetIndexValue(Object, Object) [1885]</td>
<td>Tries the get index value.</td>
</tr>
</tbody>
</table>

Reference

DynamicValue Class [1869]
TwinCAT.TypeSystem Namespace [1622]

DynamicValue.TryGetIndexValue Method (.Int32., Object.)

Reads the specified array element.
## TwinCAT.Ads Namespaces

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public bool TryGetIndexValue(
    int[] indices,
    out Object value
)
```

### Parameters

- **indices**
  - Type: `System.Int32`
  - The indices.

- **value**
  - Type: `System.Object`
  - The value.

### Return Value

- Type: `Boolean`
- `System.Object`

### Implements

- `IArrayValue.TryGetIndexValue(Int32, Object)`

### Reference

- **DynamicValue Class**
- **TryGetIndexValue Overload**
- **TwinCAT.TypeSystem Namespace**

### DynamicValue.TryGetIndexValue Method (.Object., Object.)

Tries the get index value.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public bool TryGetIndexValue(
    Object[] indexes,
    out Object result
)
```

### Parameters

- **indexes**
  - Type: `System.Object`
  - The indexes.

- **result**
  - Type: `System.Object`
  - The result.
Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

DynamicValue Class [1869]
TryGetValueOverload [1884]
TwinCAT.TypeSystem Namespace [1622]

6.11.24.2.11 DynamicValue.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public virtual bool TryGetMember(
    GetMemberBinder binder,
    out Object result
)

Parameters

binder

Type: System Dynamic GetMemberBinder
Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the Console.WriteLine(sampleObject.SampleProperty) statement, where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.

result

Type: System Object.
The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicValue Class [1869]
TwinCAT.TypeSystem Namespace [1622]
6.11.24.2.12 DynamicValue.TryGetMemberValue Method

Tries the get member value.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public virtual bool TryGetMemberValue(
    string name,
    out Object result
)
```

**Parameters**

- **name**
  - Type: System.String
  - The name.

- **result**
  - Type: System.Object
  - The result.

**Return Value**

Type: Boolean
true if XXXX, false otherwise.

**Implements**

IStructValue.TryGetMemberValue(String, Object.)

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

- DynamicValue Class
- TwinCAT.TypeSystem Namespace

6.11.24.2.13 DynamicValue.TryInvoke Method

Provides the implementation for operations that invoke an object. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as invoking an object or a delegate.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public virtual bool TryInvoke(
    InvokeBinder binder,
    Object[] args,
    out Object result
)
```

Parameters

binder Type: System.Dynamic.InvokeBinder
Provides information about the invoke operation.

args Type: System.Object
The arguments that are passed to the object during the invoke operation. For example, for the sampleObject(100) operation, where sampleObject is derived from the DynamicObject class, args is equal to 100.

result Type: System.Object
The result of the object invocation.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.

Reference

DynamicValue Class [1869]
TwinCAT.TypeSystem Namespace [1622]

6.11.24.14 DynamicValue.TryInvokeMember Method

Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public virtual bool TryInvokeMember(
    InvokeMemberBinder binder,
    Object[] args,
    out Object result
)
```

Parameters

binder Type: System.Dynamic.InvokeMemberBinder
Provides information about the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the statement sampleObject.SampleMethod(100), where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleMethod". The binder.IgnoreCase property specifies whether the member name is case-sensitive.
args

Type: System.Object.
The arguments that are passed to the object member during the invoke operation. For example, for the statement sampleObject.SampleMethod(100), where sampleObject is derived from the DynamicObject class, args[0] is equal to 100.

result

Type: System.Object.
The result of the member invocation.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicValue Class [1869]
TwinCAT.TypeSystem Namespace [1622]

6.11.24.2.15 DynamicValue.TryResolveValue Method

Tries to resolves the Value object to its primitive value.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public bool TryResolveValue(
    bool resolveEnumToPrimitive,
    out Object value
)

Parameters

resolveEnumToPrimitive

Type: System.Boolean
if set to true IEnumValue [2028]s are resolved to their primitives also.

value

Type: System.Object.
The value.

Return Value

Type: Boolean
true if value can be resolved, false otherwise.

Implements

IValue.TryResolveValue(Boolean, Object.) [2232]

Reference

DynamicValue Class [1869]
TwinCAT.TypeSystem Namespace [1622]
6.11.24.2.16 DynamicValue.TrySetIndex Method

Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public virtual bool TrySetIndex(
    SetIndexBinder binder,
    Object[] indexes,
    Object value
)
```

### Parameters

- **binder**
  - Type: `System.Dynamic.SetIndexBinder`
  - Provides information about the operation.

- **indexes**
  - Type: `System.Object`
  - The indexes that are used in the operation. For example, for the `sampleObject[3] = 10` operation in C# (sampleObject(3) = 10 in Visual Basic), where `sampleObject` is derived from the `DynamicObject` class, `indexes[0]` is equal to 3.

- **value**
  - Type: `System.Object`
  - The value to set to the object that has the specified index. For example, for the `sampleObject[3] = 10` operation in C# (sampleObject(3) = 10 in Visual Basic), where `sampleObject` is derived from the `DynamicObject` class, `value` is equal to 10.

### Return Value

- Type: `Boolean`
  - true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.

### Reference

- DynamicValue Class
- TwinCAT.TypeSystem Namespace

6.11.24.2.17 DynamicValue.TrySetIndexValue Method

Tries to set the indexed value on Arrays

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
### Syntax

**C#**

```csharp
public bool TrySetIndexValue(
    Object[] indexes,
    Object value
)
```

#### Parameters

- **indexes**
  - Type: `System.Object`
  - The indexes.

- **value**
  - Type: `System.Object`
  - The value.

#### Return Value

Type: `Boolean`

true if succeeded, false otherwise.

#### Implements

`IArrayValue.TrySetIndexValue(Object, Object)`

#### Reference

- **TwinCAT.TypeSystem Namespace**

`DynamicValue.TrySetMember Method`

Provides the implementation for operations that set member values. Classes derived from the `DynamicObject` class can override this method to specify dynamic behavior for operations such as setting a value for a property.

**Namespace:** `TwinCAT.TypeSystem`

**Assembly:** `TwinCAT.Ads` (in `TwinCAT.Ads.dll` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14)

### Syntax

**C#**

```csharp
public virtual bool TrySetMember(
    SetMemberBinder binder,
    Object value
)
```

#### Parameters

- **binder**
  - Type: `System.Dynamic.SetMemberBinder`
  - Provides information about the object that called the dynamic operation. The `binder.Name` property provides the name of the member to which the value is being assigned. For example, for the statement `sampleObject.SampleProperty = "Test"`, where `sampleObject` is an instance of the class derived from the `DynamicObject` class, `binder.Name` returns "SampleProperty". The `binder.IgnoreCase` property specifies whether the member name is case-sensitive.
value

Type: System.Object
The value to set to the member. For example, for sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, the value is "Test".

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicValue Class [1869]
TwinCAT.TypeSystem Namespace [1622]

6.11.24.2.19 DynamicValue.TrySetMemberValue Method

Tries to Set a Member/Property Value

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public bool TrySetMemberValue(
    string name,
    Object value
)

Parameters

name

Type: System.String
The name of the member

value

Type: System.Object
The value.

Return Value

Type: Boolean
true if succeeded, otherwise false otherwise.

Implements

IStructValue.TrySetMemberValue(String, Object) [2170]

Reference

DynamicValue Class [1869]
TwinCAT.TypeSystem Namespace [1622]
6.11.24.2.20 DynamicValue.Write Method

Writes the value (via ADS)

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public void Write()
```

**Implements**  
IValue.Write [2233]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolException [2401]</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

DynamicValue Class [1869]

TwinCAT.TypeSystem Namespace [1622]

6.11.24.2.21 DynamicValue.WriteAsync Method

write as an asynchronous operation.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public Task<ResultWriteAccess> WriteAsync(
    CancellationToken cancel
)
```

**Parameters**

| cancel | Type: System.Threading.CancellationToken  

The cancellation token.

**Return Value**

Type: Task<ResultWriteAccess> [2575],  
Task<WriteValueResult>.

**Implements**

IValue.WriteAsync(CancellationToken) [2233]
6.11.24.2.22 DynamicValue.WriteMember Method

Writes the specified member element.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

protected virtual void WriteMember(
    ISymbol memberInstance,
    Object value
)

Parameters

memberInstance Type: TwinCAT.TypeSystem.ISymbol
The member instance.

value Type: System.Object
The value.

Reference

DynamicValue Class
TwinCAT.TypeSystem Namespace

6.11.25 DynamicVirtualStructInstance Class

Dynamic struct instance

Inheritance Hierarchy

System.Dynamic.DynamicObject
    TwinCAT.TypeSystem.DynamicSymbol
        TwinCAT.TypeSystem.DynamicStructInstance
            TwinCAT.TypeSystem.DynamicVirtualStructInstance

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public sealed class DynamicVirtualStructInstance : DynamicStructInstance,
    IVirtualStructInstance, IStructInstance, ISymbol, IAttributedInstance, IInstance, IBitSize

The DynamicVirtualStructInstance type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerSymbol</td>
<td>Gets the inner symbol of this <a href="#">DynamicSymbol</a> (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>AccessRights</td>
<td>Gets the access rights. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <a href="#">IDataType</a> in bits. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the category. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the <a href="#">Instance</a> (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the connection bound to this DynamicSymbol (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the <a href="#">IDataType</a> of the <a href="#">Instance</a>. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>HasRpcMethods</td>
<td>Gets a value indicating whether this instance has RPC methods (Inherited from DynamicStructInstance)</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this <a href="#">ISymbol</a> is persistent. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the <a href="#">Instance</a> represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the <a href="#">Instance</a> represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### Name | Description
--- | ---
MemberInstances | Gets the member instances of the `Struct Instance` | (Inherited from `DynamicStructInstance`)
NormalizedName | Gets the normalized instance name (fixed name for dynamic property access that does't contain invalid characters), (Inherited from `DynamicSymbol`)
NotificationSettings | Gets the notification settings. (Inherited from `DynamicSymbol`)
Parent | Gets the parent Symbol (Inherited from `DynamicSymbol`)
Size | Gets the size of the `IInstance` in bytes. (Inherited from `DynamicSymbol`)
SubSymbols | Gets the SubSymbols of the `ISymbol` (Inherited from `DynamicSymbol`)
TypeName | Gets the name of the `DataType` that is used for this `IInstance`. (Inherited from `DynamicSymbol`)
ValueEncoding | Gets the value encoding. (Inherited from `DynamicSymbol`)

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddMember</td>
<td>Adds an member instance.</td>
</tr>
<tr>
<td>Equals</td>
<td>Equals (Inherited from <code>DynamicSymbol</code>).</td>
</tr>
<tr>
<td>GetDynamicMembers</td>
<td>Returns the enumeration of all dynamic member names. (Inherited from <code>DynamicStructInstance</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Inherited from <code>DynamicSymbol</code>).</td>
</tr>
<tr>
<td>ReadAnyValue</td>
<td>Reads the value of this <code>Value</code> into a new created instance of the managed type (Inherited from <code>DynamicSymbol</code>).</td>
</tr>
<tr>
<td>ReadRawValue</td>
<td>Reads the raw value of the <code>IValueSymbol</code> (Ads Read / Write) (Inherited from <code>DynamicSymbol</code>).</td>
</tr>
<tr>
<td>ReadRawValueAsync</td>
<td>Reads the Symbols raw value (Inherited from <code>DynamicSymbol</code>).</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Read raw value as an asynchronous operation. (Inherited from <code>DynamicSymbol</code>).</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the <code>IValueSymbol</code> asynchronously. (Inherited from <code>DynamicSymbol</code>).</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents this instance. (Inherited from <code>DynamicSymbol</code>.)</td>
</tr>
</tbody>
</table>
TryGetMember

Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicStructInstance.)

TryReadValue

Reads the Value of the IValueSymbol (Inherited from DynamicSymbol.)

TrySetMember

Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Inherited from DynamicStructInstance.)

TryWriteValue

Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol.)

UpdateAnyValue

Reads the value of this Value into the specified managed value. (Inherited from DynamicSymbol.)

WriteAnyValue

Writes the value represented by the managed value to this Value (Inherited from DynamicSymbol.)

WriteRawValue(Byte).

Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)

WriteRawValue(Byte., Int32).

Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)

WriteRawValueAsync

Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from DynamicSymbol.)

WriteValue(Object)

Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol.)

WriteValue(Object, Int32)

Writes the specified value to the DynamicSymbol. (Inherited from DynamicSymbol.)

WriteValueAsync

Writes the Value of the IValueSymbol (Inherited from DynamicSymbol.)

Events

RawValueChanged

Occurs when the RawValue of the IValueSymbol has changed. (Inherited from DynamicSymbol.)

ValueChanged

Occurs when the (Primitive) value of the IValueSymbol has changed. (Inherited from DynamicSymbol.)

Reference

TwinCAT.TypeSystem Namespace

6.11.25.1 DynamicVirtualStructInstance Properties

The DynamicVirtualStructInstance type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerSymbol ![1798]</td>
<td>Gets the inner symbol of this DynamicSymbol ![1791] (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>AccessRights ![1798]</td>
<td>Gets the access rights. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>Attributes ![1799]</td>
<td>Gets the Symbol Attributes (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>BitSize ![1800]</td>
<td>Gets the size of the IDataType ![1986] in bits. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>ByteSize ![1800]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>Category ![1801]</td>
<td>Gets the category. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>Comment ![1801]</td>
<td>Gets the comment of the Instance ![2052] (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>Connection ![1802]</td>
<td>Gets the connection bound to this DynamicSymbol ![1791] (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>DataType ![1802]</td>
<td>Gets the IDataType ![1986] of the IInstance ![2052]. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>HasRpcMethods ![1785]</td>
<td>Gets a value indicating whether this instance has RPC methods (Inherited from DynamicStructInstance ![1779].)</td>
</tr>
<tr>
<td>HasValue ![1803]</td>
<td>Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>InstanceName ![1803]</td>
<td>Gets the name of the instance (without periods .) (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>InstancePath ![1804]</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>IsBitType ![1805]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>IsByteAligned ![1805]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>IsContainerType ![1806]</td>
<td>Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>IsPersistent ![1806]</td>
<td>Gets a value indicating whether this ISymbol ![2176] is persistent. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>IsPointer ![1807]</td>
<td>Indicates that the Instance ![2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>IsPrimitiveType ![1807]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>IsReadOnly ![1808]</td>
<td>Indicates that this instance is read only. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>IsRecursive ![1808]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>IsReference ![1809]</td>
<td>Indicates that the Instance ![2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol ![1791].)</td>
</tr>
<tr>
<td>IsStatic ![1809]</td>
<td>Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol ![1791].)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### TC1000

#### Version: 1.1

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemberInstances</td>
<td>Gets the member instances of the Struct Instance [1786]. (Inherited from DynamicStructInstance [1779].)</td>
</tr>
<tr>
<td>NormalizedName</td>
<td>Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>NotificationSettings</td>
<td>Gets the notification settings. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the IInstance [2052] in bytes. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>SubSymbols</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

### Reference

**DynamicVirtualStructInstance Class [1894]**

**TwinCAT.TypeSystem Namespace [1622]**

### 6.11.25.2 DynamicVirtualStructInstance Methods

The DynamicVirtualStructInstance [1894] type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddMember [1900]</td>
<td>Adds an member instance.</td>
</tr>
<tr>
<td>Equals [1816]</td>
<td>Equals (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>GetDynamicMemberNames [1789]</td>
<td>Returns the enumeration of all dynamic member names. (Inherited from DynamicStructInstance [1779].)</td>
</tr>
<tr>
<td>GetHashCode [1817]</td>
<td>Gets the HashCode of the Address (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadAnyValue</td>
<td>Reads the value of this Value [2254] into a new created instance of the managed type (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue</td>
<td>Reads the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValue(Int32) [1826]</td>
<td>Reads the Symbols raw value (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadRawValueAsync [1827]</td>
<td>Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadValue(Int32) [1831]</td>
<td>Reads the value of this DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ReadValueAsync [1832]</td>
<td>Reads the Value of the IValueSymbol [2254] asynchronously. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>ToString [1832]</td>
<td>Returns a String that represents this instance. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TryGetMember [1789]</td>
<td>Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicStructInstance [1779].)</td>
</tr>
<tr>
<td>TryReadValue [1834]</td>
<td>Reads the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>TrySetMember [1790]</td>
<td>Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Inherited from DynamicStructInstance [1779].)</td>
</tr>
<tr>
<td>TryWriteValue [1834]</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>UpdateAnyValue [1837]</td>
<td>Reads the value of this Value [2254] into the specified managed value. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteAnyValue [1838]</td>
<td>Writes the value represented by the managed value to this Value [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Bye.) [1839]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValue(Bye., Int32) [1839]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteRawValueAsyn (Bye., Int32) [1840]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write) (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValue(Object) [1841]</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValue(Object, Int32) [1844]</td>
<td>Writes the specified value to the DynamicSymbol [1791]. (Inherited from DynamicSymbol [1791].)</td>
</tr>
<tr>
<td>WriteValueAsync [1847]</td>
<td>Writes the Value of the IValueSymbol [2254] (Inherited from DynamicSymbol [1791].)</td>
</tr>
</tbody>
</table>

**Reference**

DynamicVirtualStructInstance Class [1894]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.25.2.1 DynamicVirtualStructInstance.AddMember Method

Adds an member instance.
TwinCAT.Ads Namespaces

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool AddMember(
    ISymbol memberInstance,
    IVirtualStructInstance parent
)
```

Parameters

- **memberInstance**
  
  Type: TwinCAT.TypeSystem.ISymbol
  
  The member instance.

- **parent**
  
  Type: TwinCAT.TypeSystem.IVirtualStructInstance
  
  The parent struct instance. Usually the this pointer.

Return Value

Type: **Boolean**

Implements

IVirtualStructInstance.AddMember(ISymbol, IVirtualStructInstance)

Reference

DynamicVirtualStructInstance Class

TwinCAT.TypeSystem Namespace

### 6.11.25.3 DynamicVirtualStructInstance Events

The DynamicVirtualStructInstance type exposes the following members.

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol has changed. (Inherited from DynamicSymbol.)</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol has changed. (Inherited from DynamicSymbol.)</td>
</tr>
</tbody>
</table>

Reference

DynamicVirtualStructInstance Class

TwinCAT.TypeSystem Namespace

### 6.11.26 EnumValue.T Class

Enum Value
**Inheritance Hierarchy**

```
System.Object
  TwinCAT.TypeSystem.EnumValue.T.
```

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public class EnumValue<T> : IEnumValue
where T : IConvertible
```

**Type Parameters**

- **T**
  
  Enum base type (byte,sbyte,short,ushort,int,uint,long or ulong)

The `EnumValue<T>` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagedBaseType</td>
<td>Gets the enumeration base type (sint,byte,short,ushort,int,uint,Int64,UInt64 supported)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Enum Value</td>
</tr>
<tr>
<td>Primitive</td>
<td>Gets the value.</td>
</tr>
<tr>
<td>RawValue</td>
<td>Gets the raw value of the enumeration (as byte array)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the Enum value (in bytes)</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse</td>
<td>Parse Enum Type string</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents this instance. (Overrides Object.ToString.)</td>
</tr>
<tr>
<td>TryParse</td>
<td>Parse EnumType string</td>
</tr>
</tbody>
</table>
6.11.26.1   EnumValue.T. Properties

The EnumValue.T. [1901] generic type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagedBaseType</td>
<td>Gets the enumeration base type (sint, byte, short, ushort, int, uint, Int64, UInt64 supported)</td>
</tr>
<tr>
<td>Name [1904]</td>
<td>Gets the name of the Enum Value</td>
</tr>
<tr>
<td>Primitive [1904]</td>
<td>Gets the value.</td>
</tr>
<tr>
<td>RawValue [1904]</td>
<td>Gets the raw value of the enumeration (as byte array)</td>
</tr>
<tr>
<td>Size [1905]</td>
<td>Gets the size of the Enum value (in bytes)</td>
</tr>
</tbody>
</table>

Reference

EnumValue.T..Class [1901]
TwinCAT.TypeSystem Namespace [1622]

6.11.26.1.1 EnumValue.T..ManagedBaseType Property

Gets the enumeration base type (sint, byte, short, ushort, int, uint, Int64, UInt64 supported)

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public Type ManagedBaseType { get; }

Property Value

Type: Type
The type of the base.

Implements

IEnumValue.ManagedBaseType [2029]

Reference

EnumValue.T..Class [1901]
TwinCAT.TypeSystem Namespace [1622]
6.11.26.1.2 EnumValue.T..Name Property

Gets the name of the Enum Value

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public string Name { get; }
```

**Property Value**

Type: `String`
The name.

**Implements**

[EnumValue.Name [2029]]

**Reference**

EnumValue.T. Class [1901]
TwinCAT.TypeSystem Namespace [1622]

6.11.26.1.3 EnumValue.T..Primitive Property

Gets the value.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public T Primitive { get; }
```

**Property Value**

Type: `T` [1901]
The value.

**Reference**

EnumValue.T. Class [1901]
TwinCAT.TypeSystem Namespace [1622]

6.11.26.1.4 EnumValue.T..RawValue Property

Gets the raw value of the enumeration (as byte array)
### EnumValue.T..RawValue Property

**Type:** Byte

The raw value.

**Implements**

IEnumValue.RawValue

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotSupportedException</td>
<td>Base type of enum is not allowed!</td>
</tr>
</tbody>
</table>

**Reference**

EnumValue.T. Class

TwinCAT.TypeSystem Namespace

### EnumValue.T..Size Property

**Gets the size of the Enum value (in bytes)**

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int Size { get; }
```

**Property Value**

Type: Int32

The size.

**Implements**

IEnumValue.Size

**Reference**

EnumValue.T. Class
TwinCAT.TypeSystem Namespace [1622]

6.11.26.2 EnumValue.T. Methods

The EnumValue.T. [1901] generic type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse [1906]</td>
<td>Parse Enum Type string</td>
</tr>
<tr>
<td>ToString [1907]</td>
<td>Returns a String that represents this instance. (Overrides Object.ToString.)</td>
</tr>
<tr>
<td>TryParse [1907]</td>
<td>Parse EnumType string</td>
</tr>
</tbody>
</table>

Reference

EnumValue.T. Class [1901]

TwinCAT.TypeSystem Namespace [1622]

6.11.26.2.1 EnumValue.T..Parse Method

Parse Enum Type string

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static EnumValue<T> Parse(
    IEnumType<T> type,
    string str
)
```

Parameters

| type | Type: TwinCAT.TypeSystem.IEnumType [2021], T [1901]. The type. |
| str  | Type: System.String The string. |
### Return Value

Type: `EnumValue<T>`.

### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormatException</td>
<td></td>
</tr>
</tbody>
</table>

### Reference

- `EnumValue<T>` Class [1901]
- `TwinCAT.TypeSystem` Namespace [1622]

#### 6.11.26.2.2 EnumValue<T>.ToString Method

Returns a `String` that represents this instance.

**Namespace:** `TwinCAT.TypeSystem` [1622]  
**Assembly:** `TwinCAT.Ads` (in `TwinCAT.Ads.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public override string ToString()
```

### Return Value

Type: `String`

A `String` that represents this instance.

### Reference

- `EnumValue<T>` Class [1901]
- `TwinCAT.TypeSystem` Namespace [1622]

#### 6.11.26.2.3 EnumValue<T>.TryParse Method

Parse EnumType string

**Namespace:** `TwinCAT.TypeSystem` [1622]  
**Assembly:** `TwinCAT.Ads` (in `TwinCAT.Ads.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static bool TryParse(
    IEnumType<T> type,
    string str,
    out EnumValue<T> value
)
```
**Parameters**

- **type**
  - Type: TwinCAT.TypeSystem.IEnumType.2021.T.1901
  - The type.

- **str**
  - Type: System.String
  - The string.

- **value**
  - Type: TwinCAT.TypeSystem.EnumValue.1901.T.1901
  - The value.

**Return Value**

- Type: Boolean
  - true if XXXX, false otherwise.

**Reference**

- EnumValue.T.Class.1901
- TwinCAT.TypeSystem Namespace.1622

**6.11.27 EnumValueCollection Class**

Class EnumValueCollection.

**Inheritance Hierarchy**

- System.Object
- TwinCAT.TypeSystem.EnumValueCollection

**Namespace:** TwinCAT.TypeSystem.1622

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0-Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public class EnumValueCollection : IList<IEnumValue>, ICollection<IEnumValue>, IEnumerable<IEnumValue>, IEnumerable, IEnumValueCollection, IEnumValueCollection<IEnumValue, IConvertible>
```

The EnumValueCollection type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count.1910</td>
<td>Gets the number of elements contained in the ICollection.T.</td>
</tr>
<tr>
<td>IsReadOnly.1910</td>
<td>Gets a value indicating whether the ICollection.T. is read-only.</td>
</tr>
<tr>
<td>Item.Int32.1911</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item.String.1912</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add.1914</td>
<td>Adds an item to the ICollection.T.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>Gets as read only.</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T.</td>
</tr>
<tr>
<td>Contains(Object)</td>
<td>Determines whether [contains] [the specified value].</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
<tr>
<td>Contains(IEnumValue)</td>
<td>Determines whether the ICollection.T. contains a specific value.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire list.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td>GetNames</td>
<td>Gets the names.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetValues</td>
<td>Gets the values.</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList.T.</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList.T. at the specified index.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse</td>
<td>Parses the specified string to the Enum value.</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection.T.</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IList.T. item at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetInfo</td>
<td>Tries the get information.</td>
</tr>
<tr>
<td>TryParse(String, IConvertible)</td>
<td>Parse the specified string to the enum value.</td>
</tr>
<tr>
<td>TryParse(String, IEnumValue)</td>
<td>Parse the specified string to the enum value.</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

### 6.11.27.1 EnumValueCollection Properties

The EnumValueCollection[1908] type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [1910]</td>
<td>Gets the number of elements contained in the ICollection.T.</td>
</tr>
<tr>
<td>IsReadOnly [1910]</td>
<td>Gets a value indicating whether the ICollection.T is read-only.</td>
</tr>
<tr>
<td>Item.Int32. [1911]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item.String. [1912]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
</tbody>
</table>

Reference

EnumValueCollection Class [1908]
TwinCAT.TypeSystem Namespace [1622]

6.11.27.1.1 EnumValueCollection.Count Property

Gets the number of elements contained in the ICollection.T.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public int Count { get; }

Property Value

Type: Int32
The count.

Implements
ICollection.T.Count

Reference

EnumValueCollection Class [1908]
TwinCAT.TypeSystem Namespace [1622]

6.11.27.1.2 EnumValueCollection.IsReadOnly Property

Gets a value indicating whether the ICollection.T is read-only.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public bool IsReadOnly { get; }
```

Property Value

Type: Boolean
true if this instance is read only; otherwise, false.

Implements

`ICollection<T>.IsReadOnly`

Reference

`EnumValueCollection Class [1908]`

`TwinCAT.TypeSystem Namespace [1622]`

### 6.11.27.1.3 EnumValueCollection.Item Property

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
</tbody>
</table>

Reference

`EnumValueCollection Class [1908]`

`TwinCAT.TypeSystem Namespace [1622]`

**EnumValueCollection.Item Property (Int32)**

Gets or sets the element at the specified index.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public IEnumValue this[int index] { get; set; }
```

**Parameters**

<table>
<thead>
<tr>
<th>index</th>
<th>Type: System.Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td>The index.</td>
<td></td>
</tr>
</tbody>
</table>
Return Value

Type: IEnumValue
EnumValue<T>.

Implements

IList<T>.Item.Int32.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

EnumValueCollection Class
Item Overload
TwinCAT.TypeSystem Namespace

EnumValueCollection.Item Property (String)

Gets or sets the element at the specified index.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public IConvertible this[string name] { get; }
```

Parameters

name Type: System.String
The name of the value

Return Value

Type: IConvertible
EnumValue<T>.

Implements

[EnumValueCollection.TEnumValue, TValue.Item.String]
Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

EnumValueCollection Class [› 1908]

Item Overload [› 1911]

TwinCAT.TypeSystem Namespace [› 1622]

6.11.27.2  EnumValueCollection Methods

The EnumValueCollection [› 1908] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add [› 1914]</td>
<td>Adds an item to the ICollection.T.</td>
</tr>
<tr>
<td>AsReadOnly [› 1915]</td>
<td>Gets as read only.</td>
</tr>
<tr>
<td>Clear [› 1915]</td>
<td>Removes all items from the ICollection.T.</td>
</tr>
<tr>
<td>Contains(Object) [› 1916]</td>
<td>Determines whether [contains] [the specified value].</td>
</tr>
<tr>
<td>Contains(String) [› 1916]</td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
<tr>
<td>Contains(IEnumValue) [› 1917]</td>
<td>Determines whether the ICollection.T. contains a specific value.</td>
</tr>
<tr>
<td>CopyTo [› 1918]</td>
<td>Copies the entire list.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator [› 1918]</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetNames [› 1919]</td>
<td>Gets the names.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetValues [› 1919]</td>
<td>Gets the values.</td>
</tr>
<tr>
<td>IndexOf [› 1920]</td>
<td>Determines the index of a specific item in the IList.T.</td>
</tr>
<tr>
<td>Insert [› 1921]</td>
<td>Inserts an item to the IList.T. at the specified index.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>----------</td>
<td>------------------</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Parse</td>
<td>Parses the specified string to the Enum value.</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the <code>IList&lt;T&gt;</code>.</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the <code>IList&lt;T&gt;</code> item at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>TryGetInfo</td>
<td>Tries the get information.</td>
</tr>
<tr>
<td>TryParse(String, <code>IConvertible</code>)</td>
<td>Parse the specified string to the enum value.</td>
</tr>
<tr>
<td>TryParse(String, <code>IEnumValue</code>)</td>
<td>Parse the specified string to the enum value.</td>
</tr>
</tbody>
</table>

**Reference**

`EnumValueCollection Class` [→ 1908]

`TwinCAT.TypeSystem Namespace` [→ 1622]

### 6.11.27.2.1 `EnumValueCollection.Add` Method

Adds an item to the `IList<T>`.

**Namespace:** `TwinCAT.TypeSystem` [→ 1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void Add(
    `IEnumValue` item
)
```

**Parameters**

- `item` Type: `TwinCAT.TypeSystem.IEnumValue` [→ 2028]
  
  The object to add to the `IList<T>`.

**Implements**

- `IList<T>.Add(T)`

**Reference**

- `EnumValueCollection Class` [→ 1908]

- `TwinCAT.TypeSystem Namespace` [→ 1622]
6.11.27.2.2  EnumValueCollection.AsReadOnly Method

Gets as read only.

**Namespace:**  TwinCAT.TypeSystem [1622]
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ReadOnlyEnumValueCollection AsReadOnly()
```

**Field Value**

Type:  ReadOnlyEnumValueCollection [2300]
As read only.

**Return Value**

Type:  ReadOnlyEnumValueCollection [2300]
ReadOnlyEnumValueCollection.

**Reference**

EnumValueCollection Class [1908]
TwinCAT.TypeSystem Namespace [1622]

6.11.27.2.3  EnumValueCollection.Clear Method

Removes all items from the ICollection.T.

**Namespace:**  TwinCAT.TypeSystem [1622]
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void Clear()
```

**Implements**

ICollection.T.Clear.

**Reference**

EnumValueCollection Class [1908]
TwinCAT.TypeSystem Namespace [1622]
## 6.11.27.2.4 EnumValueCollection.Contains Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(Object)</td>
<td>Determines whether [contains] [the specified value].</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
<tr>
<td>Contains(IEnumValue) [1916]</td>
<td>Determines whether the ICollection&lt;T&gt; contains a specific value.</td>
</tr>
</tbody>
</table>

### Reference

**EnumValueCollection Class** [1908]

**TwinCAT.TypeSystem Namespace** [1622]

### EnumValueCollection.Contains Method (Object)

Determines whether [contains] [the specified value].

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public bool Contains(
    Object value
)
```

### Parameters

- **value**
  - Type: `System.Object`
  - The value.

### Return Value

- Type: `Boolean`
  - true if [contains] [the specified value]; otherwise, false.

### Reference

**EnumValueCollection Class** [1908]

**Contains Overload** [1916]

**TwinCAT.TypeSystem Namespace** [1622]

### EnumValueCollection.Contains Method (String)

Determines whether [contains] [the specified name].
Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool Contains(
    string name
)
```

Parameters

name Type: System.String
The name.

Return Value

Type: Boolean
true if Contains (the specified name); otherwise, false.

Implements

IEnumeratorValueCollection.TEnumValue, TValue.Contains(String)

Reference

EnumValueCollection Class
Contains Overload
TwinCAT.TypeSystem Namespace

EnumValueCollection.Contains Method (IEnumeratorValue)

Determines whether the ICollection.T. contains a specific value.

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool Contains(
    IEnumeratorValue item
)
```

Parameters

item Type: TwinCAT.TypeSystem.IEnumValue
The object to locate in the ICollection.T.

Return Value

Type: Boolean
true if item is found in the ICollection.T.; otherwise, false.
6.11.27.2.5 EnumValueCollection.CopyTo Method

Copies the entire list.

**Namespace:** TwinCAT.TypeSystem [► 1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void CopyTo(IEnumValue[] array, int arrayIndex)
```

**Parameters**

- **array**
  - Type: .TwinCAT.TypeSystem.IEnumValue [► 2028],
  - The array.
- **arrayIndex**
  - Type: System.Int32
  - Index of the array.

**Reference**

- EnumValueCollection Class [► 1908]
- TwinCAT.TypeSystem Namespace [► 1622]

6.11.27.2.6 EnumValueCollection.GetEnumerator Method

Returns an enumerator that iterates through the collection.

**Namespace:** TwinCAT.TypeSystem [► 1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public IEnumerator<EnumValue> GetEnumerator()
```

Return Value

Type: `IEnumerator<EnumValue>` [2028].

A `IEnumerator<T>` that can be used to iterate through the collection.

Implements

`IEnumerator<T>.GetEnumerator`.

Reference

EnumValueCollection Class [1908]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.27.2.7 EnumValueCollection.GetNames Method

Gets the names.

Namespace: TwinCAT.TypeSystem [1622]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public string[] GetNames()
```

Return Value

Type: `System.String[]`.

Implements

`EnumValueCollection.TEnumValue, TValue..GetNames` [2037]

Reference

EnumValueCollection Class [1908]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.27.2.8 EnumValueCollection.GetValues Method

Gets the values.

Namespace: TwinCAT.TypeSystem [1622]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public IConvertible[] GetValues()
```

Return Value

Type: `IConvertible`, `T[]`.

Implements


Reference

`EnumValueCollection Class [6.11.27.2.9]`  
`TwinCAT.TypeSystem Namespace [6.11.27.2.9]`

6.11.27.2.9  EnumValueCollection.IndexOf Method

Determines the index of a specific item in the `IList<T>`.

**Namespace:** TwinCAT.TypeSystem [6.11.27.2.9]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb91b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int IndexOf(IEnumValue item)
```

Parameters

- `item`  
  Type: TwinCAT.TypeSystem.IEnumValue [6.11.27.2.9]  
  The object to locate in the `IList<T>`.

Return Value

Type: `Int32`  
The index of `item` if found in the list; otherwise, -1.

Implements

`IList<T>.IndexOf(T)`

Reference

`EnumValueCollection Class [6.11.27.2.9]`  
`TwinCAT.TypeSystem Namespace [6.11.27.2.9]`
6.11.27.2.10  EnumValueCollection.Insert Method

Inserts an item to the IList<T> at the specified index.

**Namespace:**  TwinCAT.TypeSystem [» 1622]

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public void Insert(
    int index,
    IEnumValue item
)
```

**Parameters**

- **index**
  - Type: System.Int32
  - The zero-based index at which item should be inserted.

- **item**
  - Type: TwinCAT.TypeSystem.IEnumValue [» 2028]
  - The object to insert into the IList<T>.

**Implements**

IList<T>.Insert(Int32, T)

**Reference**

EnumValueCollection Class [» 1908]

TwinCAT.TypeSystem Namespace [» 1622]

---

6.11.27.2.11  EnumValueCollection.Parse Method

Parses the specified string to the Enum value.

**Namespace:**  TwinCAT.TypeSystem [» 1622]

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public IConvertible Parse(
    string name
)
```

**Parameters**

- **name**
  - Type: System.String
  - The name.

**Return Value**

Type: IConvertible

T.
TwinCAT.Ads Namespaces

**Implements**

| implements | TwinCAT.Ads.Namespaces.IEnumValueCollection.TEnumValue, TwinCAT.Ads.TValue.Parse(String) |

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td>name</td>
</tr>
</tbody>
</table>

**Reference**

<table>
<thead>
<tr>
<th>Reference</th>
<th>[1908]</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnumValueCollection Class</td>
<td>[1908]</td>
</tr>
<tr>
<td>TwinCAT.TypeSystem Namespace</td>
<td>[1622]</td>
</tr>
</tbody>
</table>

---

**6.11.27.2.12 EnumValueCollection.Remove Method**

Removes the first occurrence of a specific object from the ICollection.T.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# code snippet

```csharp
public bool Remove(IEnumValue item)
```

**Parameters**

<table>
<thead>
<tr>
<th>item</th>
<th>Type: TwinCAT.TypeSystem.IEnumValue [2028]</th>
</tr>
</thead>
<tbody>
<tr>
<td>item</td>
<td>The object to remove from the ICollection.T.</td>
</tr>
</tbody>
</table>

**Return Value**

Type: Boolean

ture if item was successfully removed from the ICollection.T; otherwise, false. This method also returns false if item is not found in the original ICollection.T.

**Implements**

| ICollection.T.Remove(T) |

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

---

1922 Version: 1.1 TC1000
6.11.27.2.13 EnumValueCollection.RemoveAt Method

Removes the IList<T> item at the specified index.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public void RemoveAt(int index)

Parameters

index Type: System.Int32
The zero-based index of the item to remove.

Implements

IList<T>.RemoveAt(Int32)

Reference

EnumValueCollection Class [1908]
TwinCAT.TypeSystem Namespace [1622]

6.11.27.2.14 EnumValueCollection.TryGetInfo Method

Tries the get information.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public bool TryGetInfo(Object val, out IEnumValue ei)

Parameters

val Type: System.Object
The value.
ei
Type: TwinCAT.TypeSystem.IEnumValue
The ei.

Return Value
Type: Boolean
true if XXXX, false otherwise.

Reference
EnumValueCollection Class [1908]
TwinCAT.TypeSystem Namespace [1622]

6.11.27.2.15 EnumValueCollection.TryParse Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="1924" alt="TryParse(String, IConvertible.)" /></td>
<td>Parse the specified string to the enum value.</td>
</tr>
<tr>
<td><img src="1925" alt="TryParse(String, IEnumValue.)" /></td>
<td>Parse the specified string to the enum value.</td>
</tr>
</tbody>
</table>

Reference
EnumValueCollection Class [1908]
TwinCAT.TypeSystem Namespace [1622]

EnumValueCollection.TryParse Method (String, IConvertible.)
Parse the specified string to the enum value.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool TryParse(
    string name,
    out IConvertible value
)
```

Parameters

| name | Type: System.String
|      | The name.
| value| Type: System.IConvertible.
|      | The value.
Return Value
Type: Boolean
true if XXXX, false otherwise.

Implements
IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue.) [2039]

Reference
EnumValueCollection Class [1908]
TryParse Overload [1924]
TwinCAT.TypeSystem Namespace [1622]

EnumValueCollection.TryParse Method (String, IEnumValue.)
Parse the specified string to the enum value.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
public bool TryParse(
    string name,
    out TwinCAT.TypeSystem.IEnumValue value
)

Parameters
name Type: System.String
The name.
value Type: TwinCAT.TypeSystem.IEnumValue [2028]
The value.

Return Value
Type: Boolean
true if XXXX, false otherwise.

Implements
IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue.) [2039]

Reference
EnumValueCollection Class [1908]
TryParse Overload [1924]
TwinCAT.TypeSystem Namespace [1622]
6.11.28   EnumValueCollection<T>. Class

Collection of EnumValues [1901]

Inheritance Hierarchy

System.Object
   TwinCAT.TypeSystem.EnumValueCollection<T>

Namespace:  TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public class EnumValueCollection<T> : IList<EnumValue<T>>,
   ICollection<EnumValue<T>>, IEnumerble<EnumValue<T>>, IEnumerable,
   IEnumValueCollection<EnumValue<T>, T>
where T : IConvertible

Type Parameters

T    Base type of enum

The EnumValueCollection<T> type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection&lt;T&gt; is read-only.</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the enumeration value str from the string representation.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>Gets as read only.</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether [contains] [the specified value].</td>
</tr>
<tr>
<td>Contains(EnumValue&lt;T&gt;)</td>
<td>Determines whether the ICollection&lt;T&gt; contains a specific value.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire list.</td>
</tr>
<tr>
<td>Empty</td>
<td>Return an Empty Collection.</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### Reference

TwinCAT.TypeSystem Namespace [» 1622]

### 6.11.28.1 EnumValueCollection.T. Properties

The `EnumValueCollection.T` generic type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [1928]</td>
<td>Gets the number of elements contained in the ICollection.T.</td>
</tr>
<tr>
<td>IsReadOnly [1928]</td>
<td>Gets a value indicating whether the ICollection.T is read-only.</td>
</tr>
<tr>
<td>Item.Int32 [1929]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item.String [1930]</td>
<td>Gets the enumeration value str from the string representation.</td>
</tr>
</tbody>
</table>

Reference

EnumValueCollection.T..Class [1926]
TwinCAT.TypeSystem Namespace [1622]

6.11.28.1.1 EnumValueCollection.T..Count Property

Gets the number of elements contained in the ICollection.T.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int Count { get; }
```

Property Value

Type: Int32
The count.

Implements

ICollection.T.Count

Reference

EnumValueCollection.T..Class [1926]
TwinCAT.TypeSystem Namespace [1622]

6.11.28.1.2 EnumValueCollection.T..IsReadOnly Property

Gets a value indicating whether the ICollection.T is read-only.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#
```csharp
public bool IsReadOnly { get; }
```

Property Value

Type: Boolean
true if this instance is read only; otherwise, false.

Implements

ICollection<T>.IsReadOnly

Reference

EnumValueCollection<T>.Class [1926]
TwinCAT.TypeSystem Namespace [1622]

6.11.28.1.3 EnumValueCollection<T>.Item Property

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32.1929</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item.String.1930</td>
<td>Gets the enumeration value str from the string representation.</td>
</tr>
</tbody>
</table>

Reference

EnumValueCollection<T>.Class [1926]
TwinCAT.TypeSystem Namespace [1622]

EnumValueCollection<T>.Item Property (Int32)

Gets or sets the element at the specified index.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
```csharp
public EnumValue<T> this[int index] { get; set; }
```

Parameters

index Type: System.Int32
The index.
Return Value

Type: EnumValue[T].

Implements

IList<T>.Item.Int32.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

EnumValueCollection<T>.Class

Item Overload

TwinCAT.TypeSystem Namespace

EnumValueCollection<T>.Item Property (String)

Gets the enumeration value str from the string representation.

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public T this[string str] { get; }
```

Parameters

str  
Type: System.String
The string.

Return Value

Type: T

Implements

|EnumValueCollection<TEnumValue, TValue>.Item.String|
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

## Reference

- EnumValueCollection.T.Class [1926]
- Item Overload [1929]
- TwinCAT.TypeSystem Namespace [1622]

### 6.11.28.2 EnumValueCollection.T. Methods

The EnumValueCollection.T. [1926] generic type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add [1932]</td>
<td>Adds an item to the ICollection.T.</td>
</tr>
<tr>
<td>AsReadOnly [1933]</td>
<td>Gets as read only.</td>
</tr>
<tr>
<td>Clear [1933]</td>
<td>Removes all items from the ICollection.T.</td>
</tr>
<tr>
<td>Contains(String) [1934]</td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
<tr>
<td>Contains(T) [1935]</td>
<td>Determines whether [contains] [the specified value].</td>
</tr>
<tr>
<td>Contains(EnumValue.T.) [1935]</td>
<td>Determines whether the ICollection.T contains a specific value.</td>
</tr>
<tr>
<td>CopyTo [1936]</td>
<td>Copies the entire list.</td>
</tr>
<tr>
<td>Empty [1936]</td>
<td>Return an Empty Collection.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator [1937]</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetValues [1938]</td>
<td>Gets the values.</td>
</tr>
<tr>
<td>IndexOf [1938]</td>
<td>Determines the index of a specific item in the IList.T.</td>
</tr>
<tr>
<td>Insert [1939]</td>
<td>Inserts an item to the IList.T. at the specified index.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Parse</td>
<td>Parses the specified string to the Enum value.</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the <code>ICollection&lt;T&gt;</code>.</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the <code>IList&lt;T&gt;</code> item at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>TryGetInfo</td>
<td>Tries the get information.</td>
</tr>
<tr>
<td>TryParse(String, EnumValue&lt;T&gt;)</td>
<td>Parse the specified string to the enum value.</td>
</tr>
<tr>
<td>TryParse(String, T)</td>
<td>Parse the specified string to the enum value.</td>
</tr>
</tbody>
</table>

**Reference**

*EnumValueCollection<T>. Class*  

*TwinCAT.TypeSystem Namespace*

#### 6.11.28.2.1 EnumValueCollection<T>. Add Method

Adds an item to the `ICollection<T>`.

**Namespace:** TwinCAT.TypeSystem  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll)  
**Version:** 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public void Add(
    EnumValue<T> item
)
```

**Parameters**

*item*  
Type: TwinCAT.TypeSystem.EnumValue<T>  
The object to add to the `ICollection<T>`.

**Implements**

`ICollection<T>.Add(T)`

**Reference**

*EnumValueCollection<T>. Class*  

*TwinCAT.TypeSystem Namespace*
6.11.28.2.2  EnumValueCollection<T>.AsReadOnly Method

Gets as read only.

**Namespace:** TwinCAT.TypeSystem
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ReadOnlyEnumValueCollection<T> AsReadOnly()
```

**Field Value**

Type: `ReadOnlyEnumValueCollection<T>`

As read only.

**Return Value**

Type: `ReadOnlyEnumValueCollection<T>`

**Reference**

EnumValueCollection<T> Class

TwinCAT.TypeSystem Namespace

6.11.28.2.3  EnumValueCollection<T>.Clear Method

Removes all items from the `ICollection<T>`.

**Namespace:** TwinCAT.TypeSystem
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void Clear()
```

**Implements**

`ICollection<T>.Clear`

**Reference**

EnumValueCollection<T> Class

TwinCAT.TypeSystem Namespace
6.11.28.2.4 EnumValueCollection.T..Contains Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String)</td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether [contains] [the specified value].</td>
</tr>
<tr>
<td>Contains(EnumValue.T.)</td>
<td>Determines whether the ICollection.T contains a specific value.</td>
</tr>
</tbody>
</table>

Reference

EnumValueCollection.T..Class [1926]
TwinCAT.TypeSystem Namespace [1622]

EnumValueCollection.T..Contains Method (String)
Determines whether [contains] [the specified name].

Namespace:  TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd6ca3e72bc0ea15da1c14

Syntax

C#
public bool Contains(
    string name
)

Parameters

name      Type: System.String
          The name.

Return Value

Type: Boolean
type [contains] [the specified name]; otherwise, false.

Implements

IEnumerable<TEnumValue, TValue>.Contains(String) [2036]

Reference

EnumValueCollection.T..Class [1926]
Contains Overload [1934]
TwinCAT.TypeSystem Namespace [1622]
EnumValueCollection.T..Contains Method (T)

Determines whether [contains] [the specified value].

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool Contains(
    T value
)
```

Parameters

- value
  - Type: T
  - The value.

Return Value

- Type: Boolean
  - true if [contains] [the specified value]; otherwise, false.

Reference

EnumValueCollection.T..Contains Method (EnumValue.T.)

Determines whether the ICollection{T} contains a specific value.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool Contains(
    EnumValue<T> item
)
```

Parameters

- item
  - Type: TwinCAT.TypeSystem.EnumValue
  - The object to locate in the ICollection{T}.

Return Value

- Type: Boolean
  - true if item is found in the ICollection{T}; otherwise, false.
### EnumValueCollection.T..CopyTo Method

Copies the entire list.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public void CopyTo(
    EnumValue<T>[] array,  
    int arrayIndex  
)
```

**Parameters**

- **array**  
  Type: .TwinCAT.TypeSystem.EnumValue
  The array.

- **arrayIndex**  
  Type: System.Int32
  Index of the array.

**Implements**

- ICollection.T..CopyTo(T, Int32)

**Reference**

- EnumValueCollection.T..Class
- TwinCAT.TypeSystem Namespace
Syntax

C#

```csharp
public static EnumValueCollection<T> Empty()
```

Return Value

Type: `EnumValueCollection<T>`

Reference

`EnumValueCollection<T>.Class`

**6.11.28.2.7 EnumValueCollection<T>.GetEnumerator Method**

Returns an enumerator that iterates through the collection.

**Namespace:** `TwinCAT.TypeSystem`

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public IEnumerator<EnumValue<T>> GetEnumerator()
```

Return Value

Type: `IEnumerator<EnumValueCollection<T>>`

A `IEnumerator<T>` that can be used to iterate through the collection.

**Implements**

`IEnumerable<T>.GetEnumerator`

Reference

`EnumValueCollection<T>.Class`

**6.11.28.2.8 EnumValueCollection<T>.GetNames Method**

Gets the names.

**Namespace:** `TwinCAT.TypeSystem`

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public string[] GetNames()
```
Return Value

Type: `System.String[]`.

Implements

`[EnumValueCollection.TEnumValue, TValue..GetNames.]` [2037]

Reference

`EnumValueCollection.T. Class` [1926]

`TwinCAT.TypeSystem Namespace` [1622]

### 6.11.28.2.9 EnumValueCollection.T..GetValues Method

Gets the values.

**Namespace:** `TwinCAT.TypeSystem` [1622]

**Assembly:** `TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`

**Syntax**

C#

```
public T[] GetValues()
```

Return Value

Type: `T` [1926],

`T[]`.

Implements

`[EnumValueCollection.TEnumValue, TValue..GetValues.]` [2037]

Reference

`EnumValueCollection.T. Class` [1926]

`TwinCAT.TypeSystem Namespace` [1622]

### 6.11.28.2.10 EnumValueCollection.T..IndexOf Method

Determines the index of a specific item in the `IList<T>`.

**Namespace:** `TwinCAT.TypeSystem` [1622]

**Assembly:** `TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`
Syntax

C#

```csharp
public int IndexOf(
    EnumValue<T> item
)
```

Parameters

item Type: `TwinCAT.TypeSystem.EnumValue`<T>`[1901].T`<T>`[1926].` The object to locate in the `IList<T>`. 

Return Value

Type: `Int32` 
The index of item if found in the list; otherwise, -1.

Implements

`IList<T>.IndexOf(T)`

Reference

`EnumValueCollection<T>.Class`<T>`[1926]`
`TwinCAT.TypeSystem.Namespace`<T>`[1622]`

6.11.28.2.11 EnumValueCollection<T>.Insert Method

Inserts an item to the `IList<T>` at the specified index.

Namespace: `TwinCAT.TypeSystem`<T>`[1622]`

Assembly: `TwinCAT.Ads` (in `TwinCAT.Ads.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void Insert(
    int index,
    EnumValue<T> item
)
```

Parameters

index Type: `System.Int32` The zero-based index at which item should be inserted.

item Type: `TwinCAT.TypeSystem.EnumValue`<T>`[1901].T`<T>`[1926].` The object to insert into the `IList<T>`.

Implements

`IList<T>.Insert(Int32, T)`
### 6.11.28.2.12 EnumValueCollection.T..Parse Method

Parses the specified string to the Enum value.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

```csharp
public T Parse(
    string name
)
```

#### Parameters

<table>
<thead>
<tr>
<th>name</th>
<th>Type: System.String</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The name.</td>
</tr>
</tbody>
</table>

#### Return Value

Type: T

T.

#### Implements

IEnumValueCollection.TEnumValue, TValue..Parse(String)

#### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td>name</td>
</tr>
</tbody>
</table>

### 6.11.28.2.13 EnumValueCollection.T..Remove Method

Removes the first occurrence of a specific object from the ICollection.T..

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public bool Remove(
    EnumValue<T> item
)
```

Parameters

**item**

Type: `TwinCAT.TypeSystem.EnumValue`<br>
The object to remove from the `ICollection<T>`.

Return Value

Type: `Boolean`
true if item was successfully removed from the `ICollection<T>`; otherwise, false. This method also returns false if item is not found in the original `ICollection<T>`.

Implements

`ICollection<T>.Remove(T)`

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

`EnumValueCollection.T.Class`<br>`TwinCAT.TypeSystem Namespace`

6.11.28.2.14 `EnumValueCollection.T..RemoveAt Method`

Removes the `IList<T>.Item` at the specified index.

**Namespace:** `TwinCAT.TypeSystem`<br>**Assembly:** `TwinCAT.Ads (in TwinCAT.Ads.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void RemoveAt(
    int index
)
```

Parameters

**index**

Type: `System.Int32`
The zero-based index of the item to remove.
TwinCAT.Ads Namespaces

Implements
IList<T>.RemoveAt(Int32)

Reference
EnumValueCollection.T.Class [► 1926]
TwinCAT.TypeSystem Namespace [► 1622]

6.11.28.2.15 EnumValueCollection.T..TryGetInfo Method
Tries the get information.

Namespace: TwinCAT.TypeSystem [► 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool TryGetInfo(
    T val,
    out EnumValue<T> ei
)
```

Parameters

val
Type: T [► 1926]
The value.

ei
Type: TwinCAT.TypeSystem.EnumValue [► 1901].
The ei.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference
EnumValueCollection.T.Class [► 1926]
TwinCAT.TypeSystem Namespace [► 1622]

6.11.28.2.16 EnumValueCollection.T..TryParse Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryParse(String, EnumValue&lt;T&gt;) [► 1943]</td>
<td>Parse the specified string to the enum value.</td>
</tr>
<tr>
<td>TryParse(String, T) [► 1943]</td>
<td>Parse the specified string to the enum value.</td>
</tr>
</tbody>
</table>
EnumValueCollection<T>.TryParse Method (String, EnumValue<T>)

Parse the specified string to the enum value.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool TryParse(
    string name,
    out EnumValue<T> value
)
```

**Parameters**

- **name**
  Type: System.String
  The name.

- **value**
  Type: TwinCAT.TypeSystem.EnumValue
  The value.

**Return Value**

Type: Boolean
true if XXXX, false otherwise.

**Implements**

IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue.)

**Reference**

EnumValueCollection.T..Class

TryParse Overload

TwinCAT.TypeSystem Namespace
Syntax

C#

```csharp
public bool TryParse(
    string name,
    out T value
)
```

Parameters

- `name` Type: `System.String`  
The name.
- `value` Type: `T`  
The value.

Return Value

Type: `Boolean`  
true if XXXX, false otherwise.

Implements

- `EnumValueCollection.TEnumValue, TValue.TryParse(String, TValue)`  

Reference

- `EnumValueCollection.T. Class`  
- `TryParse Overload`  
- `TwinCAT.TypeSystem Namespace`

6.11.28.3 EnumValueCollection.T. Type Conversions

The `EnumValueCollection.T. Class` generic type exposes the following members.

Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code> EnumValueCollection.T. to EnumValueCollection</code></td>
<td>Performs an explicit conversion from <code>EnumValueCollection.T. Class</code> to <code>EnumValueCollection</code>.</td>
</tr>
</tbody>
</table>

Reference

- `EnumValueCollection.T. Class`  
- `TwinCAT.TypeSystem Namespace`

6.11.28.3.1 EnumValueCollection.T. Conversion (EnumValueCollection.T to EnumValueCollection)

Performs an explicit conversion from `EnumValueCollection.T. Class` to `EnumValueCollection`. 
**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static explicit operator EnumValueCollection (EnumValueCollection<T> coll)
```

**Parameters**

- **coll**
  - Type: TwinCAT.TypeSystem.EnumValueCollection
  - T

**Return Value**

- Type: EnumValueCollection
  - The result of the conversion.

**Reference**

- EnumValueCollection<T> Class
- TwinCAT.TypeSystem Namespace

### 6.11.29 FieldCollection Class

Collection of IField objects.

**Inheritance Hierarchy**

- System.Object
  - TwinCAT.TypeSystem.Generic.InstanceCollection
  - TwinCAT.TypeSystem.IField
  - TwinCAT.TypeSystem.FieldCollection

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public class FieldCollection : InstanceCollection<IField>,
    IFieldCollection, IInstanceCollection<IField>, ICollection<IField>, IEnumerable<IField>, IEnumerable
```

The FieldCollection type exposes the following members.

**Constructors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FieldCollection</td>
<td>Initializes a new instance of the FieldCollection class.</td>
</tr>
</tbody>
</table>
### FieldCollection(IEnumerable<Field>)
Initializes a new instance of the FieldCollection class (copy constructor)

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the collection count. (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>InnerList</td>
<td>Gets the List of instances. (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>InnerPathDict</td>
<td>The Path dictionary (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether this instance is read only. (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the IInstance at the specified index. (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the IInstance with the specified instance path. (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>Mode</td>
<td>The mode this InstanceCollection&lt;T&gt; is working in. (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds the specified item. (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>AddRange</td>
<td>Adds the specified items to this collection. (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>Returns a read only copy of this collection (shallow copy)</td>
</tr>
<tr>
<td>Clear</td>
<td>Clears this instance. (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>Clone</td>
<td>Clones this FieldCollection.</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether this collection contains an IInstance with the specified InstanceName / InstancePath (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether this collection contains the specified IInstance (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>ContainsName</td>
<td>Determines whether the specified instance name contains name. (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies this InstanceCollection&lt;T&gt; to the specified array. (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Gets the enumerator. (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInstance [1947]</td>
<td>Gets the IInstance [2052] by instance path. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>GetInstanceByName</td>
<td>Gets the name of the instance by. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf [2476]</td>
<td>Determines the index of the specified IInstance [2052]. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Insert [2477]</td>
<td>Inserts the specified IInstance [2052] at the specified index. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove [2478]</td>
<td>Removes the specified item. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>RemoveAt [2479]</td>
<td>Removes the IInstance [2052] at the specified index. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetInstance</td>
<td>Tries to get the IInstance [2052] of the specified path. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>TryGetInstanceByName</td>
<td>Tries to get Instances by name. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>TryGetMember [1951]</td>
<td>Tries to get the specified member</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.29.1 FieldCollection Constructor

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FieldCollection</td>
<td>Initializes a new instance of the FieldCollection [1945] class.</td>
</tr>
<tr>
<td>FieldCollection(IEnumerable.IField)</td>
<td>Initializes a new instance of the FieldCollection [1945] class (copy constructor)</td>
</tr>
</tbody>
</table>

**Reference**

FieldCollection Class [1945]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.29.1.1 FieldCollection Constructor

Initializes a new instance of the FieldCollection [1945] class.
**FieldCollection Constructor (IEnumerable.IField.)**

Initializes a new instance of the `FieldCollection` class (copy constructor)

- **Namespace:** TwinCAT.TypeSystem
- **Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public FieldCollection()
```

### Parameters

- `coll`  
  - Type: `System.Collections.Generic.IEnumerable<IField>`
  - The coll.

### Reference

- FieldCollection Class [1945]
- FieldCollection Overload [1947]
- TwinCAT.TypeSystem Namespace [1945]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InnerPathDict</td>
<td>The Path dictionary (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether this instance is read only. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Item.Int32.</td>
<td>Gets or sets the IInstance [2052] at the specified index. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Item.String.</td>
<td>Gets the IInstance [2052] with the specified instance path. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Mode</td>
<td>The mode this InstanceCollection.T. [2460] is working in. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
</tbody>
</table>

**Reference**

FieldCollection Class [1945]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.29.3 FieldCollection Methods

The FieldCollection [1945] type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add [2469]</td>
<td>Adds the specified item. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>AddRange [2470]</td>
<td>Adds the specified items to this collection. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>AsReadOnly [1950]</td>
<td>Returns a read only copy of this collection (shallow copy)</td>
</tr>
<tr>
<td>Clear [2471]</td>
<td>Clears this instance. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Clone [1951]</td>
<td>Clones this FieldCollection [1945].</td>
</tr>
<tr>
<td>Contains(String) [2472]</td>
<td>Determines whether this collection contains an IInstance [2052] with the specified InstanceName / InstancePath (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Contains(T) [2472]</td>
<td>Determines whether this collection contains the specified IInstance [2052] (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>ContainsName [2473]</td>
<td>Determines whether the specified instance name contains name. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>CopyTo [2474]</td>
<td>Copies this InstanceCollection.T. [2460] to the specified array. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator [2474]</td>
<td>Gets the enumerator. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInstance [2475]</td>
<td>Gets the IInstance [2052] by instance path. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>GetInstanceByName [2476]</td>
<td>Gets the name of the instance by. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf [2476]</td>
<td>Determines the index of the specified IInstance [2052]. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>Insert [2477]</td>
<td>Inserts the specified IInstance [2052] at the specified index. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove [2478]</td>
<td>Removes the specified item. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>RemoveAt [2479]</td>
<td>Removes the IInstance [2052] at the specified index. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetInstance [2479]</td>
<td>Tries to get the IInstance [2052] of the specified path. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>TryGetInstanceByName [2480]</td>
<td>Tries to get Instances by name. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>TryGetMember [1951]</td>
<td>Tries to get the specified member</td>
</tr>
</tbody>
</table>

Reference

FieldCollection Class [1945]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.29.3.1 FieldCollection.AsReadOnly Method

Returns a read only copy of this collection (shallow copy)

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b609593ffdc3e72bc0ea15da1c14

**Syntax**

```csharp
public ReadonlyFieldCollection AsReadOnly()
```

**Return Value**

Type: ReadonlyFieldCollection [2317]

The readonly copy.

Reference

FieldCollection Class [1945]

TwinCAT.TypeSystem Namespace [1622]
6.11.29.3.2 FieldCollection.Clone Method

Clones this FieldCollection.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public FieldCollection Clone()

Return Value

Type: FieldCollection
A cloned FieldCollection.

Reference

FieldCollection Class
TwinCAT.TypeSystem Namespace

6.11.29.3.3 FieldCollection.TryGetMember Method

Tries to get the specified member

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool TryGetMember(
    string fieldName,
    out TwinCAT.TypeSystem.IField symbol
)

Parameters

fieldName Type: System.String
Name of the member.
symbol Type: TwinCAT.TypeSystem.IField
The symbol.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

TwinCAT.TypeSystem.IFieldCollection.TryGetMember(String, IField)
6.11.30  IAliasInstance Interface

Interface representing an instance of an IAliasType.

Namespace:  TwinCAT.TypeSystem
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdc3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IAliasInstance : ISymbol, IAttributedInstance, IInstance, IBitSize
```

The IAliasInstance type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the IInstance (Inherited from IInstance.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType of the IInstance. (Inherited from IInstance.)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods .) (Inherited from IInstance.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from IInstance.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a fullDataType but instead of some sort of bit mapping (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this ISymbol is persistent. (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the IInstance represents a Pointer type (Pointer TO) (Inherited from IInstance.)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that this instance is read only. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984]</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052]. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]

TwinCAT.TypeSystem.ISymbol [2176]

#### 6.11.30.1 IAliasInstance Properties

The IAliasInstance [1952] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the Instance [2052] (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the Instance [2052]. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods .) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
</tbody>
</table>
### IAliasType Interface

Interface representing an Alias Type

**Namespace:** TwinCAT.TypeSystem [►1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public interface IAliasType : IDataType, IBitSize
```

The IAliasType type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1987]</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>BaseType [1956]</td>
<td>Gets the Base Type</td>
</tr>
<tr>
<td>BaseType Name [1957]</td>
<td>Gets the BaseType name</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [1988]</td>
<td>Gets the Data Type category (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Comment [1988]</td>
<td>Gets the comment behind the variable declaration. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>FullName [1989]</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Id [1989]</td>
<td>Gets the ID of the DataType (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a container type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPointer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a pointer type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPrimitive [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is primitive (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsReference [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is a reference type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Name [1992]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Namespace [1992]</td>
<td>Gets the namespace string within the IDataType [1986] exists. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]

6.11.31.1 IAliasType Properties

The IAliasType [1954] type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1987]</td>
<td>Gets the attributes of the [IDataType] (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>BaseType [1956]</td>
<td>Gets the Base Type</td>
</tr>
<tr>
<td>BaseType Name [1957]</td>
<td>Gets the BaseType name</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the [IDataType] in bits. (Inherited from [IBitSize].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from [IBitSize].)</td>
</tr>
<tr>
<td>Category [1988]</td>
<td>Gets the Data Type category (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>Comment [1988]</td>
<td>Gets the comment behind the variable declaration. (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>FullName [1989]</td>
<td>Gets the full name of the [IDataType] (Namespace + Name) (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>Id [1989]</td>
<td>Gets the ID of the Data Type (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full Data Type but instead of some sort of bit mapping (Inherited from [IBitSize].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from [IBitSize].)</td>
</tr>
<tr>
<td>IsContainer [1990]</td>
<td>Gets a value indicating whether this [IDataType] is a container type (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>IsPointer [1990]</td>
<td>Gets a value indicating whether this [IDataType] is a pointer type (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>IsPrimitive [1991]</td>
<td>Gets a value indicating whether this [IDataType] is primitive (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>IsReference [1991]</td>
<td>Gets a value indicating whether this [IDataType] is a reference type (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>Name [1992]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>Namespace [1992]</td>
<td>Gets the namespace string within the [IDataType] exists. (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from [BitSize].)</td>
</tr>
</tbody>
</table>

### Reference

- IAliasType Interface [1954]
- TwinCAT.TypeSystem Namespace [1622]

### 6.11.31.1.1 IAliasType.BaseType Property

Gets the Base Type
Namespace:  TwinCAT.TypeSystem [1622]
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
IDataType BaseType { get; }

Property Value

Type: IDataType [1986]
The alias base type or null if not resolved.

Reference

IAliasType Interface [1954]
TwinCAT.TypeSystem Namespace [1622]

6.11.31.1.2 IAliasType.BaseTypeName Property

Gets the BaseType name

Namespace:  TwinCAT.TypeSystem [1622]
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
string BaseTypeName { get; }

Property Value

Type: String

Reference

IAliasType Interface [1954]
TwinCAT.TypeSystem Namespace [1622]

6.11.32 IAnyTypeMarshaler Interface

Interface IAnyTypeMarshaler Implements the IGenericTypeMarshaler [2046]

Namespace:  TwinCAT.TypeSystem [1622]
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public interface IAnyTypeMarshaler : IGenericTypeMarshaler,
ITypeMarshaler
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CanMarshal(Object)</strong> [2217]</td>
<td>Determines whether ADS can marshal the specified value (Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td><strong>CanMarshal(Type)</strong> [2218]</td>
<td>Determines whether ADS can marshal the specified managed data type. (Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td><strong>CanMarshal(Object, Int32.)</strong> [1960]</td>
<td>Determines whether ADS can marshal the specified managed data type.</td>
</tr>
<tr>
<td><strong>CanMarshal(Type, Int32.)</strong> [1960]</td>
<td>Determines whether ADS can marshal the specified managed data type.</td>
</tr>
<tr>
<td><strong>Marshal(Object, Encoding, Span)</strong> [2219]</td>
<td>(Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td><strong>Marshal(Object, Int32., Encoding, Span)</strong> [1961]</td>
<td></td>
</tr>
<tr>
<td><strong>MarshalSize(Object, Encoding)</strong> [2219]</td>
<td>Gets the byte size of the value when marshalled. (Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td><strong>MarshalSize(Object, Int32., Encoding)</strong> [1962]</td>
<td>Gets the byte size of the value when marshalled.</td>
</tr>
<tr>
<td><strong>Unmarshal(Type, ReadOnlySpan, Void, Byte)</strong> [2220]</td>
<td>(Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td><strong>Unmarshal(Type, Int32., ReadOnlySpan, Void, Byte)</strong> [1964]</td>
<td></td>
</tr>
<tr>
<td><strong>Unmarshal.T. (ReadOnlySpan, Void, Byte)</strong> [2048]</td>
<td>(Inherited from IGenericTypeMarshaler [2046].)</td>
</tr>
</tbody>
</table>

## Remarks

The IAnyTypeMarshaler supports value marshalling / Unmarshalling with the ADS ANY_TYPE concept, what means that the specified managed type is supported by an arguments metadata array (args parameter).

## Reference

- TwinCAT.TypeSystem Namespace [1622]
- TwinCAT.TypeSystem.IGenericTypeMarshaler [2046]
6.11.32.1  IAnyTypeMarshaler Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanMarshal(Object)</td>
<td>Determines whether ADS can marshal the specified value (Inherited from ITypeMarshaler.)</td>
</tr>
<tr>
<td>CanMarshal(Types)</td>
<td>Determines whether ADS can marshal the specified managed data type.</td>
</tr>
<tr>
<td>CanMarshal(Object, .Int32.)</td>
<td>Determines whether ADS can marshal the specified managed data type.</td>
</tr>
<tr>
<td>Marshal(Object, Encoding, Span)</td>
<td>Gets the byte size of the value when marshalled. (Inherited from ITypeMarshaler.)</td>
</tr>
<tr>
<td>MarshalSize(Object, Encoding)</td>
<td>Gets the byte size of the value when marshalled. (Inherited from ITypeMarshaler.)</td>
</tr>
<tr>
<td>Unmarshal(Types, ReadOnlySpan, Void, Byte)</td>
<td>(Inherited from ITypeMarshaler.)</td>
</tr>
<tr>
<td>Unmarshal(T, ReadOnlySpan, Void, Byte)</td>
<td>(Inherited from IGenericTypeMarshaler.)</td>
</tr>
</tbody>
</table>

Reference

IAnyTypeMarshaler Interface [1957]

TwinCAT.TypeSystem Namespace [1622]

6.11.32.1.1  IAnyTypeMarshaler.CanMarshal Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanMarshal(Object)</td>
<td>Determines whether ADS can marshal the specified value (Inherited from ITypeMarshaler.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>CanMarshal(Type)</td>
<td>Determines whether ADS can marshal the specified managed data type. (Inherited from ITypeMarshaler.)</td>
</tr>
<tr>
<td>CanMarshal(Object, Int32.)</td>
<td>Determines whether ADS can marshal the specified managed data type.</td>
</tr>
<tr>
<td>CanMarshal(Type, Int32.)</td>
<td>Determines whether ADS can marshal the specified managed data type.</td>
</tr>
</tbody>
</table>

Reference
---

IAnyTypeMarshaler Interface [1957]

TwinCAT.TypeSystem Namespace [1622]

IAnyTypeMarshaler.CanMarshal Method (Object, Int32.)

Determines whether ADS can marshal the specified managed data type.

**Namespace**: TwinCAT.TypeSystem [1622]

**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0e15da1c14

**Syntax**

C#

```csharp
bool CanMarshal(
    Object value,
    int[] args
)
```

**Parameters**

- **value**
  - Type: System.Object
  - The managed value.

- **args**
  - Type: System.Int32.
  - The arguments.

**Return Value**

Type: Boolean

true if this instance can marshal the specified managed type; otherwise, false.

Reference
---

IAnyTypeMarshaler Interface [1957]

CanMarshal Overload [1959]

TwinCAT.TypeSystem Namespace [1622]

IAnyTypeMarshaler.CanMarshal Method (Type, Int32.)

Determines whether ADS can marshal the specified managed data type.
Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
bool CanMarshal(
    Type anyType,
    int[] args
)
```

Parameters

- `anyType` Type: System.Type
  The Managed data type.
- `args` Type: System.Int32
  The arguments.

Return Value

Type: Boolean
true if this instance can marshal the specified managed type; otherwise, false.

Reference

IAnyTypeMarshaler Interface [1957]
CanMarshal Overload [1959]
TwinCAT.TypeSystem Namespace [1622]

6.11.32.1.2 IAnyTypeMarshaler.Marshal Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Inherited Symbol]</td>
<td>Marshal(Object, Encoding, Span) [2219]</td>
</tr>
<tr>
<td>![Inherited Symbol]</td>
<td>Marshal(Object, Int32, Encoding, Span) [1961]</td>
</tr>
</tbody>
</table>

Reference

IAnyTypeMarshaler Interface [1957]
TwinCAT.TypeSystem Namespace [1622]

IAnyTypeMarshaler.Marshal Method (Object, .Int32., Encoding, Span`1)

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
int Marshal(
    Object anyValue,
    int[] args,
    Encoding encoding,
    Span destination
)
```

**Parameters**

- **anyValue**
  - Type: `System.Object`

- **args**
  - Type: `System.Int32`.

- **encoding**
  - Type: `System.Text.Encoding`

- **destination**
  - Type: `Span`

**Return Value**

Type: `Int32`

**Reference**

- `IAnyTypeMarshaler Interface [1957]`
- `Marshal Overload [1961]`
- `TwinCAT.TypeSystem Namespace [1622]`

### 6.11.32.1.3 `IAnyTypeMarshaler.MarshalSize Method`

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MarshalSize(Object, Encoding) [2219]</td>
<td>Gets the byte size of the value when marshalled. (Inherited from <code>ITypeMarshaler [2216].</code>)</td>
</tr>
<tr>
<td>MarshalSize(Object, Int32, Encoding) [1962]</td>
<td>Gets the byte size of the value when marshalled.</td>
</tr>
</tbody>
</table>

**Reference**

- `IAnyTypeMarshaler Interface [1957]`
- `TwinCAT.TypeSystem Namespace [1622]`

**IAnyTypeMarshaler.MarshalSize Method (Object, Int32, Encoding)**

Gets the byte size of the value when marshalled.
**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
int MarshalSize(
    Object anyValue,
    int[] args,
    Encoding encoding
)
```

### Parameters

- **anyValue**
  - Type: `System.Object`
  - Any value.
- **args**
  - Type: `System.Int32`
  - The arguments.
- **encoding**
  - Type: `System.Text.Encoding`
  - The encoding.

### Return Value

Type: `Int32`

The marshal size of the value.

### Reference

- `IAnyTypeMarshaler Interface` [1957]
- `MarshalSize Overload` [1962]
- `TwinCAT.TypeSystem Namespace` [1622]

### 6.11.32.1.4 `IAnyTypeMarshaler.Unmarshal Method`

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Unmarshal" /> .T. (ReadOnlySpan, Void, Byte)</td>
<td>(Inherited from <code>IGenericTypeMarshaler</code> [2046].)</td>
</tr>
<tr>
<td><img src="image" alt="Unmarshal" /> (Type, ReadOnlySpan, Void, Byte)</td>
<td>(Inherited from <code>ITypeMarshaler</code> [2216].)</td>
</tr>
<tr>
<td><img src="image" alt="Unmarshal" /> (Type, Int32, ReadOnlySpan, Void, Byte)</td>
<td><img src="image" alt="Unmarshal" /> .T. (ReadOnlySpan, Void, Byte)</td>
</tr>
</tbody>
</table>

### Reference

- `IAnyTypeMarshaler Interface` [1957]
- `TwinCAT.TypeSystem Namespace` [1622]
IAnyTypeMarshaler.Unmarshal Method (Type, .Int32., ReadOnlySpan`1, Void, Byte)

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```c#
int Unmarshal(
    Type anyType,
    int[] args,
    ReadOnlySpan source,
    void encoding,
    byte value
)
```

Parameters

- anyType Type: System.Type
- args Type: System.Int32
- source Type: ReadOnlySpan
- encoding Type: System.Void
- value Type: System.Byte

Return Value

Type: Int32

Reference

IAnyTypeMarshaler Interface [1957]
Unmarshal Overload [1963]
TwinCAT.TypeSystem Namespace [1622]

6.11.33 IArrayInstance Interface

Interface representing an array instance

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```c#
public interface IArrayInstance : ISymbol,
    IAttributedInstance, IInstance, IBitSize
```

The IArrayInstance type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Symbol/Datatype Category. (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the Instance. (Inherited from Instance.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType of the Instance. (Inherited from Instance.)</td>
</tr>
<tr>
<td>Dimensions</td>
<td>Gets the dimensions as read only collection.</td>
</tr>
<tr>
<td>Elements</td>
<td>Gets the contained Array Elements as read only collection.</td>
</tr>
<tr>
<td>ElementType</td>
<td>Gets the type of the contained elements.</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods .) (Inherited from Instance.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from Instance.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this ISymbol is persistent. (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the Instance represents a Pointer type (Pointer TO) (Inherited from Instance.)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the Instance represents a Reference type (REFERENCE TO) (Inherited from Instance.)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this Instance is static. (Inherited from Instance.)</td>
</tr>
<tr>
<td>Item</td>
<td>Gets the ISymbol with the specified indices.</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol. (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType (Inherited from IBitSize.)</td>
</tr>
</tbody>
</table>
### IArrayInstance Properties

The `IArrayInstance` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from <code>IAtributedInstance</code>.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>IDataType</code> in bits. (Inherited from <code>IBitSize</code>.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from <code>IBitSize</code>.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Symbol/Datatype Category (Inherited from <code>ISymbol</code>).</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the <code>IInstance</code> (Inherited from <code>IInstance</code>).</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the <code>IDataType</code> of the <code>IInstance</code>. (Inherited from <code>IInstance</code>).</td>
</tr>
<tr>
<td>Dimensions</td>
<td>Gets the dimensions as read only collection.</td>
</tr>
<tr>
<td>Elements</td>
<td>Gets the contained Array Elements as read only collection.</td>
</tr>
<tr>
<td>ElementType</td>
<td>Gets the type of the contained elements.</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.) (Inherited from <code>IInstance</code>).</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from <code>IInstance</code>).</td>
</tr>
</tbody>
</table>

---

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryGetElement([IList, Int32, ISymbol][1970])</td>
<td>Tries to get the array element with the specified indices (jagged array support).</td>
</tr>
<tr>
<td>TryGetElement([Int32, ISymbol][1971])</td>
<td>Tries to get the array element with specified indices (only first level on jagged arrays)</td>
</tr>
</tbody>
</table>

---

### Reference

TwinCAT.TypeSystem Namespace [1622]

**6.11.33.1**  
**IArrayInstance Properties**

The `IArrayInstance` type exposes the following members.
## TwinCAT.Ads Namespaces

### IArrayInstance.Dimensions Property

Gets the dimensions as read only collection.

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
IDimensionCollection Dimensions { get; }
```
6.11.33.1.2 IArrayInstance.Elements Property

Gets the contained Array Elements as read only collection.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp  
ISymbolCollection<ISymbol> Elements { get; }  
```

**Property Value**

Type: ISymbolCollection [2185] | Symbol [2176].
The elements.

**Reference**

IArrayInstance Interface [1964]
TwinCAT.TypeSystem Namespace [1622]

6.11.33.1.3 IArrayInstance.ElementType Property

Gets the type of the contained elements.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp  
IDataType ElementType { get; }  
```

**Property Value**

Type: IDataType [1986]
The type of the element.

**Reference**

IArrayInstance Interface [1964]
TwinCAT.TypeSystem Namespace [1622]
6.11.33.1.4  IArrayInstance.Item Property

Gets the `ISymbol` with the specified indices.

**Namespace:**  TwinCAT.TypeSystem

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
ISymbol this[int[] indices] { get; }
```

**Parameters**

- `indices`  
  Type: `System.Int32`.  
  The indices.

**Property Value**

Type: `ISymbol`  
The `ISymbol`.

**Return Value**

Type: `ISymbol`  

**Reference**

IArrayInstance Interface

TwinCAT.TypeSystem Namespace

---

6.11.33.2  IArrayInstance Methods

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryGetElement(List&lt;Int32&gt;, ISymbol)</td>
<td>Tries to get the array element with the specified indices (jagged array support).</td>
</tr>
<tr>
<td>TryGetElement(Int32, ISymbol)</td>
<td>Tries to get the array element with specified indices (only first level on jagged arrays)</td>
</tr>
</tbody>
</table>

**Reference**

IArrayInstance Interface

TwinCAT.TypeSystem Namespace
6.11.33.2.1  IArrayInstance.TryGetElement Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>➤ TryGetElement(IList&lt;Int32.., ISymbol.&gt; [1970])</td>
<td>Tries to get the array element with the specified indices (jagged array support).</td>
</tr>
<tr>
<td>➤ TryGetElement(Int32.., ISymbol.) [1971]</td>
<td>Tries to get the array element with specified indices (only first level on jagged arrays).</td>
</tr>
</tbody>
</table>

Reference

| IArrayInstance Interface [1964] |
| IArrayInstance.TryGetElement Method (IList..Int32.., ISymbol.) | Tries to get the array element with the specified indices (jagged array support). |

Namespace:  TwinCAT.TypeSystem [1622]  Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
bool TryGetElement(IList<int[]> jaggedIndices, out ISymbol symbol)
```

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>jaggedIndices</td>
<td>System.Collections.Generic.IList..Int32.. The jagged indices list.</td>
</tr>
<tr>
<td>symbol</td>
<td>TwinCAT.TypeSystem.ISymbol [2176]. The symbol.</td>
</tr>
</tbody>
</table>

Return Value

Type:  Boolean  true if found, false if the jagged indices specifiers is out-of-range.

Reference

| IArrayInstance Interface [1964] |
| TryGetElement Overload [1970] |
| TwinCAT.TypeSystem Namespace [1622] |
IArrayInstance.TryGetElement Method (.Int32., ISymbol.)

Tries to get the array element with specified indices (only first level on jagged arrays)

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
bool TryGetElement(
    int[] indices,
    out ISymbol symbol
)
```

Parameters

- **indices**
  - Type: System.Int32
  - The indices.

- **symbol**
  - Type: TwinCAT.TypeSystem.ISymbol
  - The found Array element symbol (out-parameter).

Return Value

Type: Boolean
true if found, false if the indices specifiers is out-of-range.

Reference

IArrayInstance Interface
TryGetElement Overload
TwinCAT.TypeSystem Namespace

6.11.34 IArrayType Interface

Interface representing an array DataType.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IArrayType : IDataType, IBitSize
```

The IArrayType type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType (Inherited from IDataType)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [1988]</td>
<td>Gets the Data Type category (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Comment [1988]</td>
<td>Gets the comment behind the variable declaration. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Dimensions [1974]</td>
<td>Gets the dimensions as read only collection.</td>
</tr>
<tr>
<td>ElementType [1974]</td>
<td>Gets the type of the contained elements.</td>
</tr>
<tr>
<td>ElementTypeName [1974]</td>
<td>Gets the name of the element datatype.</td>
</tr>
<tr>
<td>FullName [1989]</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Id [1989]</td>
<td>Gets the ID of the Data Type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full Data Type but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a container type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsJagged [1975]</td>
<td>Gets a value indicating whether this instance is jagged.</td>
</tr>
<tr>
<td>IsPointer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a pointer type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPrimitive [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is primitive (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsReference [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is a reference type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>JaggedLevel [1975]</td>
<td>Gets the jagged level (Non-Jagged Arrays have level 1)</td>
</tr>
<tr>
<td>Name [1992]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Namespace [1992]</td>
<td>Gets the namespace string within the IDataType [1986] exists. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

### 6.11.34.1 IArrayProperty Members

The IArrayType [1971] type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the <code>IDataType</code> [1986] (Inherited from <code>IDataType</code> [1986].)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>IDataType</code> [1986] in bits. (Inherited from <code>IBitSize</code> [1982].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from <code>IBitSize</code> [1982].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from <code>IDataType</code> [1986].)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment behind the variable declaration. (Inherited from <code>IDataType</code> [1986].)</td>
</tr>
<tr>
<td>Dimensions</td>
<td>Gets the dimensions as read only collection.</td>
</tr>
<tr>
<td>ElementType</td>
<td>Gets the type of the contained elements.</td>
</tr>
<tr>
<td>ElementTypeName</td>
<td>Gets the name of the element datatype.</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the <code>IDataType</code> [1986] (Namespace + Name) (Inherited from <code>IDataType</code> [1986].)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the DataType (Inherited from <code>IDataType</code> [1986].)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from <code>IBitSize</code> [1982].)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <code>IBitSize</code> [1982].)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this <code>IDataType</code> [1986] is a container type (Inherited from <code>IDataType</code> [1986].)</td>
</tr>
<tr>
<td>IsJagged</td>
<td>Gets a value indicating whether this instance is jagged.</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this <code>IDataType</code> [1986] is a pointer type (Inherited from <code>IDataType</code> [1986].)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this <code>IDataType</code> [1986] is primitive (Inherited from <code>IDataType</code> [1986].)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this <code>IDataType</code> [1986] is a reference type (Inherited from <code>IDataType</code> [1986].)</td>
</tr>
<tr>
<td>JaggedLevel</td>
<td>Gets the jagged level (Non-Jagged Arrays have level 1)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from <code>IDataType</code> [1986].)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the <code>IDataType</code> [1986] exists. (Inherited from <code>IDataType</code> [1986].)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on <code>IsBitType</code> [1984] (Inherited from <code>IBitSize</code> [1982].)</td>
</tr>
</tbody>
</table>

### Reference

- `IArrayType Interface` [1971]
- `TwinCAT.TypeSystem Namespace` [1622]
6.11.34.1.1  IArrayType.Dimensions Property

Gets the dimensions as read only collection.

Namespace:  TwinCAT.TypeSystem [1622]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

IDimensionCollection Dimensions { get; }

Property Value

Type:  IDimensionCollection [2000]
The dimensions.

Reference

IArrayType Interface [1971]
TwinCAT.TypeSystem Namespace [1622]

6.11.34.1.2  IArrayType.ElementType Property

Gets the type of the contained elements.

Namespace:  TwinCAT.TypeSystem [1622]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

IDataType ElementType { get; }

Property Value

Type:  IDataType [1986]
The type of the element or null if not resolved.

Reference

IArrayType Interface [1971]
TwinCAT.TypeSystem Namespace [1622]

6.11.34.1.3  IArrayType.ElementTypeName Property

Gets the name of the element datatype.

Namespace:  TwinCAT.TypeSystem [1622]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
### 6.11.34.1.4 IArrayType.IsJagged Property

Gets a value indicating whether this instance is jagged.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
bool IsJagged { get; }
```

#### Property Value

**Type:** Boolean

true if this instance is jagged; otherwise, false.

**Reference**

IArrayType Interface

TwinCAT.TypeSystem Namespace

### 6.11.34.1.5 IArrayType.JaggedLevel Property

Gets the jagged level (Non-Jagged Arrays have level 1)

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
int JaggedLevel { get; }
```

#### Property Value

**Type:** Int32

The jagged level.
6.11.35  IArrayValue Interface

Interface IArrayValue

Namespace:  TwinCAT.TypeSystem [1622]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IArrayValue : IValue
```

The IArrayValue type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Gets the age of the value (last successful read of the value) (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>CachedRaw</td>
<td>Gets the cached Raw internal Data. (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the data type bound to this IValue [2226] (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IValue [2226] is a primitive value. (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol bound to this IValue [2226]. (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Gets the Time stamp of the last successful read of the Value (local user time, UTC) (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>UpdateMode</td>
<td>Gets the update mode (not implemented yet) (Inherited from IValue [2226].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Reads the value (via ADS) (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>ReadAsync</td>
<td>Reads the value (via ADS) (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>ResolveValue</td>
<td>Resolves the Value object to its primitive value. (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>TryGetArrayElement Values</td>
<td>Returns Array Element values.</td>
</tr>
<tr>
<td>TryGetIndexValue</td>
<td>Tries to get the specified Array Element</td>
</tr>
<tr>
<td>TryResolveValue</td>
<td>Tries to resolves the Value object to its primitive value. (Inherited from IValue [2226].)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrySetIndexValue</td>
<td>Tries to set the indexed value on Arrays</td>
</tr>
<tr>
<td>Write</td>
<td>Writes the value (via ADS) (Inherited from IValue)</td>
</tr>
<tr>
<td>WriteAsync</td>
<td>Writes the value (via ADS) (Inherited from IValue)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace

6.11.35.1 IArrayValue Properties

The IArrayValue type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Gets the age of the value (last successful read of the value)</td>
</tr>
<tr>
<td>CachedRaw</td>
<td>Gets the cached Raw internal Data.</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the data type bound to this IValue</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IValue is a primitive value.</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol bound to this IValue</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Gets the Time stamp of the last successful read of the Value</td>
</tr>
<tr>
<td>UpdateMode</td>
<td>Gets the update mode (not implemented yet)</td>
</tr>
</tbody>
</table>

### Reference

IArrayValue Interface

6.11.35.2 IArrayValue Methods

The IArrayValue type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Reads the value (via ADS) (Inherited from IValue)</td>
</tr>
<tr>
<td>ReadAsync</td>
<td>Reads the value (via ADS) (Inherited from IValue)</td>
</tr>
<tr>
<td>ResolveValue</td>
<td>Resolves the Value object to its primitive value. (Inherited from IValue)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TryGetArrayElementValues</td>
<td>Returns Array Element values.</td>
</tr>
<tr>
<td>TryGetIndexValue</td>
<td>Tries to get the specified Array Element</td>
</tr>
<tr>
<td>TryResolveValue</td>
<td>Tries to resolves the Value object to its primitive value. (Inherited from IValue.)</td>
</tr>
<tr>
<td>TrySetIndexValue</td>
<td>Tries to set the indexed value on Arrays</td>
</tr>
<tr>
<td>Write</td>
<td>Writes the value (via ADS) (Inherited from IValue.)</td>
</tr>
<tr>
<td>WriteAsync</td>
<td>Writes the value (via ADS) (Inherited from IValue.)</td>
</tr>
</tbody>
</table>

**Reference**

IArrayValue Interface

TwinCAT.TypeSystem Namespace

### 6.11.35.2.1 IArrayValue.TryGetArrayElementValues Method

**Description:** Returns Array Element values.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
bool TryGetArrayElementValues(
    out IEnumerable<Object> elementValues
)
```

**Parameters**

- **elementValues**
  - The element values.

**Return Value**

- Type: Boolean
  - true if XXXX, false otherwise.

**Reference**

IArrayValue Interface

TwinCAT.TypeSystem Namespace

### 6.11.35.2.2 IArrayValue.TryGetIndexValue Method

**Description:** Tries to get the specified Array Element
**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
bool TryGetIndexValue(
    int[] indices,
    out Object value
)
```

**Parameters**

- **indices**
  - Type: `System.Int32`.
  - The indices.

- **value**
  - Type: `System.Object`.
  - The value.

**Return Value**

- Type: `Boolean`
  - `true` if XXXX, `false` otherwise.

### Reference

- IArrayValue Interface [1976]
- TwinCAT.TypeSystem Namespace [1622]

### 6.11.35.2.3 IArrayValue.TrySetIndexValue Method

Tries to set the indexed value on Arrays

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
bool TrySetIndexValue(
    Object[] indexes,
    Object value
)
```

**Parameters**

- **indexes**
  - Type: `System.Object`.
  - The indexes.

- **value**
  - Type: `System.Object`.
  - The value.

**Return Value**

- Type: `Boolean`
  - `true` if succeeded, `false` otherwise.
6.11.36  IAttributedInstance Interface

Interface IAttributedInstance

Namespace:  TwinCAT.TypeSystem
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public interface IAttributedInstance : IInstance, IBitSize

The IAttributedInstance type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes.</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the IInstance. (Inherited from IInstance.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType of the IInstance. (Inherited from IInstance.)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods .) (Inherited from IInstance.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from IInstance.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the IInstance represents a Pointer type (Pointer TO) (Inherited from IInstance.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the IInstance represents a Reference type (REFERENCE TO) (Inherited from IInstance.)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this IInstance is static. (Inherited from IInstance.)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType (Inherited from IBitSize.)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TypeName</strong> [1986]</td>
<td>Gets the name of the <strong>DataType</strong> that is used for this <strong>IInstance</strong>.</td>
</tr>
<tr>
<td><strong>ValueEncoding</strong> [1982]</td>
<td>Gets the value encoding.</td>
</tr>
</tbody>
</table>

### Reference

**TwinCAT.TypeSystem Namespace** [1.622]

**TwinCAT.TypeSystem.IInstance** [1.2052]

### 6.11.36.1 **IAtributedInstance** Properties

The **IAtributedInstance** [1.1980] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attributes</strong> [1982]</td>
<td>Gets the Type Attributes.</td>
</tr>
<tr>
<td><strong>BitSize</strong> [1984]</td>
<td>Gets the size of the <strong>IDataType</strong> in bits. (Inherited from <strong>IBitSize</strong> [1982].)</td>
</tr>
<tr>
<td><strong>ByteSize</strong> [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from <strong>IBitSize</strong> [1982].)</td>
</tr>
<tr>
<td><strong>Comment</strong> [2053]</td>
<td>Gets the comment of the <strong>Instance</strong> (Inherited from <strong>IInstance</strong> [2052].)</td>
</tr>
<tr>
<td><strong>DataType</strong> [2054]</td>
<td>Gets the <strong>IDataType</strong> of the <strong>Instance</strong>. (Inherited from <strong>IInstance</strong> [2052].)</td>
</tr>
<tr>
<td><strong>InstanceName</strong> [2054]</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from <strong>IInstance</strong> [2052].)</td>
</tr>
<tr>
<td><strong>InstancePath</strong> [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from <strong>IInstance</strong> [2052].)</td>
</tr>
<tr>
<td><strong>IsBitType</strong> [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full <strong>DataType</strong> but instead of some sort of bit mapping (Inherited from <strong>IBitSize</strong> [1982].)</td>
</tr>
<tr>
<td><strong>IsByteAligned</strong> [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <strong>IByteAligned</strong> [1982].)</td>
</tr>
<tr>
<td><strong>IsPointer</strong> [2055]</td>
<td>Indicates that the <strong>Instance</strong> represents a Pointer type (Pointer TO) (Inherited from <strong>IPointer</strong> [2052].)</td>
</tr>
<tr>
<td><strong>IsReference</strong> [2056]</td>
<td>Indicates that the <strong>Instance</strong> represents a Reference type (REFERENCE TO) (Inherited from <strong>IReference</strong> [2052].)</td>
</tr>
<tr>
<td><strong>IsStatic</strong> [2056]</td>
<td>Gets a value indicating whether this <strong>Instance</strong> is static. (Inherited from <strong>IStatic</strong> [2052].)</td>
</tr>
<tr>
<td><strong>Size</strong> [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on <strong>IsBitType</strong> [1984] (Inherited from <strong>ISize</strong> [1982].)</td>
</tr>
<tr>
<td><strong>TypeName</strong> [2056]</td>
<td>Gets the name of the <strong>DataType</strong> that is used for this <strong>Instance</strong>.</td>
</tr>
<tr>
<td><strong>ValueEncoding</strong> [1982]</td>
<td>Gets the value encoding.</td>
</tr>
</tbody>
</table>
6.11.36.1.1  IAttributedInstance.Attributes Property

Gets the Type Attributes.

**Namespace:**  TwinCAT.TypeSystem

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
ITypeAttributeCollection Attributes { get; }
```

**Property Value**

Type:  ITypeAttributeCollection

The attributes.

**Reference**

IAttributedInstance Interface
TwinCAT.TypeSystem Namespace

6.11.36.1.2  IAttributedInstance.ValueEncoding Property

Gets the value encoding.

**Namespace:**  TwinCAT.TypeSystem

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
Encoding ValueEncoding { get; }
```

**Property Value**

Type:  Encoding

The value encoding.

**Reference**

IAttributedInstance Interface
TwinCAT.TypeSystem Namespace

6.11.37  IBitSize Interface

Interface IBitSize
Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

The IBitSize type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits.</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType</td>
</tr>
</tbody>
</table>

Remarks

Specifies the Bitness of the the object and the bit resp. byte sizes.

Reference

TwinCAT.TypeSystem Namespace

6.11.37.1 IBitSize Properties

The IBitSize type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits.</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType</td>
</tr>
</tbody>
</table>

Reference

IBitSize Interface

TwinCAT.TypeSystem Namespace
6.11.37.1.1 IBitSize.BitSize Property

Gets the size of the [DataType] in bits.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
int BitSize { get; }
```

Property Value

Type: Int32
The size of [DataType] in bits.

Reference

IBitSize Interface
TwinCAT.TypeSystem Namespace

6.11.37.1.2 IBitSize.ByteSize Property

Gets the (aligned) size of the Type/Instance in Bytes

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
int ByteSize { get; }
```

Property Value

Type: Int32
The size of the byte.

Reference

IBitSize Interface
TwinCAT.TypeSystem Namespace

6.11.37.1.3 IBitSize.IsBitType Property

Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#
```csharp
bool IsBitType { get; }
```

Property Value

Type: Boolean
true if this instance is bit mapping; otherwise, false.

Reference

IBitSize Interface [1982]
TwinCAT.TypeSystem Namespace [1622]

6.11.37.1.4 IBitSize.IsByteAligned Property

Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
```csharp
bool IsByteAligned { get; }
```

Property Value

Type: Boolean
true if this instance is byte aligned; otherwise, false.

Reference

IBitSize Interface [1982]
TwinCAT.TypeSystem Namespace [1622]

6.11.37.1.5 IBitSize.Size Property

Gets the size of the object in bytes or Bits dependant on IsBitType [1984]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
```csharp
int Size { get; }
```

Property Value

Type: Int32
The size of the bit.
Reference

IBitSize Interface [1982]

TwinCAT.TypeSystem Namespace [1622]

6.11.38 IDataType Interface

Base interface for objects representing data types

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public interface IDataType : IBitSize

The IQ>DataType type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Bytesize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment behind the variable declaration.</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the IDataType (Namespace + Name)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the Data Type</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataTy</td>
</tr>
</tbody>
</table>
6.11.38.1  IDataType Properties

The IDataType [1986] type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1987]</td>
<td>Gets the attributes of the IDataType [1986]</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [1988]</td>
<td>Gets the Data Type category</td>
</tr>
<tr>
<td>Comment [1988]</td>
<td>Gets the comment behind the variable declaration.</td>
</tr>
<tr>
<td>FullName [1989]</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name)</td>
</tr>
<tr>
<td>Id [1989]</td>
<td>Gets the ID of the DataType</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full Data Type but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a container type</td>
</tr>
<tr>
<td>IsPointer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a pointer type</td>
</tr>
<tr>
<td>IsPrimitive [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is primitive</td>
</tr>
<tr>
<td>IsReference [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is a reference type</td>
</tr>
<tr>
<td>Name [1992]</td>
<td>Gets the name of the Data Type (without namespace)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
</tbody>
</table>

Reference

IDataType Interface [1986]

TwinCAT.TypeSystem Namespace [1622]

6.11.38.1.1  IDataType.Attributes Property

Gets the attributes of the IDataType [1986]
Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

ITypeAttributeCollection Attributes { get; }

Property Value

Type: ITypeAttributeCollection [2211]
The attributes.

Reference

IDataType Interface [1986]
TwinCAT.TypeSystem Namespace [1622]

6.11.38.1.2 IDataType.Category Property

Gets the Data Type category

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

DataTypeCategory Category { get; }

Property Value

Type: DataTypeCategory [1649]
The category.

Reference

IDataType Interface [1986]
TwinCAT.TypeSystem Namespace [1622]

6.11.38.1.3 IDataType.Comment Property

Gets the comment behind the variable declaration.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

string Comment { get; }
6.11.38.1.4  **IDataType.FullName Property**

Gets the full name of the [IDataType](#) (Namespace + Name)

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
string FullName { get; }
```

**Property Value**

Type: **String**

The full name.

**Reference**

[IDataType Interface](#) [1986]

TwinCAT.TypeSystem Namespace [1622]

6.11.38.1.5  **IDataType.Id Property**

Gets the ID of the DataType

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
int Id { get; }
```

**Property Value**

Type: **Int32**

The id.

**Reference**

[IDataType Interface](#) [1986]

TwinCAT.TypeSystem Namespace [1622]
6.11.38.1.6  IDatatype.IsContainer Property

Gets a value indicating whether this IDatatype [1986] is a container type

**Namespace:**  TwinCAT.TypeSystem [1622]

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
bool IsContainer { get; }
```

**Property Value**

Type:  Boolean

ture if this instance is container type; otherwise, false.

**Remarks**

Container Types are all types that contain SubElements like

- Array [1649]
- Pointer [1649]
- Union [1649]
- Struct [1649]
- Function [1649]
- FunctionBlock [1649]
- Program [1649]

and the Alias [1649] and Reference [1649] types, if they have a container type as base type.

**Reference**

IDatatype Interface [1986]

TwinCAT.TypeSystem Namespace [1622]

IDatatype.Category [1988]

6.11.38.1.7  IDatatype.IsPointer Property

Gets a value indicating whether this IDatatype [1986] is a pointer type

**Namespace:**  TwinCAT.TypeSystem [1622]

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
bool IsPointer { get; }
```
Property Value

Type: Boolean
true if this instance is pointer type; otherwise, false.

Remarks

Pointer types can be dereferenced with the '^' operator.

Reference

IDataType Interface [1986]
TwinCAT.TypeSystem Namespace [1622]
IDataType.Category [1988]

6.11.38.1.8 IDataType.IsPrimitive Property

Gets a value indicating whether this IDataType [1986] is primitive

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0e15da1c14

Syntax

C#
bool IsPrimitive { get; }

Property Value

Type: Boolean
true if this instance is primitive; otherwise, false.

Reference

IDataType Interface [1986]
TwinCAT.TypeSystem Namespace [1622]

6.11.38.1.9 IDataType.IsReference Property

Gets a value indicating whether this IDataType [1986] is a reference type

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0e15da1c14

Syntax

C#
bool IsReference { get; }

Property Value

Type: Boolean
true if this instance is container type; otherwise, false.
Remarks
Reference types can be dereferenced.

Reference

IDataType Interface [1986]
TwinCAT.TypeSystem Namespace [1622]

6.11.38.1.10 IDatatype.Name Property

Gets the name of the Data Type (without namespace)

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
string Name { get; }

Property Value

Type: String
The name.

Reference

IDataType Interface [1986]
TwinCAT.TypeSystem Namespace [1622]

6.11.38.1.11 IDatatype.Namespace Property

Gets the namespace string within the IDatatype [1986] exists.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
string Namespace { get; }

Property Value

Type: String
The namespace.

Reference

IDataType Interface [1986]
6.11.39 **IDataTypeCollection Interface**

Interface `IDataTypeCollection` implements the `[ICollection<T>`.

**Namespace:** `TwinCAT.TypeSystem`  
**Assembly:** `TwinCAT.Ads.Abstractions` (in `TwinCAT.Ads.Abstractions.dll`)  
**Version:** 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public interface IDataTypeCollection : IDataTypeCollection<IDataType>, ICollection<IDataType>, IEnumerable<IDataType>, IEnumerable
```

The `IDataTypeCollection` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>[ICollection&lt;T&gt;</code>.. (Inherited from <code>[ICollection&lt;T.Interface&gt;](#)</code>.)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the <code>[ICollection&lt;T&gt;</code> is read-only. (Inherited from <code>[ICollection&lt;T.Interface&gt;](#)</code>.)</td>
</tr>
<tr>
<td>Item [1996]</td>
<td>Gets the <code>[IDataType](#)</code> with the specified name. (Inherited from <code>[IDataTypeCollection&lt;T.Interface&gt;](#)</code>.)</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the <code>[ICollection&lt;T&gt;</code>.. (Inherited from <code>[ICollection&lt;T.Interface&gt;](#)</code>.)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the <code>[ICollection&lt;T&gt;</code>.. (Inherited from <code>[ICollection&lt;T.Interface&gt;](#)</code>.)</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the <code>[ICollection&lt;T&gt;</code> contains a specific value. (Inherited from <code>[ICollection&lt;T.Interface&gt;](#)</code>.)</td>
</tr>
<tr>
<td>ContainsType [1997]</td>
<td>Determines whether the container contains the specified <code>[IDataType](#)</code> [1986]. (Inherited from <code>[IDataTypeCollection&lt;T.Interface&gt;](#)</code>.)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the <code>[ICollection&lt;T&gt;</code> to an <code>Array</code>, starting at a particular <code>Array</code> index. (Inherited from <code>[ICollection&lt;T.Interface&gt;](#)</code>.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from <code>[GetEnumerator](#)</code>.)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the <code>[ICollection&lt;T&gt;</code>.. (Inherited from <code>[Remove](#)</code>.)</td>
</tr>
<tr>
<td>TryGetType [1998]</td>
<td>Tries to get the specified <code>[IDataType](#)</code> from the <code>[IDataTypeCollection&lt;T.Interface&gt;](#)</code>.. (Inherited from <code>[TryGetType](#)</code>.)</td>
</tr>
</tbody>
</table>

**Reference**

`TwinCAT.TypeSystem Namespace` [1622]
6.11.39.1  **IDataTypeCollection Properties**

The `IDataTypeCollection` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;IDataType&gt;</code>.)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the <code>ICollection&lt;T&gt;</code> is read-only. (Inherited from <code>ICollection&lt;IDataType&gt;</code>.)</td>
</tr>
<tr>
<td>Item</td>
<td>Gets the <code>IDataType</code> with the specified name. (Inherited from <code>DataTypeCollection&lt;T&gt;</code>.)</td>
</tr>
</tbody>
</table>

Reference

`IDataTypeCollection Interface` [1993]

`TwinCAT.TypeSystem Namespace` [1622]

6.11.39.2  **IDataTypeCollection Methods**

The `IDataTypeCollection` type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;IDataType&gt;</code>.)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;IDataType&gt;</code>.)</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the <code>ICollection&lt;T&gt;</code> contains a specific value. (Inherited from <code>ICollection&lt;IDataType&gt;</code>.)</td>
</tr>
<tr>
<td>ContainsType</td>
<td>Determines whether the container contains the specified <code>IDataType</code>. (Inherited from <code>IDataTypeCollection&lt;T&gt;</code>.)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the <code>ICollection&lt;T&gt;</code> to an <code>Array</code>, starting at a particular <code>Array</code> index. (Inherited from <code>ICollection&lt;IDataType&gt;</code>.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from <code>IEnumerable&lt;IDataType&gt;</code>.)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;IDataType&gt;</code>.)</td>
</tr>
<tr>
<td>TryGetType</td>
<td>Tries to get the specified <code>IDataType</code> from the <code>DataTypeCollection&lt;T&gt;</code> (Inherited from <code>IDataTypeCollection&lt;T&gt;</code>.)</td>
</tr>
</tbody>
</table>

Reference

`IDataTypeCollection Interface` [1993]

`TwinCAT.TypeSystem Namespace` [1622]
### IDataTypeCollection<T> Interface

Data Type container interface

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public interface IDataTypeCollection<T> : ICollection<T>, IEnumerable<T>, IEnumerable
where T : class, IDataType
```

**Type Parameters**

- **T**  

  Data Type type.

The `IDataTypeCollection<T>` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the <code>ICollection&lt;T&gt;</code> is read-only. (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>Item [1996]</td>
<td>Gets the <code>IDataType [1986]</code> with the specified name.</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the <code>ICollection&lt;T&gt;</code> contains a specific value. (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>ContainsType [1997]</td>
<td>Determines whether the container contains the specified <code>IDataType [1986]</code>.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the <code>ICollection&lt;T&gt;</code> to an <code>Array</code>, starting at a particular <code>Array</code> index. (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from <code>IEnumerable&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
</tbody>
</table>
| TryGetType [1998] | Tries to get the specified `IDataType [1986]` from the `IDataTypeCollection<T>`.

**Reference**

TwinCAT.TypeSystem Namespace [1622]
6.11.40.1 `IDataTypeCollection.T. Properties`

The `IDataTypeCollection.T. [1995]` generic type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Count</strong></td>
<td>Gets the number of elements contained in the <code>ICollection.T. (Inherited from ICollection.T [1995]..)</code></td>
</tr>
<tr>
<td><strong>IsReadOnly</strong></td>
<td>Gets a value indicating whether the <code>ICollection.T</code> is read-only. (Inherited from ICollection.T [1995]..)</td>
</tr>
<tr>
<td><strong>Item [1996]</strong></td>
<td>Gets the `IDataType [1986] with the specified name.</td>
</tr>
</tbody>
</table>

### Reference

`IDataTypeCollection.T. Interface [1995]`

`TwinCAT.TypeSystem Namespace [1622]`

#### 6.11.40.1.1 `IDataTypeCollection.T..Item Property`

Gets the `IDataType [1986] with the specified name.`

**Namespace:** `TwinCAT.TypeSystem [1622]`

**Assembly:** `TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`

### Syntax

**C#**

```csharp
T this[string name] { get; }
```

### Parameters

- **name**
  - Type: `System.String`
  - The name.

### Return Value

- **Type:** `T [1995]
  - `T`.

### Reference

`IDataTypeCollection.T. Interface [1995]`

`TwinCAT.TypeSystem Namespace [1622]`

#### 6.11.40.2 `IDataTypeCollection.T. Methods`

The `IDataTypeCollection.T. [1995]` generic type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add</strong></td>
<td>Adds an item to the ICollection.T. (Inherited from ICollection_T [1995].)</td>
</tr>
<tr>
<td><strong>Clear</strong></td>
<td>Removes all items from the ICollection.T. (Inherited from ICollection_T [1995].)</td>
</tr>
<tr>
<td><strong>Contains</strong></td>
<td>Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection_T [1995].)</td>
</tr>
<tr>
<td><strong>ContainsType</strong></td>
<td>Determines whether the container contains the specified IDataType [1997].</td>
</tr>
<tr>
<td><strong>CopyTo</strong></td>
<td>Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection_T [1995].)</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong></td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable_T [1995].)</td>
</tr>
<tr>
<td><strong>Remove</strong></td>
<td>Removes the first occurrence of a specific object from the ICollection.T. (Inherited from ICollection_T [1995].)</td>
</tr>
<tr>
<td><strong>TryGetType</strong></td>
<td>Tries to get the specified IDataType [1986] from the IDataTypeCollection_T.T [1995].</td>
</tr>
</tbody>
</table>

### Reference

IDataTypeCollection_T.Interface [1995]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.40.2.1 IDataTypeCollection_T..ContainsType Method

Determines whether the container contains the specified IDataType [1986].

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
bool ContainsType(
    string name
)
```

### Parameters

**name**  
Type: System.String  
The name.

### Return Value

Type: Boolean  
true if contained; otherwise, false.

### Reference

IDataTypeCollection_T.Interface [1995]  
TwinCAT.TypeSystem Namespace [1622]
### 6.11.40.2.2 IDataTypeCollection.T..TryGetType Method

Tries to get the specified IDataType from the IDataTypeCollection.T.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
bool TryGetType(
    string name,
    out T type
)
```

**Parameters**

- **name**
  - Type: System.String
  - The name.
  
- **type**
  - Type: T
  - The type (Out parameter)

**Return Value**

Type: Boolean
true if found

**Reference**

IDataTypeCollection.T. Interface
TwinCAT.TypeSystem Namespace

### 6.11.41 IDimension Interface

Interface representing a single Dimension of an ArrayType.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public interface IDimension
```

The IDimension type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ElementCount</td>
<td>Gets the number of elements within that IDimension.</td>
</tr>
<tr>
<td>LowerBound</td>
<td>Gets the lower bound of elements within that IDimension.</td>
</tr>
</tbody>
</table>
6.11.41.1 IDimension Properties

The IDimension type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ElementCount</td>
<td>Gets the number of elements within that IDimension.</td>
</tr>
<tr>
<td>LowerBound</td>
<td>Gets the lower bound of elements within that IDimension.</td>
</tr>
</tbody>
</table>

6.11.41.1.1 IDimension.ElementCount Property

Gets the number of elements within that IDimension.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
int ElementCount { get; }
```

**Property Value**

Type: Int32

The element count.

6.11.41.1.2 IDimension.LowerBound Property

Gets the lower bound of elements within that IDimension.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
int LowerBound { get; }
```

Property Value

Type: Int32
The lower bound.

Reference

[IDimension Interface](#1998)
[TwinCAT.TypeSystem Namespace](#1622)

### 6.11.42 IDimensionCollection Interface

Interface IDimensionCollection

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IDimensionCollection : IList<IDimension>, ICollection<IDimension>, IEnumerable<IDimension>, IEnumerable
```

The IDimensionCollection type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T. (Inherited from ICollection.IDimension [1998].)</td>
</tr>
<tr>
<td>ElementCount</td>
<td>Gets the Number of elements in all Dimensions</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T is read-only. (Inherited from ICollection.IDimension [1998].)</td>
</tr>
<tr>
<td>Item</td>
<td>Gets or sets the element at the specified index. (Inherited from IList.IDimension [1998].)</td>
</tr>
<tr>
<td>LowerBounds</td>
<td>Gets the lower bounds.</td>
</tr>
<tr>
<td>UpperBounds</td>
<td>Gets the lower bounds.</td>
</tr>
</tbody>
</table>

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection.T. (Inherited from ICollection.IDimension [1998].)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T. (Inherited from ICollection.IDimension [1998].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IDimension [1998].)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IDimension [1998].)</td>
</tr>
<tr>
<td>GetDimensionLengths [2004]</td>
<td>Gets an array the specifies the Lengths of each Array Dimension</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IDimension [1998].)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList.T. (Inherited from IList.IDimension [1998].)</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList.T. at the specified index. (Inherited from IList.IDimension [1998].)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection.T. (Inherited from ICollection.IDimension [1998].)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IList.T. item at the specified index. (Inherited from IList.IDimension [1998].)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]

TwinCAT.TypeSystem.IDimension [1998]

6.11.42.1 IDimensionCollection Properties

The IDimensionCollection [2000] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IDimension [1998].)</td>
</tr>
<tr>
<td>ElementCount</td>
<td>Gets the Number of elements in all Dimensions</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IDimension [1998].)</td>
</tr>
<tr>
<td>Item</td>
<td>Gets or sets the element at the specified index. (Inherited from IList.IDimension [1998].)</td>
</tr>
<tr>
<td>LowerBounds</td>
<td>Gets the lower bounds.</td>
</tr>
<tr>
<td>UpperBounds</td>
<td>Gets the lower bounds.</td>
</tr>
</tbody>
</table>

Reference

IDimensionCollection Interface [2000]

TwinCAT.TypeSystem Namespace [1622]
6.11.42.1.1  IDimensionCollection.ElementCount Property

Gets the Number of elements in all Dimensions

Namespace:  TwinCAT.TypeSystem

Syntax

C#

```c#
int ElementCount { get; }
```

Property Value

Type:  Int32

Reference

IDimensionCollection Interface
TwinCAT.TypeSystem Namespace

6.11.42.1.2  IDimensionCollection.LowerBounds Property

Gets the lower bounds.

Namespace:  TwinCAT.TypeSystem

Syntax

C#

```c#
int[] LowerBounds { get; }
```

Property Value

Type:  .Int32.
The lower bounds.

Reference

IDimensionCollection Interface
TwinCAT.TypeSystem Namespace

6.11.42.1.3  IDimensionCollection.UpperBounds Property

Gets the lower bounds.

Namespace:  TwinCAT.TypeSystem

Syntax

C#

```csharp
int[] UpperBounds { get; }
```

Property Value

Type: `Int32`

The lower bounds.

Reference

IDimensionCollection Interface [► 2000]

TwinCAT.TypeSystem Namespace [► 1622]

### 6.11.42.2 IDimensionCollection Methods

The `IDimensionCollection` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;IDimension&gt;[► 1998]</code>..)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;IDimension&gt;[► 1998]</code>..)</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the <code>ICollection&lt;T&gt;</code> contains a specific value. (Inherited from <code>ICollection&lt;IDimension&gt;[► 1998]</code>..)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the <code>ICollection&lt;T&gt;</code> to an <code>Array</code>, starting at a particular <code>Array</code> index. (Inherited from <code>ICollection&lt;IDimension&gt;[► 1998]</code>..)</td>
</tr>
<tr>
<td>GetDimensionLengths [► 2004]</td>
<td>Gets an array the specifies the Lengths of each Array Dimension</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from <code>IEnumerable&lt;IDimension&gt;[► 1998]</code>..)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the <code>IList&lt;T&gt;</code>. (Inherited from <code>IList&lt;IDimension&gt;[► 1998]</code>..)</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the <code>IList&lt;T&gt;</code> at the specified index. (Inherited from <code>IList&lt;IDimension&gt;[► 1998]</code>..)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;IDimension&gt;[► 1998]</code>..)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the <code>IList&lt;T&gt;</code> item at the specified index. (Inherited from <code>IList&lt;IDimension&gt;[► 1998]</code>..)</td>
</tr>
</tbody>
</table>

Reference

IDimensionCollection Interface [► 2000]

TwinCAT.TypeSystem Namespace [► 1622]
6.11.42.1  IDimensionCollection.GetDimensionLengths Method

Gets an array the specifies the Lengths of each Array Dimension

**Namespace:**  TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
int[] GetDimensionLengths();
```

**Return Value**

Type: `System.Int32[]`.

**Reference**

IDimensionCollection Interface

TwinCAT.TypeSystem Namespace

6.11.43  IDynamicSymbol Interface

Interface IDynamicSymbol

**Namespace:**  TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public interface IDynamicSymbol : ISymbol, IAttributedInstance, IInstance, IBitSize
```

The IDynamicSymbol type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the IInstance [2052] (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.) (Inherited from IInstance [2052]).</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from IInstance [2052]).</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982]).</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982]).</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol [2176].</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from ISymbol [2176].</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [2052]).</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [2176].</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [2052].</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this IInstance [2052] is static. (Inherited from IInstance [2052].</td>
</tr>
<tr>
<td>NormalizedName</td>
<td>Gets the normalized instance name (fixed name for dynamic property access)</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].</td>
</tr>
<tr>
<td>SubSymbols</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from IInstance [2052].</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

**6.11.43.1 IDynamicSymbol Properties**

The IDynamicSymbol [2004] type exposes the following members.
## TwinCAT.Ads Namespaces

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the Instance [2052] (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the Instance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the Instance [2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this Instance [2052] is static. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>NormalizedName [2007]</td>
<td>Gets the normalized instance name (fixed name for dynamic property access)</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance.)</td>
</tr>
</tbody>
</table>

### Reference

IDynamicSymbol Interface [› 2004]
TwinCAT.TypeSystem Namespace [› 1622]

#### 6.11.43.1.1 IDynamicSymbol.NormalizedName Property

Gets the normalized instance name (fixed name for dynamic property access)

**Namespace:** TwinCAT.TypeSystem [› 1622]
**Assembly:** TwinCAT.Ads.Abstractions.dll (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
string NormalizedName { get; }
```

**Property Value**

Type: String  
The normalized instance name (can be the same like InstanceName [› 2054])

**Reference**

IDynamicSymbol Interface [› 2004]
TwinCAT.TypeSystem Namespace [› 1622]
IInstance.InstanceName [› 2054]
ISymbolFactory.InvalidCharacters [› 2190]

#### 6.11.44 IDynamicSymbolLoader Interface

Dynamic symbol loader interface

**Namespace:** TwinCAT.TypeSystem [› 1622]
**Assembly:** TwinCAT.Ads.Abstractions.dll (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public interface IDynamicSymbolLoader : ISymbolLoader,  
    ISymbolProvider, ISymbolServer
```

The IDynamicSymbolLoader type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BuildInTypes [2202]</td>
<td>Gets the build in types. (Inherited from ISymbolLoader [2200].)</td>
</tr>
<tr>
<td>DataTypes [2206]</td>
<td>Gets the data types (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>DefaultValueEncoding [2206]</td>
<td>Gets the default value encoding. (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>RootNamespaceName [2204]</td>
<td>Gets the name of the root namespace (Inherited from ISymbolProvider [2203].)</td>
</tr>
<tr>
<td>Settings [2202]</td>
<td>Gets or sets the access Method (Inherited from ISymbolLoader [2200].)</td>
</tr>
<tr>
<td>Symbols [2207]</td>
<td>Gets the symbols. (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>SymbolsDynamic [2009]</td>
<td>Gets the symbols (late bound as dynamic objects)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDataTypesAsync [2208]</td>
<td>Gets the data types asynchronously. (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>GetDynamicSymbolsAsync [2010]</td>
<td>Gets the dynamic symbols asynchronously</td>
</tr>
<tr>
<td>GetSymbolsAsync [2208]</td>
<td>Gets the symbols asynchronously (Inherited from ISymbolServer [2205].)</td>
</tr>
</tbody>
</table>

Remarks

Dynamic symbols are late bound symbols who are created and will expand during runtime. These symbols can represent complex user defined type instances like PLC Structures and Arrays created during PLC access and assure type safe to their fields and elements.

Reference

TwinCAT>TypeSystem Namespace [1622]

6.11.44.1  IDynamicSymbolLoader Properties

The IDynamicSymbolLoader [2007] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BuildInTypes [2202]</td>
<td>Gets the build in types. (Inherited from ISymbolLoader [2200].)</td>
</tr>
<tr>
<td>DataTypes [2206]</td>
<td>Gets the data types (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>DefaultValueEncoding [2206]</td>
<td>Gets the default value encoding. (Inherited from ISymbolServer [2205].)</td>
</tr>
</tbody>
</table>

2008  Version: 1.1  TC1000
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RootNamespaceName</td>
<td>Gets the name of the root namespace (Inherited from ISymbolProvider.)</td>
</tr>
<tr>
<td>Settings</td>
<td>Gets or sets the access Method (Inherited from ISymbolLoader.)</td>
</tr>
<tr>
<td>Symbols</td>
<td>Gets the symbols. (Inherited from ISymbolServer.)</td>
</tr>
<tr>
<td>SymbolsDynamic</td>
<td>Gets the symbols (late bound as dynamic objects)</td>
</tr>
</tbody>
</table>

**Reference**

IDynamicSymbolLoader Interface [2007]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.44.1.1 IDynamicSymbolLoader.SymbolsDynamic Property

Gets the symbols (late bound as dynamic objects)

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```
IDynamicSymbolsCollection SymbolsDynamic { get; }
```

**Property Value**

Type: IDynamicSymbolsCollection [2010]

The dynamic symbols.

**Reference**

IDynamicSymbolLoader Interface [2007]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.44.2 IDynamicSymbolLoader Methods

The IDynamicSymbolLoader [2007] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDataTypesAsync</td>
<td>Gets the data types asynchronously. (Inherited from ISymbolServer.)</td>
</tr>
<tr>
<td>GetDynamicSymbolsAsync</td>
<td>Gets the dynamic symbols asynchronously</td>
</tr>
<tr>
<td>GetSymbolsAsync</td>
<td>Gets the symbols asynchronously (Inherited from ISymbolServer.)</td>
</tr>
</tbody>
</table>
**6.11.44.2.1 IDynamicSymbolLoader.GetDynamicSymbolsAsync Method**

Gets the dynamic symbols asynchronously.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
Task<ResultDynamicSymbols> GetDynamicSymbolsAsync(
    CancellationToken cancel
)
```

**Parameters**

cancel Type: System.Threading.CancellationToken

The cancellation token.

**Return Value**

Type: Task<ResultDynamicSymbols>

Task<ResultDynamicSymbols>.

**Reference**

IDynamicSymbolLoader Interface [► 2007]

TwinCAT.TypeSystem Namespace [► 1622]

---

**6.11.45 IDynamicSymbolsCollection Interface**

Interface IDynamicSymbolsContainer Implements the IDynamicMetaObjectProvider

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public interface IDynamicSymbolsCollection : IDynamicMetaObjectProvider, IEnumerable<ISymbol>, IEnumerable
```

The IDynamicSymbolsCollection type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable&lt;ISymbol&gt;[► 2176].)</td>
</tr>
</tbody>
</table>
6.11.45.1   IDynamicSymbolsCollection Methods

The IDynamicSymbolsCollection [2010] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>IEnumerable.IEnumerableSymbol)</td>
</tr>
</tbody>
</table>

Reference

IDynamicSymbolsCollection Interface [2010]

TwinCAT.TypeSystem Namespace [1622]

6.11.46   IDynamicValue Interface

Interface IDynamicValue implements the IDynamicMetaObjectProvider. Implements the IValue [2226], Implements the IStructValue [2167] Implements the IArrayValue [1976].

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IDynamicValue : IDynamicMetaObjectProvider,
    IValue, IStructValue, IArrayValue
```

The IDynamicValue type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Gets the age of the value (last successful read of the value) (Inherited from</td>
</tr>
<tr>
<td></td>
<td>IValue)</td>
</tr>
<tr>
<td>CachedRaw</td>
<td>Gets the cached Raw internal Data. (Inherited from IValue)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the data type bound to this IValue (Inherited from IValue)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IValue is a primitive value. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from IValue)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol bound to this IValue (Inherited from IValue)</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Gets the Time stamp of the last successful read of the Value (local user</td>
</tr>
<tr>
<td></td>
<td>time, UTC)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### UpdateMode

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Image] UpdateMode [2230]</td>
<td>Gets the update mode (not implemented yet) (Inherited from IValue [2226].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read [2231]</td>
<td>Reads the value (via ADS) (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>ReadAsync [2231]</td>
<td>Reads the value (via ADS) (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>ResolveValue [2232]</td>
<td>Resolves the Value object to its primitive value. (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>TryGetArrayElementValues [1978]</td>
<td>Returns Array Element values. (Inherited from IArrayValue [1976].)</td>
</tr>
<tr>
<td>TryGetIndexValue [1978]</td>
<td>Tries to get the specified Array Element (Inherited from IArrayValue [1976].)</td>
</tr>
<tr>
<td>TryGetMemberValue [2169]</td>
<td>Tries to get a property/Member value. (Inherited from IStructValue [2167].)</td>
</tr>
<tr>
<td>TryResolveValue [2232]</td>
<td>Tries to resolves the Value object to its primitive value. (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>TrySetIndexValue [1979]</td>
<td>Tries to set the indexed value on Arrays (Inherited from IArrayValue [1976].)</td>
</tr>
<tr>
<td>TrySetMemberValue [2170]</td>
<td>Tries to Set a Member/Property Value (Inherited from IStructValue [2167].)</td>
</tr>
<tr>
<td>Write [2233]</td>
<td>Writes the value (via ADS) (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>WriteAsync [2233]</td>
<td>Writes the value (via ADS) (Inherited from IValue [2226].)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]

System.Dynamic.IDynamicMetaObjectProvider

TwinCAT.TypeSystem.IValue [2226]

TwinCAT.TypeSystem.IStructValue [2167]

TwinCAT.TypeSystem.IArrayValue [1976]

### 6.11.46.1 IDynamicValue Properties

The IDynamicValue [2011] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age [2228]</td>
<td>Gets the age of the value (last successful read of the value) (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>CachedRaw [2228]</td>
<td>Gets the cached Raw internal Data. (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the data type bound to this <a href="#">IValue</a>. (Inherited from <a href="#">IValue</a>.)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this <a href="#">IValue</a> is a primitive value.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">IValue</a>.)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol bound to this <a href="#">IValue</a>. (Inherited from <a href="#">IValue</a>.)</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Gets the Time stamp of the last successful read of the Value (local user</td>
</tr>
<tr>
<td></td>
<td>time, UTC) (Inherited from <a href="#">IValue</a>.)</td>
</tr>
<tr>
<td>UpdateMode</td>
<td>Gets the update mode (not implemented yet) (Inherited from <a href="#">IValue</a>.)</td>
</tr>
</tbody>
</table>

**Reference**

[IDynamicValue Interface](#) [2011]

[TwinCAT.TypeSystem Namespace](#) [1622]

### 6.11.46.2 IDynamicValue Methods

The [IDynamicValue](#) type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Reads the value (via ADS) (Inherited from <a href="#">IValue</a>.)</td>
</tr>
<tr>
<td>ReadAsync</td>
<td>Reads the value (via ADS) (Inherited from <a href="#">IValue</a>.)</td>
</tr>
<tr>
<td>ResolveValue</td>
<td>Resolves the Value object to its primitive value. (Inherited from <a href="#">IValue</a>.)</td>
</tr>
<tr>
<td></td>
<td>Returns Array Element values. (Inherited from <a href="#">IArrayValue</a>.)</td>
</tr>
<tr>
<td></td>
<td>Tries to get the specified Array Element (Inherited from <a href="#">IArrayValue</a>.)</td>
</tr>
<tr>
<td></td>
<td>Tries to get a property/Member value. (Inherited from <a href="#">IStructValue</a>.)</td>
</tr>
<tr>
<td></td>
<td>Tries to resolves the Value object to its primitive value. (Inherited from</td>
</tr>
<tr>
<td></td>
<td><a href="#">IValue</a>.)</td>
</tr>
<tr>
<td></td>
<td>Tries to set the indexed value on Arrays (Inherited from <a href="#">IArrayValue</a>.)</td>
</tr>
<tr>
<td></td>
<td>Tries to Set a Member/Property Value (Inherited from <a href="#">IStructValue</a>.)</td>
</tr>
<tr>
<td>Write</td>
<td>Writes the value (via ADS) (Inherited from <a href="#">IValue</a>.)</td>
</tr>
<tr>
<td>WriteAsync</td>
<td>Writes the value (via ADS) (Inherited from <a href="#">IValue</a>.)</td>
</tr>
</tbody>
</table>

**Reference**

[IDynamicValue Interface](#) [2011]
### IEnumType Interface

Common Enum type interface

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public interface IEnumType : IAliasType, IDataType, IBitSize
```

The IEnumType type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1987]</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>BaseType [1956]</td>
<td>Gets the Base Type (Inherited from IAliasType [1954].)</td>
</tr>
<tr>
<td>BaseTypeName [1957]</td>
<td>Gets the BaseType name (Inherited from IAliasType [1954].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [1988]</td>
<td>Gets the Data Type category (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Comment [1988]</td>
<td>Gets the comment behind the variable declaration. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>EnumValues [2016]</td>
<td>Enumeration specification (if enum)</td>
</tr>
<tr>
<td>FullName [1989]</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name)  (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Id [1989]</td>
<td>Gets the ID of the DataType (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a container type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPointer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a pointer type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPrimitive [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is primitive (Inherited from IDataType [1986].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains [1987]</td>
<td>Determines whether the enum values contains the specified name</td>
</tr>
<tr>
<td>GetNames [1986]</td>
<td>Gets the filed names of the EnumType.T. [2021]</td>
</tr>
<tr>
<td>GetValues [1986]</td>
<td>Gets the values of the EnumType.T. [2021]</td>
</tr>
<tr>
<td>Parse [1986]</td>
<td>Parses a value name of the EnumType.T. [2021] and returns the value (as base type)</td>
</tr>
<tr>
<td>ToString [1986]</td>
<td>Returns a String that represents the specified value.</td>
</tr>
<tr>
<td>TryParse(String, IConvertible.) [1987]</td>
<td>Parses the value from value name.</td>
</tr>
<tr>
<td>TryParse(String, IEnumValue.) [1988]</td>
<td>Parses the value from value name.</td>
</tr>
</tbody>
</table>

#### Reference

TwinCAT.TypeSystem Namespace [1622]

### 6.11.47.1 IEnumType Properties

The IEnumType [1984] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1987]</td>
<td>Gets the attributes of the DataType [1986] (Inherited from DataType [1986].)</td>
</tr>
<tr>
<td>BaseType [1956]</td>
<td>Gets the Base Type (Inherited from AliasType [1954].)</td>
</tr>
<tr>
<td>BaseTypeName [1957]</td>
<td>Gets the BaseType name (Inherited from AliasType [1954].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from BitSize [1982].)</td>
</tr>
<tr>
<td>Category [1988]</td>
<td>Gets the Data Type category (Inherited from DataType [1986].)</td>
</tr>
</tbody>
</table>
TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comment [1988]</td>
<td>Gets the comment behind the variable declaration. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>EnumValues [2016]</td>
<td>Enumeration specification (if enum)</td>
</tr>
<tr>
<td>FullName [1989]</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Id [1989]</td>
<td>Gets the ID of the DataType (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a container type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPointer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a pointer type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPrimitive [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is primitive (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsReference [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is a reference type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Name [1992]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Namespace [1992]</td>
<td>Gets the namespace string within the IDataType [1986] exists. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
</tbody>
</table>

Reference
IEnumType Interface [2014]

TwinCAT.TypeSystem Namespace [1622]

6.11.47.1.1 IEnumType.EnumValues Property

Enumeration specification (if enum)

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
IEnumValueCollection EnumValues { get; }

Property Value

Type: IEnumValueCollection [2031]
The enum specification.
6.11.47.2  IEnumType Methods

The IEnumType [2014] type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains [2017]</td>
<td>Determines whether the enum values contains the specified name</td>
</tr>
<tr>
<td>GetNames [2018]</td>
<td>Gets the filed names of the IEnumType.T. [2021]</td>
</tr>
<tr>
<td>GetValues [2018]</td>
<td>Gets the values of the IEnumType.T. [2021]</td>
</tr>
<tr>
<td>Parse [2019]</td>
<td>Parses a value name of the IEnumType.T. [2021] and returns the value (as base type)</td>
</tr>
<tr>
<td>ToString [2019]</td>
<td>Returns a String that represents the specified value.</td>
</tr>
<tr>
<td>TryParse(String, IConvertible.) [2020]</td>
<td>Parses the value from value name.</td>
</tr>
<tr>
<td>TryParse(String, IEnumValue.) [2021]</td>
<td>Parses the value from value name.</td>
</tr>
</tbody>
</table>

## Reference

IEnumType Interface [2014]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.47.2.1  IEnumType.Contains Method

Determines whether the enum values contains the specified name

- **Namespace:** TwinCAT.TypeSystem [1622]
- **Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
bool Contains(
    string name
)
```

**Parameters**

- **name**  
  Type: System.String  
  The name.
Return Value

Type: Boolean
true if contains the value, otherwise, false.

Reference

[EnumType Interface [› 2014]

TwinCAT.TypeSystem Namespace [› 1622]

6.11.47.2.2  IEnumType.GetNames Method

Gets the field names of the [EnumType.T. [› 2021]

Namespace:  TwinCAT.TypeSystem [› 1622]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
string[] GetNames()  

Return Value

Type:  System.String[].

Reference

[EnumType Interface [› 2014]

TwinCAT.TypeSystem Namespace [› 1622]

6.11.47.2.3  IEnumType.GetValues Method

Gets the values of the [EnumType.T. [› 2021]

Namespace:  TwinCAT.TypeSystem [› 1622]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
IConvertible[] GetValues()  

Return Value

Type:  System.IConvertible.T[].

Reference

[EnumType Interface [› 2014]
6.11.47.2.4 IEnumType.Parse Method

Parses a value name of the IEnumType.T. and returns the value (as base type)

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
IConvertible Parse(
    string name
)
```

**Parameters**

- **name**  
  Type: System.String  
  The value name.

**Return Value**

Type: IConvertible  
T.

**Reference**

IEnumType Interface | 2014

TwinCAT.TypeSystem Namespace | 1622

6.11.47.2.5 IEnumType.ToString Method

Returns a String that represents the specified value.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
string ToString(
    IConvertible val
)
```

**Parameters**

- **val**  
  Type: System.IConvertible  
  The value.

**Return Value**

Type: String  
A String that represents this value.
6.11.47.2.6  IEnumType.TryParse Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryParse(String, IConvertible.)</td>
<td>Parses the value from value name.</td>
</tr>
<tr>
<td>TryParse(String, IEnumValue.)</td>
<td>Parses the value from value name.</td>
</tr>
</tbody>
</table>

C#

```csharp
bool TryParse(
    string name,
    out IConvertible value
)
```

Parameters

<table>
<thead>
<tr>
<th>name</th>
<th>Type: System.String</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The value name.</td>
</tr>
<tr>
<td>value</td>
<td>Type: System.IConvertible.</td>
</tr>
<tr>
<td></td>
<td>The value.</td>
</tr>
</tbody>
</table>

Return Value

Type: Boolean
true if value name was found, false otherwise.

Reference

<table>
<thead>
<tr>
<th>IEnumType Interface [› 2014]</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT.TypeSystem Namespace [› 1622]</td>
</tr>
</tbody>
</table>
IEnumType.TryParse Method (String, IEnumValue.)

Parses the value from value name.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
bool TryParse(
    string name,
    out IEnumValue value
)
```

**Parameters**

- `name`  
  Type: System.String  
  The value name.

- `value`  
  Type: TwinCAT.TypeSystem.IEnumValue  
  The value.

**Return Value**

Type: Boolean  
true if value name was found, false otherwise.

**Reference**

[IEnumType Interface](#)  
[TryParse Overload](#)  
[TwinCAT.TypeSystem Namespace](#)

### 6.11.48 IEnumType.T. Interface

Interface representing an enum type

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public interface IEnumType<T> : IAliasType,
    IDataType, IBitSize
where T : IConvertible
```

**Type Parameters**

- `T`  
  Base type of the Enum

The IEnumType.T. type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1987]</td>
<td>Gets the attributes of the [IDataType [1986]] (Inherited from [IDataType [1986]].)</td>
</tr>
<tr>
<td>BaseType [1956]</td>
<td>Gets the Base Type (Inherited from [IAliasType [1954]].)</td>
</tr>
<tr>
<td>BaseTypeName [1957]</td>
<td>Gets the BaseType name (Inherited from [IAliasType [1954]].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the [IDataType [1986]] in bits. (Inherited from [IBitSize [1982]].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from [IBitSize [1982]].)</td>
</tr>
<tr>
<td>Category [1988]</td>
<td>Gets the Data Type category (Inherited from [IDataType [1986]].)</td>
</tr>
<tr>
<td>Comment [1988]</td>
<td>Gets the comment behind the variable declaration. (Inherited from [IDataType [1986]].)</td>
</tr>
<tr>
<td>EnumValues [2024]</td>
<td>Enumeration specification (if enum)</td>
</tr>
<tr>
<td>FullName [1989]</td>
<td>Gets the full name of the [IDataType [1986]] (Namespace + Name) (Inherited from [IDataType [1986]].)</td>
</tr>
<tr>
<td>Id [1989]</td>
<td>Gets the ID of the DataType (Inherited from [IDataType [1986]].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from [IBitSize [1982]].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from [IBitSize [1982]].)</td>
</tr>
<tr>
<td>IsContainer [1990]</td>
<td>Gets a value indicating whether this [IDataType [1986]] is a container type (Inherited from [IDataType [1986]].)</td>
</tr>
<tr>
<td>IsPointer [1990]</td>
<td>Gets a value indicating whether this [IDataType [1986]] is a pointer type (Inherited from [IDataType [1986]].)</td>
</tr>
<tr>
<td>IsPrimitive [1991]</td>
<td>Gets a value indicating whether this [IDataType [1986]] is primitive (Inherited from [IDataType [1986]].)</td>
</tr>
<tr>
<td>IsReference [1991]</td>
<td>Gets a value indicating whether this [IDataType [1986]] is a reference type (Inherited from [IDataType [1986]].)</td>
</tr>
<tr>
<td>Name [1991]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from [IDataType [1986]].)</td>
</tr>
<tr>
<td>Namespace [1992]</td>
<td>Gets the namespace string within the [IDataType [1986]] exists. (Inherited from [IDataType [1986]].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from [IBitSize [1982]].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains [2025]</td>
<td>Determines whether the enum values contains the specified name</td>
</tr>
<tr>
<td>GetNames [2025]</td>
<td>Gets the filed names of the IEnumType.T.</td>
</tr>
<tr>
<td>GetValues [2026]</td>
<td>Gets the values of the IEnumType.T.</td>
</tr>
<tr>
<td>Parse [2026]</td>
<td>Parses a name of the IEnumType.T. and returns the value (as base type)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToString</td>
<td>Returns a String that represents the specified value.</td>
</tr>
<tr>
<td>TryParse</td>
<td>Parses the value from value name.</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

### 6.11.48.1 IEnumType.T. Properties

The IEnumType.T [2021] generic type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1987]</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from IDataType [1986]).</td>
</tr>
<tr>
<td>BaseType [1956]</td>
<td>Gets the Base Type (Inherited from IAliasType [1954]).</td>
</tr>
<tr>
<td>BaseType Name [1957]</td>
<td>Gets the BaseType name (Inherited from IAliasType [1954]).</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982]).</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982]).</td>
</tr>
<tr>
<td>Category [1988]</td>
<td>Gets the Data Type category (Inherited from IDataType [1986]).</td>
</tr>
<tr>
<td>Comment [1988]</td>
<td>Gets the comment behind the variable declaration. (Inherited from IDataType [1986]).</td>
</tr>
<tr>
<td>EnumValues [2024]</td>
<td>Enumeration specification (if enum)</td>
</tr>
<tr>
<td>FullName [1989]</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from IDataType [1986]).</td>
</tr>
<tr>
<td>Id [1989]</td>
<td>Gets the ID of the DataType (Inherited from IDataType [1986]).</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982]).</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982]).</td>
</tr>
<tr>
<td>IsContainer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a container type (Inherited from IDataType [1986]).</td>
</tr>
<tr>
<td>IsPointer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a pointer type (Inherited from IDataType [1986]).</td>
</tr>
<tr>
<td>IsPrimitive [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is primitive (Inherited from IDataType [1986]).</td>
</tr>
<tr>
<td>IsReference [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is a reference type (Inherited from IDataType [1986]).</td>
</tr>
<tr>
<td>Name [1992]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from IDataType [1986]).</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the IDataType exists. (Inherited from IDataType.)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType (Inherited from IBitSize.)</td>
</tr>
</tbody>
</table>

### Reference

IEnumType.T. Interface [2021]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.48.1.1 IEnumType.T..EnumValues Property

Enumeration specification (if enum)

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
IEnumValueCollection EnumValues { get; }
```

**Property Value**

Type: IEnumValueCollection [2031]
The enum specification.

**Reference**

IEnumType.T. Interface [2021]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.48.2 IEnumType.T. Methods

The IEnumType.T. [2021] generic type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>Determines whether the enum values contains the specified name</td>
</tr>
<tr>
<td>GetNames</td>
<td>Gets the filed names of the IEnumType.T. [2021]</td>
</tr>
<tr>
<td>GetValues</td>
<td>Gets the values of the IEnumType.T. [2021]</td>
</tr>
<tr>
<td>Parse</td>
<td>Parses a name of the IEnumType.T. [2021] and returns the value (as base type)</td>
</tr>
<tr>
<td>Tostring</td>
<td>Returns a String that represents the specified value.</td>
</tr>
<tr>
<td>TryParse</td>
<td>Parses the value from value name.</td>
</tr>
</tbody>
</table>
6.11.48.2.1  IEnumType.T..Contains Method

Determines whether the enum values contains the specified name

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
bool Contains(
    string name
)

Parameters

name  Type: System.String
The name.

Return Value

Type: Boolean
true if contains the value, otherwise, false.

Reference

IEnumType.T. Interface
TwinCAT.TypeSystem Namespace

6.11.48.2.2  IEnumType.T..GetNames Method

Gets the filed names of the IEnumType.T..

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
string[] GetNames()

Return Value

Type: String,
System.String[].

Reference

IEnumType.T. Interface

### 6.11.48.2.3 IEnumType.T..GetValues Method

 Gets the values of the IEnumType.T. 

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
T[] GetValues()
```

**Return Value**

Type: T[]

**Reference**

IEnumType.T. Interface

TwinCAT.TypeSystem Namespace

### 6.11.48.2.4 IEnumType.T..Parse Method

 Parses a name of the IEnumType.T. and returns the value (as base type)

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
T Parse(
    string name
)
```

**Parameters**

name Type: System.String
The name.

**Return Value**

Type: T

**Reference**

IEnumType.T. Interface

TwinCAT.TypeSystem Namespace
6.11.48.2.5  IEnumType.T.ToString Method

Returns a String that represents the specified value.

Namespace:  TwinCAT.TypeSystem
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
string ToString(
    T val
)

Parameters

val  Type: T

Return Value

Type: String
A String that represents this value.

Reference

IEnumType.T. Interface
TwinCAT.TypeSystem Namespace

6.11.48.2.6  IEnumType.T.TryParse Method

Parses the value from value name.

Namespace:  TwinCAT.TypeSystem
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
bool TryParse(
    string name,
    out T value
)

Parameters

name  Type: System.String
value  Type: T

Return Value

Type: Boolean
true if value name was found, false otherwise.
6.11.49 IEnumValue Interface

Generic interface for EnumValues

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IEnumValue
```

The IEnumValue type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagedBaseType</td>
<td>Gets the enumeration base type (sint, byte, short, ushort, int, uint, Int64, UInt64 supported)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Enumeration Value (value as string)</td>
</tr>
<tr>
<td>Primitive</td>
<td>Gets the (Primitive, BaseType) Value of the enumeration as object</td>
</tr>
<tr>
<td>RawValue</td>
<td>Gets the raw value of the enumeration (as byte array)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the Enum value (in bytes)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]

6.11.49.1 IEnumValue Properties

The IEnumValue [2028] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagedBaseType</td>
<td>Gets the enumeration base type (sint, byte, short, ushort, int, uint, Int64, UInt64 supported)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Enumeration Value (value as string)</td>
</tr>
<tr>
<td>Primitive</td>
<td>Gets the (Primitive, BaseType) Value of the enumeration as object</td>
</tr>
<tr>
<td>RawValue</td>
<td>Gets the raw value of the enumeration (as byte array)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the Enum value (in bytes)</td>
</tr>
</tbody>
</table>
Reference

IEnumValue Interface [› 2028]
TwinCAT.TypeSystem Namespace [› 1622]

6.11.49.1.1 IEnumValue.ManagedBaseType Property

Gets the enumeration base type (sint,byte,short,ushort,int,uint,Int64,UInt64 supported)

Namespace: TwinCAT.TypeSystem [› 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Type ManagedBaseType { get; }
```

Property Value

Type: Type
The type of the base.

Reference

IEnumValue Interface [› 2028]
TwinCAT.TypeSystem Namespace [› 1622]

6.11.49.1.2 IEnumValue.Name Property

Gets the name of the Enumeration Value (value as string)

Namespace: TwinCAT.TypeSystem [› 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
string Name { get; }
```

Property Value

Type: String
The name.

Reference

IEnumValue Interface [› 2028]
TwinCAT.TypeSystem Namespace [› 1622]

6.11.49.1.3 IEnumValue.Primitive Property

Gets the (Primitive, BaseType) Value of the enumeration as object
**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```
Object Primitive { get; }
```

**Property Value**

**Type:** Object

The object value.

**Reference**

IEnumValue Interface [2028]

TwinCAT.TypeSystem Namespace [1622]

**6.11.49.1.4 IEnumValue.RawValue Property**

Gets the raw value of the enumeration (as byte array)

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```
byte[] RawValue { get; }
```

**Property Value**

**Type:** Byte

The raw value.

**Reference**

IEnumValue Interface [2028]

TwinCAT.TypeSystem Namespace [1622]

**6.11.49.1.5 IEnumValue.Size Property**

Gets the size of the Enum value (in bytes)

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```
int Size { get; }
```
TwinCAT.Ads Namespaces

Property Value
Type: Int32
The size.

Reference
IEnumValue Interface [2028]

TwinCAT.TypeSystem Namespace [1622]

6.11.50 IEnumValueCollection Interface

Interface IEnumValueCollection

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IEnumValueCollection : IEnumValueCollection<IEnumValue, IConvertible>, ICollection<IEnumValue>, IEnumerable<IEnumValue>, IEnumerable
```

The IEnumValueCollection type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection. (Inherited from ICollection.IEnumValue...)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection is read-only. (Inherited from ICollection.IEnumValue...)</td>
</tr>
<tr>
<td>Item [2035]</td>
<td>Gets or sets the element at the specified index. (Inherited from IEnumValueCollection.TEnumValue, TValue...)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection. (Inherited from ICollection.IEnumValue...)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection. (Inherited from ICollection.IEnumValue...)</td>
</tr>
<tr>
<td>Contains(String) [2036]</td>
<td>Determines whether [contains] [the specified name]. (Inherited from IEnumValueCollection.TEnumValue, TValue...)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection contains a specific value. (Inherited from ICollection.IEnumValue...)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection to an Array, starting at a particular Array index. (Inherited from ICollection.IEnumValue...)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IEnumValue...)</td>
</tr>
<tr>
<td>GetNames [2037]</td>
<td>Gets the Value Names. (Inherited from IEnumValueCollection.TEnumValue, TValue...)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetValues</td>
<td>Gets the values. (Inherited from IEnumValueCollection.TEnumValue, TValue. [2037].)</td>
</tr>
<tr>
<td>Parse</td>
<td>Parses the specified name. (Inherited from IEnumValueCollection.TEnumValue, TValue. [2033].)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection. (Inherited from ICollection, IEnumValue. [2028].)</td>
</tr>
<tr>
<td>TryParse(String, TValue.)</td>
<td>Tries to parse the string value of the Enum. (Inherited from IEnumValueCollection.TEnumValue, TValue. [2033].)</td>
</tr>
</tbody>
</table>

### 6.11.50.1 IEnumValueCollection Properties

The IEnumValueCollection type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection. (Inherited from ICollection, IEnumValue. [2028].)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection is read-only. (Inherited from ICollection, IEnumValue. [2028].)</td>
</tr>
<tr>
<td>Item</td>
<td>Gets or sets the element at the specified index. (Inherited from IEnumValueCollection.TEnumValue, TValue. [2033].)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]

### 6.11.50.2 IEnumValueCollection Methods

The IEnumValueCollection type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection. (Inherited from ICollection, IEnumValue. [2028].)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection. (Inherited from ICollection, IEnumValue. [2028].)</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether [contains] the specified name. (Inherited from IEnumValueCollection.TEnumValue, TValue. [2033].)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection contains a specific value. (Inherited from ICollection, IEnumValue. [2028].)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection to an Array, starting at a particular Array index. (Inherited from ICollection, IEnumValue. [2028].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from(IEnumerable, IEnumValue..)</td>
</tr>
<tr>
<td>GetNames [2037]</td>
<td>Gets the Value Names. (Inherited from IEnumerableValueCollection.TEnumValue, TValue.[2033].)</td>
</tr>
<tr>
<td>GetValues [2037]</td>
<td>Gets the values. (Inherited from IEnumerableValueCollection.TEnumValue, TValue.[2033].)</td>
</tr>
<tr>
<td>Parse [2038]</td>
<td>Parses the specified name. (Inherited from IEnumerableValueCollection.TEnumValue, TValue.[2033].)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection.. (Inherited from ICollection, TEnumValue.[2028]..)</td>
</tr>
<tr>
<td>TryParse(String, TValue.) [2039]</td>
<td>Tries to parse the string value of the Enum. (Inherited from IEnumerableValueCollection.TEnumValue, TValue.[2033].)</td>
</tr>
</tbody>
</table>

### Reference

- IEnumValueCollection Interface [2031]
- TwinCAT.TypeSystem Namespace [1622]

### 6.11.51 IEnumValueCollection.TEnumValue, TValue. Interface

Interface for collections of IEnumValues [2028]. Implements the ICollection..

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public interface IEnumValueCollection<TEnumValue, TValue> : ICollection<TEnumValue>,
   IEnumerable<TEnumValue>, IEnumerable
   where TEnumValue : class, IEnumValue
   where TValue : IConvertible
```

**Type Parameters**

- TEnumValue
- TValue

The IEnumValueCollection.TEnumValue, TValue. type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.. (Inherited from ICollection, TEnumValue..)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.. is read-only. (Inherited from ICollection, TEnumValue..)</td>
</tr>
<tr>
<td>Item [2035]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Add</code></td>
<td>Adds an item to the <code>ICollection&lt;T&gt;.</code> (Inherited from <code>ICollection&lt;TEnumValue..&gt;</code>)</td>
</tr>
<tr>
<td><code>Clear</code></td>
<td>Removes all items from the <code>ICollection&lt;T&gt;.</code> (Inherited from <code>ICollection&lt;TEnumValue..&gt;</code>)</td>
</tr>
<tr>
<td><code>Contains(T)</code></td>
<td>Determines whether the <code>ICollection&lt;T&gt;</code> contains a specific value. (Inherited from <code>ICollection&lt;TEnumValue..&gt;</code>)</td>
</tr>
<tr>
<td><code>Contains(String)</code></td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
<tr>
<td><code>CopyTo</code></td>
<td>Copies the elements of the <code>ICollection&lt;T&gt;</code> to an <code>Array</code>, starting at a particular <code>Array</code> index. (Inherited from <code>ICollection&lt;TEnumValue..&gt;</code>)</td>
</tr>
<tr>
<td><code>GetEnumerator</code></td>
<td>Returns an enumerator that iterates through the collection. (Inherited from <code>IEnumerable&lt;TEnumValue..&gt;</code>)</td>
</tr>
<tr>
<td><code>GetNames</code></td>
<td>Gets the Value Names.</td>
</tr>
<tr>
<td><code>GetValues</code></td>
<td>Gets the values.</td>
</tr>
<tr>
<td><code>Parse(String, TEnumValue.)</code></td>
<td>Parses the specified name.</td>
</tr>
<tr>
<td><code>Remove</code></td>
<td>Removes the first occurrence of a specific object from the <code>ICollection&lt;T&gt;.</code> (Inherited from <code>ICollection&lt;TEnumValue..&gt;</code>)</td>
</tr>
<tr>
<td><code>TryParse(String, TEnumValue.)</code></td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
</tbody>
</table>

## Reference

**TwinCAT.TypeSystem Namespace** | 1622 |
---|---|
**System.Collections.Generic.ICollection<T>** |  |
6.11.51.1  IEnumValueCollection.TEnumValue, TValue..Item Property

Gets or sets the element at the specified index.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
TValue this[string name] { get; }
```

**Parameters**

- **name**
  - Type: System.String
  - The name of the value

**Return Value**

- Type: TValue

**Exception**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

- IEnumValueCollection.TEnumValue, TValue. Interface
- TwinCAT.TypeSystem Namespace

6.11.51.2  IEnumValueCollection.TEnumValue, TValue. Methods

The IEnumValueCollection.TEnumValue, TValue generic type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection&lt;T&gt;. (Inherited from ICollection&lt;TEnumValue&gt;</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection&lt;T&gt;. (Inherited from ICollection&lt;TEnumValue&gt;</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection&lt;T&gt; contains a specific value. (Inherited from ICollection&lt;TEnumValue&gt;</td>
</tr>
<tr>
<td>Contains(String) [2036]</td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection.T to an Array, starting at a particular Array index. (Inherited from ICollection.TEnumValue[2033].)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.TEnumValue[2033].)</td>
</tr>
<tr>
<td>GetValues [2037]</td>
<td>Gets the values.</td>
</tr>
<tr>
<td>Parse [2038]</td>
<td>Parses the specified name.</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection.T. (Inherited from ICollection.TEnumValue[2033].)</td>
</tr>
<tr>
<td>TryParse(String, TEnumValue.) [2039]</td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
<tr>
<td>TryParse(String, TValue.) [2039]</td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
</tbody>
</table>

### Reference

IEnumValueCollection.TEnumValue, TValue. Interface [2033]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.51.2.1 IEnumValueCollection.TEnumValue, TValue..Contains Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection.T contains a specific value. (Inherited from ICollection.TEnumValue[2033].)</td>
</tr>
<tr>
<td>Contains(String) [2036]</td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
</tbody>
</table>

### Reference

IEnumValueCollection.TEnumValue, TValue. Interface [2033]

TwinCAT.TypeSystem Namespace [1622]

### IEnumValueCollection.TEnumValue, TValue..Contains Method (String)

Determines whether [contains] [the specified name].

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
bool Contains(
    string value
)
```
Parameters

value 
Type: System.String
Value

Return Value

Type: Boolean
true if [contains] [the specified name]; otherwise, false.

Reference

IEnumValueCollection.TEnumValue, TValue. Interface [2033]
Contains Overload [2036]
TwinCAT.TypeSystem Namespace [1622]

6.11.51.2.2 IEnumValueCollection.TEnumValue, TValue..GetNames Method

Gets the Value Names.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
string[] GetNames()

Return Value

Type: String.
System.String[].

Reference

IEnumValueCollection.TEnumValue, TValue. Interface [2033]
TwinCAT.TypeSystem Namespace [1622]

6.11.51.2.3 IEnumValueCollection.TEnumValue, TValue..GetValues Method

Gets the values.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
TValue[] GetValues()
Return Value

Type: TValue
T[].

Reference

IEnumValueCollection.TEnumValue, TValue. Interface

TwinCAT.TypeSystem Namespace

6.11.51.4 IEnumValueCollection.TEnumValue, TValue..Parse Method

Parses the specified name.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
TValue Parse(
    string name
)

Parameters

name Type: System.String
The name.

Return Value

Type: TValue
T[].

Reference

IEnumValueCollection.TEnumValue, TValue. Interface

TwinCAT.TypeSystem Namespace

6.11.51.5 IEnumValueCollection.TEnumValue, TValue..TryParse Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryParse(String, TEnumValue.)</td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
<tr>
<td>TryParse(String, TValue.)</td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
</tbody>
</table>

Reference

IEnumValueCollection.TEnumValue, TValue. Interface
IEnumValueCollection.TEnumValue, TValue..TryParse Method (String, TEnumValue.)

Tries to parse the string value of the Enum.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
bool TryParse(
    string strValue,
    out TEnumValue value
)
```

**Parameters**

- **strValue**
  - Type: `System.String`
  - The value in string representation.
- **value**
  - Type: `TEnumValue`.
  - The value.

**Return Value**

Type: Boolean
true if XXXX, false otherwise.

**Reference**

- IEnumValueCollection.TEnumValue, TValue. Interface
- TryParse Overload
- TwinCAT.TypeSystem Namespace
Parameters

strValue
Type: System.String
The Value in string representation.

value
Type: TValue
The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

IEnumValueCollection.TEnumValue, TValue. Interface
TryParse Overload
TwinCAT.TypeSystem Namespace

6.11.52 IField Interface

Specifies a single field/member of a Struct DataType.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public interface IField : IAttributedInstance, IInstance, IBitSize

The IField type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the IInstance (Inherited from IInstance.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType of the IInstance. (Inherited from IInstance.)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.) (Inherited from IInstance.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from IInstance.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize.)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### IField Properties

The `IField` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from <code>IAtributedInstance</code>.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>IDataType</code> in bits. (Inherited from <code>IBitSize</code>.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes. (Inherited from <code>IBitSize</code>.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the <code>IInstance</code> (Inherited from <code>IInstance</code>.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the <code>IDataType</code> of the <code>IInstance</code>. (Inherited from <code>IInstance</code>.)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.)). (Inherited from <code>IInstance</code>.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)). (Inherited from <code>IInstance</code>.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full <code>DataType</code> but instead of some sort of bit mapping. (Inherited from <code>IBitSize</code>.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0). (Inherited from <code>IBitSize</code>.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the <code>IInstance</code> represents a Pointer type (Pointer TO). (Inherited from <code>IInstance</code>.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the <code>IInstance</code> represents a Reference type (REFERENCE TO). (Inherited from <code>IInstance</code>.)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this <code>IInstance</code> is static. (Inherited from <code>IInstance</code>.)</td>
</tr>
<tr>
<td>ParentType</td>
<td>Gets the Parent Struct/Union of this <code>IField</code>.</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on <code>IsBitType</code>. (Inherited from <code>IBitSize</code>.)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the <code>DataType</code> that is used for this <code>IInstance</code>. (Inherited from <code>IInstance</code>.)</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from <code>IAtributedInstance</code>.)</td>
</tr>
</tbody>
</table>

### Reference

`TwinCAT.TypeSystem Namespace` [1622]

## 6.11.52.1 IField Properties

The `IField` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from <code>IAtributedInstance</code>.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>IDataType</code> in bits. (Inherited from <code>IBitSize</code>.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes. (Inherited from <code>IBitSize</code>.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the <code>IInstance</code> (Inherited from <code>IInstance</code>.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the <code>IDataType</code> of the <code>IInstance</code>. (Inherited from <code>IInstance</code>.)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.)). (Inherited from <code>IInstance</code>.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)). (Inherited from <code>IInstance</code>.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full <code>DataType</code> but instead of some sort of bit mapping. (Inherited from <code>IBitSize</code>.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0). (Inherited from <code>IBitSize</code>.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the <code>IInstance</code> represents a Pointer type (Pointer TO). (Inherited from <code>IInstance</code>.)</td>
</tr>
</tbody>
</table>

---

TC1000          
Version: 1.1   
2041
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE TO) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this Instance [2052] is static. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>ParentType [2042]</td>
<td>Gets the Parent Struct/Union of this Field [2040].</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052]. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

Reference

IField Interface [2040]

TwinCAT.TypeSystem Namespace [1622]

6.11.52.1.1 IField.ParenType Property

Gets the Parent Struct/Union of this Field [2040].

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
IDataType ParentType { get; }

Property Value

Type: IDataType [1986]
The type of the parent.

Reference

IField Interface [2040]

TwinCAT.TypeSystem Namespace [1622]

6.11.53 IFieldCollection Interface

Interface IFieldCollection Implements the IInstanceCollection<T>. [2057]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

public interface IFieldCollection : IInstanceCollection<IField>,
   IList<IField>, ICollection<IField>, IEnumerable<IField>, IEnumerable

The IFieldCollection type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T. (Inherited from ICollection.IField)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T is read-only. (Inherited from ICollection.IField)</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the element at the specified index. (Inherited from IList.IField)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the IInstance with the specified instance path. (Inherited from InstanceCollection.T)</td>
</tr>
<tr>
<td>Mode</td>
<td>Gets the InstanceCollectionMode. (Inherited from InstanceCollection.T)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection.T. (Inherited from ICollection.IField)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T. (Inherited from ICollection.IField)</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether this collection contains an instance with the specified instance path. (Inherited from IInstanceCollection.T)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection.T contains a specific value. (Inherited from ICollection.IField)</td>
</tr>
<tr>
<td>ContainsName</td>
<td>Determines whether this collection contains an instance with the specified instance name. (Inherited from InstanceCollection.T)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection.T to an Array, starting at a particular Array index. (Inherited from ICollection.IField)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IField)</td>
</tr>
<tr>
<td>GetInstance</td>
<td>Gets the IInstance by instance path. (Inherited from InstanceCollection.T)</td>
</tr>
<tr>
<td>GetInstanceByName</td>
<td>Gets the IInstance by instance name. (Inherited from InstanceCollection.T)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the List.T. (Inherited from List.IField)</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the List.T at the specified index. (Inherited from List.IField)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection.T. (Inherited from ICollection.IField)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the List.T item at the specified index. (Inherited from List.IField)</td>
</tr>
</tbody>
</table>
## 6.11.53.1 IFIELDCOLLECTION Properties

The IFIELDCOLLECTION type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T. (Inherited from ICollection.IFIELD..)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T is read-only. (Inherited from ICollection.IFIELD..)</td>
</tr>
<tr>
<td>Item.INT32</td>
<td>Gets or sets the element at the specified index. (Inherited from IList.IFIELD..)</td>
</tr>
<tr>
<td>Item.STRING</td>
<td>Gets the IInstance with the specified instance path. (Inherited from IInstanceCollection.T..)</td>
</tr>
<tr>
<td>Mode</td>
<td>Gets the InstanceCollectionMode.. (Inherited from IInstanceCollection.T..)</td>
</tr>
</tbody>
</table>

### Reference

IFIELDCOLLECTION Interface [2042]

TwinCAT.TypeSystem Namespace [1622]

## 6.11.53.2 IFIELDCOLLECTION Methods

The IFIELDCOLLECTION type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection.T. (Inherited from ICollection.IFIELD..)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T. (Inherited from ICollection.IFIELD..)</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether this collection contains an instance with the specified instance path. (Inherited from IInstanceCollection.T..)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection.T contains a specific value. (Inherited from ICollection.T)</td>
</tr>
<tr>
<td>ContainsName [2062]</td>
<td>Determines whether this collection contains an instance with the specified instance name. (Inherited from InstanceCollection.T)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection.T to an Array, starting at a particular Array index. (Inherited from ICollection.T)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.T)</td>
</tr>
<tr>
<td>GetInstance [2062]</td>
<td>Gets the IInstance by instance path. (Inherited from InstanceCollection.T)</td>
</tr>
<tr>
<td>GetInstanceByName [2063]</td>
<td>Gets the IInstance by instance name. (Inherited from InstanceCollection.T)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList. (Inherited from IList.T)</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList at the specified index. (Inherited from IList.T)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection.T. (Inherited from ICollection.T)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IList item at the specified index. (Inherited from IList.T)</td>
</tr>
<tr>
<td>TryGetInstance [2063]</td>
<td>Tries to get the specified instance. (Inherited from InstanceCollection.T)</td>
</tr>
<tr>
<td>TryGetInstanceByName [2064]</td>
<td>Tries to get the specified instance by name. (Inherited from InstanceCollection.T)</td>
</tr>
<tr>
<td>TryGetMember [2045]</td>
<td>Gets the Field/Member with the specified name from the collection.</td>
</tr>
</tbody>
</table>

Reference

IFieldCollection Interface [2042]

TwinCAT.TypeSystem Namespace [1622]

6.11.53.2.1 IFieldCollection.TryGetMember Method

Gets the Field/Member with the specified name from the collection.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

bool TryGetMember(  
    string fieldName,  
    out IField symbol)

TwinCAT.Ads Namespaces
Parameters

fieldName  
Type: System.String  
Name of the field/member.

symbol  
Type: TwinCAT.TypeSystem.IField [2040].  
The symbol with the specified name.

Return Value

Type: Boolean  
true if found, false otherwise.

Reference

IFieldCollection Interface [2042]

TwinCAT.TypeSystem Namespace [1622]

6.11.54 IGenericTypeMarshaler Interface

Interface IGenericTypeMarshaler Implements the ITypeMarshaler [2216]

Namespace: TwinCAT.TypeSystem [1622]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

public interface IGenericTypeMarshaler : ITypeMarshaler

The IGenericTypeMarshaler type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanMarshal(Object)</td>
<td>Determines whether ADS can marshal the specified value (Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td>CanMarshal(Type)</td>
<td>Determines whether ADS can marshal the specified managed data type. (Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td>Marshal [2219]</td>
<td>(Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td>MarshalSize [2219]</td>
<td>Gets the byte size of the value when marshalled. (Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td>Unmarshal(Type, ReadOnlySpan, Void, Byte) [2220]</td>
<td>(Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td>Unmarshal.T, (ReadOnlySpan, Void, Byte) [2048]</td>
<td></td>
</tr>
</tbody>
</table>

Remarks

The IGenericTypeMarshaler extends the ITypeMarshaler [2216] by generic methods.
6.11.54.1 IGenericTypeMarshaler Methods

The IGenericTypeMarshaler [2046] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanMarshal(Object)</td>
<td>Determines whether ADS can marshal the specified value (Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td>CanMarshal(Type)</td>
<td>Determines whether ADS can marshal the specified managed data type. (Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td>Marshal</td>
<td>(Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td>MarshalSize</td>
<td>Gets the byte size of the value when marshalled. (Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td>Unmarshal(Type, ReadOnlySpan, Void, Byte)</td>
<td>(Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td>Unmarshal(T, (ReadOnlySpan, Void, Byte)]</td>
<td></td>
</tr>
</tbody>
</table>

Reference

IGenericTypeMarshaler Interface [2046]
TwinCAT.TypeSystem Namespace [1622]

6.11.54.1.1 IGenericTypeMarshaler.Unmarshal Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unmarshal(T, (ReadOnlySpan, Void, Byte))</td>
<td>(Inherited from ITypeMarshaler [2216].)</td>
</tr>
<tr>
<td>Unmarshal(Type, ReadOnlySpan, Void, Byte)</td>
<td></td>
</tr>
</tbody>
</table>

Reference

IGenericTypeMarshaler Interface [2046]
TwinCAT.TypeSystem Namespace [1622]
IGenericTypeMarshaler.Unmarshal<T>. Method (ReadOnlySpan`1, Void, Byte)

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
int Unmarshal<T>(
    ReadOnlySpan source,
    void encoding,
    byte value
)
```

**Parameters**

- **source**  
  Type: `ReadOnlySpan`

- **encoding**  
  Type: `System.Void`

- **value**  
  Type: `System.Byte`

**Type Parameters**

- **T**

**Return Value**

Type: `Int32`

**Reference**

IGenericTypeMarshaler Interface [2046]

Unmarshal Overload [2047]

TwinCAT.TypeSystem Namespace [1622]

---

6.11.55  **IHierarchicalSymbol Interface**

Bindable Symbol interface (for internal use only)

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public interface IHierarchicalSymbol : ISymbol,
    IAttributedInstance, IInstance, IBitSize
```

The IHierarchicalSymbol type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the Instance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the Instance [2052]. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods .) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the Instance [2052] represents a Pointer type (Pointer TO) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE TO) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this Instance [2052] is static. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052]. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetParent</td>
<td>Sets the parent of the Symbol</td>
</tr>
</tbody>
</table>

## Reference

TwinCAT.TypeSystem Namespace [1622]

### 6.11.55.1 IHierarchicalSymbol Properties

The IHierarchicalSymbol [2048] type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the IInstance [2052] (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this Instance [2052] is static. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984]</td>
</tr>
<tr>
<td>(Inherited from</td>
<td></td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052].</td>
</tr>
<tr>
<td>(Inherited from</td>
<td></td>
</tr>
<tr>
<td>Instance [2052].)</td>
<td>)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

Reference

IHierarchicalSymbol Interface [2048]

TwinCAT.TypeSystem Namespace [1622]

6.11.55.2 IHierarchicalSymbol Methods

The IHierarchicalSymbol [2048] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetParent [2051]</td>
<td>Sets the parent of the Symbol</td>
</tr>
</tbody>
</table>

Reference

IHierarchicalSymbol Interface [2048]

TwinCAT.TypeSystem Namespace [1622]

6.11.55.2.1 IHierarchicalSymbol.SetParent Method

Sets the parent of the Symbol

Namespace: TwinCAT.TypeSystem [1622]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
void SetParent(
    ISymbol symbol
)
```

Parameters

- symbol Type: TwinCAT.TypeSystem.ISymbol [2176]
  The symbol.
## 6.11.56 IInstance Interface

Interface specifying instance objects.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**
```
public interface IInstance : IBitSize
```

The `IInstance` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from <code>IBitSize [1982]</code>.)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the IInstance</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the <code>IDataType [1986]</code> of the IInstance.</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods (.))</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.))</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from <code>IBitSize [1982]</code>.)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <code>IBitSize [1982]</code>.)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the IInstance represents a Pointer type (Pointer TO)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the IInstance represents a Reference type (REFERENCE TO)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this IInstance is static.</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on <code>IsBitType [1984]</code> (Inherited from <code>IBitSize [1982]</code>.)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the <code>DataType [1986]</code> that is used for this IInstance.</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]
### 6.11.56.1 Instance Properties

The `IInstance` type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>IDataType</code> in bits. (Inherited from <code>IBitSize</code>.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from <code>IBitSize</code>.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the <code>IInstance</code></td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the <code>IDataType</code> of the <code>IInstance</code>.</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods .)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods .)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full <code>DataType</code> but instead of some sort of bit mapping (Inherited from <code>IBitSize</code>.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <code>IBitSize</code>)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the <code>IInstance</code> represents a Pointer type (Pointer TO)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the <code>IInstance</code> represents a Reference type (REFERENCE TO)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this <code>IInstance</code> is static.</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on <code>IsBitType</code> (Inherited from <code>IBitSize</code>)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the <code>DataType</code> that is used for this <code>IInstance</code></td>
</tr>
</tbody>
</table>

#### Reference

- `IInstance Interface`<br>- `TwinCAT.TypeSystem Namespace`<br>- `TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### 6.11.56.1.1 Instance.Comment Property

Gets the comment of the `IInstance`.

**Namespace:** `TwinCAT.TypeSystem`<br>
**Assembly:** `TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
string Comment { get; }
```
Property Value
Type: String
The comment.

Reference
IInstance Interface [2052]
TwinCAT.TypeSystem Namespace [1622]

6.11.56.1.2 IInstance.DataType Property
Gets the IDataType of the IInstance.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
IDataType DataType { get; }

Property Value
Type: IDataType
The type of the data.

Remarks
The DataType can be unresolved in rare circumstances and therefore could have value null.

Reference
IInstance Interface [2052]
TwinCAT.TypeSystem Namespace [1622]

6.11.56.1.3 IInstance.InstanceName Property
Gets the name of the instance (without periods .)

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
string InstanceName { get; }

Property Value
Type: String
The name of the instance.
### 6.11.56.1.4 IInstance.InstancePath Property

Gets the relative / absolute access path to the instance (with periods ( . ))

**Namespace:** TwinCAT.TypeSystem
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
string InstancePath { get; }
```

**Property Value**

Type: **String**  
The instance path.

**Remarks**

If this path is relative or absolute depends on the context. IMember [2065] are using relative paths, ISymbol [2176]s are using absolute ones.

### 6.11.56.1.5 IInstance.IsPointer Property

Indicates that the IInstance [2052] represents a Pointer type (Pointer TO)

**Namespace:** TwinCAT.TypeSystem
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
bool IsPointer { get; }
```

**Property Value**

Type: **Boolean**  
true if is ReferenceTo, otherwise false.
TwinCAT.TypeSystem Namespace

6.11.56.1.6 IInstance.IsReference Property

Indicates that the IInstance represents a Reference type (REFERENCE TO)

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

bool IsReference { get; }

Property Value

Type: Boolean
true if is ReferenceTo, otherwise false.

Reference

IInstance Interface
TwinCAT.TypeSystem Namespace

6.11.56.1.7 IInstance.IsStatic Property

Gets a value indicating whether this IInstance is static.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

bool IsStatic { get; }

Property Value

Type: Boolean
true if this instance is static; otherwise, false.

Reference

IInstance Interface
TwinCAT.TypeSystem Namespace

6.11.56.1.8 IInstance.TypeName Property

Gets the name of the DataType that is used for this IInstance.

Reference

IInstance Interface
TwinCAT.TypeSystem Namespace
Syntax

C#

```csharp
string TypeName { get; }
```

Property Value

Type: `String`
The name of the type.

Reference

IInstance Interface [2052]

TwinCAT.TypeSystem Namespace [1622]

6.11.57 IInstanceCollection<T>. Interface

Generic IInstanceCollection interface.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IInstanceCollection<T> : IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
where T : class, IInstance
```

Type Parameters

T

The IInstanceCollection<T> type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection&lt;T&gt; (Inherited from ICollection&lt;T&gt;)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection&lt;T&gt; is read-only. (Inherited from ICollection&lt;T&gt;)</td>
</tr>
<tr>
<td>Item(Int32)</td>
<td>Gets or sets the element at the specified index. (Inherited from IList&lt;T&gt;)</td>
</tr>
<tr>
<td>Item(String)</td>
<td>Gets the IInstance [2052] with the specified instance path.</td>
</tr>
<tr>
<td>Mode [2060]</td>
<td>Gets the InstanceCollectionMode [2075].</td>
</tr>
</tbody>
</table>
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection.T.. (Inherited from ICollection.T..)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T.. (Inherited from ICollection.T..)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection.T.. contains a specific value. (Inherited from ICollection.T..)</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether this collection contains an instance with the specified instance path.</td>
</tr>
<tr>
<td>ContainsName</td>
<td>Determines whether this collection contains an instance with the specified instance name.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection.T.. to an Array, starting at a particular Array index. (Inherited from ICollection.T..)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.T..)</td>
</tr>
<tr>
<td>GetInstance</td>
<td>Gets the IInstance by instance path.</td>
</tr>
<tr>
<td>GetInstanceByName</td>
<td>Gets the IInstance by instance name.</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList.T.. (Inherited from IList.T..)</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList.T.. at the specified index. (Inherited from IList.T..)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.T..)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IList.T.. item at the specified index. (Inherited from IList.T..)</td>
</tr>
<tr>
<td>TryGetInstance</td>
<td>Tries to get the specified instance.</td>
</tr>
<tr>
<td>TryGetInstanceByName</td>
<td>Tries to get the specified instance by name.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]

6.11.57.1 IInstanceCollection.T. Properties

The IInstanceCollection.T.. generic type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.T..)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T.. is read-only. (Inherited from ICollection.T..)</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the element at the specified index. (Inherited from IList.T..)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the IInstance with the specified instance path. (Inherited from IInstance.T..)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Mode</td>
<td>Gets the InstanceCollectionMode.</td>
</tr>
</tbody>
</table>

**Reference**

IInstanceCollection.T. Interface [2057]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.57.1.1 IInstanceCollection.T..Item Property

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the element at the specified index. (Inherited from IList&lt;T&gt;.)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the Instance with the specified instance path.</td>
</tr>
</tbody>
</table>

**Reference**

IInstanceCollection.T. Interface [2057]

TwinCAT.TypeSystem Namespace [1622]

### IInstanceCollection.T..Item Property (String)

Gets the Instance with the specified instance path.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
T this[string instancePath] { get; }
```

**Parameters**

instancePath Type: System.String

**Property Value**

Type: T [2057]

**Reference**

IInstanceCollection.T. Interface [2057]

Item Overload [2059]
6.11.57.1.2  \texttt{IInstanceCollection.T..Mode} Property

Gets the \texttt{InstanceCollectionMode}~\cite{2075}.

\textbf{Syntax}

\texttt{C#}

\begin{verbatim}
InstanceCollectionMode Mode { get; }
\end{verbatim}

\textbf{Property Value}

Type: \texttt{InstanceCollectionMode}~\cite{2075}

The mode.

\textbf{Reference}

\texttt{IInstanceCollection.T. Interface}~\cite{2057}

\texttt{TwinCAT.TypeSystem Namespace}~\cite{1622}

6.11.57.2  \texttt{IInstanceCollection.T. Methods}

The \texttt{IInstanceCollection.T}~\cite{2057} generic type exposes the following members.

\textbf{Methods}

\begin{tabular}{|l|l|}
\hline
\textbf{Name} & \textbf{Description} \\
\hline
\texttt{Add} & Adds an item to the \texttt{ICollection.T}. (Inherited from \texttt{ICollection.T}~\cite{2057}..) \\
\hline
\texttt{Clear} & Removes all items from the \texttt{ICollection.T}. (Inherited from \texttt{ICollection.T} \cite{2057}..) \\
\hline
\texttt{Contains(T)} & Determines whether the \texttt{ ICollection.T} contains a specific value. (Inherited from \texttt{ ICollection.T} \cite{2057}..) \\
\hline
\texttt{Contains(String)} \cite{2061} & Determines whether this collection contains an instance with the specified instance path. \\
\hline
\texttt{ContainsName} \cite{2062} & Determines whether this collection contains an instance with the specified instance name. \\
\hline
\texttt{CopyTo} & Copies the elements of the \texttt{ICollection.T} to an \texttt{Array}, starting at a particular \texttt{Array} index. (Inherited from \texttt{ICollection.T}~\cite{2057}..) \\
\hline
\texttt{GetEnumerator} & Returns an enumerator that iterates through the collection. (Inherited from \texttt{IEnumerable.T}~\cite{2057}..) \\
\hline
\texttt{GetInstance} \cite{2062} & Gets the \texttt{IInstance} \cite{2052} by instance path. \\
\hline
\texttt{GetInstanceByName} \cite{2063} & Gets the \texttt{IInstance} \cite{2052} by instance name. \\
\hline
\texttt{IndexOf} & Determines the index of a specific item in the \texttt{IList.T}. (Inherited from \texttt{IList.T} \cite{2057}..) \\
\hline
\end{tabular}
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert</td>
<td>Inserts an item to the <code>IList&lt;T&gt;</code> at the specified index. (Inherited from <code>IList&lt;T&gt;[..]</code>)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;T&gt;[..]</code>)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the <code>IList&lt;T&gt;</code> item at the specified index. (Inherited from <code>IList&lt;T&gt;[..]</code>)</td>
</tr>
<tr>
<td>TryGetInstance</td>
<td>Tries to get the specified instance.</td>
</tr>
<tr>
<td>TryGetInstanceByName</td>
<td>Tries to get the specified instance by name.</td>
</tr>
</tbody>
</table>

### Reference

- `IInstanceCollection<T>.Interface[..]`
- `TwinCAT.TypeSystem Namespace[.]1622`

## 6.11.57.2.1 InstanceCollection<T>.Contains Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(T)</td>
<td>Determines whether the <code>ICollection&lt;T&gt;</code> contains a specific value. (Inherited from <code>ICollection&lt;T&gt;[..]</code>)</td>
</tr>
<tr>
<td>Contains(String)[..]2061</td>
<td>Determines whether this collection contains an instance with the specified instance path.</td>
</tr>
</tbody>
</table>

### Reference

- `IInstanceCollection<T>.Interface[..]2057`
- `TwinCAT.TypeSystem Namespace[.]1622`

### InstanceCollection<T>.Contains Method (String)

Determines whether this collection contains an instance with the specified instance path.

**Namespace:** TwinCAT.TypeSystem[.]1622  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdc3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
bool Contains(
    string instancePath
)
```

### Parameters

- `instancePath`  
  Type: `System.String`  
  The instance path.
**Return Value**

Type: Boolean  
true if this collection contains the specified instance path; otherwise, false.

**Reference**

IInstanceCollection.T. Interface [2057]

Contains Overload [2061]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.57.2.2  IInstanceCollection.T..ContainsName Method

Determines whether this collection contains an instance with the specified instance name.

**Namespace**: TwinCAT.TypeSystem [1622]  
**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```c#
bool ContainsName(
    string instanceName
)
```

**Parameters**

instanceName  
Type: System.String  
Name of the instance.

**Return Value**

Type: Boolean  
true if this collection contains the specified instance path; otherwise, false.

**Reference**

IInstanceCollection.T. Interface [2057]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.57.2.3  IInstanceCollection.T..GetInstance Method

Gets the IInstance [2052] by instance path.

**Namespace**: TwinCAT.TypeSystem [1622]  
**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```c#
T GetInstance(
    string instancePath
)
```
Parameters

instancePath
Type: System.String
The instance path.

Return Value

Type: T [2057]
T.

Reference

IInstanceCollection.T. Interface [2057]
TwinCAT.TypeSystem Namespace [1622]

6.11.57.2.4 IInstanceCollection.T..GetInstanceByName Method

Gets the IInstance [2052] by instance name.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

IList<T> GetInstanceByName(
    string instanceName
)

Parameters

instanceName
Type: System.String
Name of the instance.

Return Value

Type: IList<T> [2057].
IList<T>.

Reference

IInstanceCollection.T. Interface [2057]
TwinCAT.TypeSystem Namespace [1622]

6.11.57.2.5 IInstanceCollection.T..TryGetInstance Method

Tries to get the specified instance.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
### Syntax

**C#**

```
bool TryGetInstance(
    string instancePath,
    out T symbol
)
```

**Parameters**

- **instancePath**
  - Type: `System.String`
  - The instance path.

- **symbol**
  - Type: `T` [2057]
  - The symbol.

**Return Value**

- Type: `Boolean`
  - `true` if the `IInstance` [2052] is found; otherwise, `false`.

**Reference**

- `IInstanceCollection.T.Interface` [2057]
- `TwinCAT.TypeSystem Namespace` [1622]

### 6.11.57.2.6 `IInstanceCollection.T..TryGetInstanceByName` Method

Tries to get the specified instance by name.

**Namespace:** `TwinCAT.TypeSystem` [1622]

**Assembly:** `TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```
bool TryGetInstanceByName(
    string instanceName,
    out IList<T> symbols
)
```

**Parameters**

- **instanceName**
  - Type: `System.String`
  - Name of the instance.

- **symbols**
  - Type: `System.Collections.Generic.IList<T>` [2057]
  - The found symbols.

**Return Value**

- Type: `Boolean`
  - `true` if the `IInstance` [2052] is found; otherwise, `false`.

**Reference**

- `IInstanceCollection.T.Interface` [2057]
6.11.58 **IMember Interface**

Specifies a single field/member of a **Struct DataType**.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public interface IMember : IField,
    IAttributedInstance, IInstance, IBitSize
```

The IMember type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>BitOffset [2067]</td>
<td>Gets the bit offset.</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteOffset [2067]</td>
<td>Gets the byte offset.</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the IInstance [2052] (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods .) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this IInstance [2052] is static. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>Offset [2068]</td>
<td>Gets the offset of the IMember within the parent IStructType [2162] in bits or bytes dependent on IsBitType.</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParentType</td>
<td>Gets the Parent Struct/Union of this IField. (Inherited from IField.)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType. (Inherited from BitSize.)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType that is used for this IInstance. (Inherited from IInstance.)</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance.)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace

### 6.11.58.1 IMember Properties

The IMember type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance.)</td>
</tr>
<tr>
<td>BitOffset</td>
<td>Gets the bit offset.</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>ByteOffset</td>
<td>Gets the byte offset.</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the IInstance. (Inherited from IInstance.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType of the IInstance. (Inherited from IInstance.)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.). (Inherited from IInstance.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)). (Inherited from IInstance.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the IInstance represents a Pointer type (Pointer TO) (Inherited from IInstance.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the IInstance represents a Reference type (REFERENCE TO) (Inherited from IInstance.)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this IInstance is static. (Inherited from IInstance.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Offset [2068]</td>
<td>Gets the offset of the IMember [2065] within the parent IStructType [2162] in bits or bytes dependent on IsBitType.</td>
</tr>
<tr>
<td>ParentType [2042]</td>
<td>Gets the Parent Struct/Union of this IField [2040]. (Inherited from IField [2040].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

**Reference**

IMember Interface [2065]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.58.1.1 IMember.BitOffset Property

Gets the bit offset.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
int BitOffset { get; }
```

**Property Value**

Type: Int32

The bit offset.

**Reference**

IMember Interface [2065]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.58.1.2 IMember.ByteOffset Property

Gets the byte offset.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
int ByteOffset { get; }
```
**Property Value**

Type: `Int32`
The byte offset.

**Reference**

- [IMember Interface](#)
- [TwinCAT.TypeSystem Namespace](#)

### 6.11.58 IMember.Offset Property

Gets the offset of the `IMember` within the parent `IStructType` in bits or bytes dependent on `IsBitType`.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
int Offset { get; }
```

**Property Value**

Type: `Int32`
The bit offset.

**Reference**

- [IMember Interface](#)
- [TwinCAT.TypeSystem Namespace](#)

### 6.11.59 IMemberCollection Interface

Interface `IMemberCollection` Implements the `IInstanceCollection<T>`.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public interface IMemberCollection : IInstanceCollection<IMember>, IList<IMember>, ICollection<IMember>, IEnumerable<IMember>, IEnumerable
```

The `IMemberCollection` type exposes the following members.
**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T. (Inherited from ICollection.IMember [2065]..)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T is read-only. (Inherited from ICollection.IMember [2065]..)</td>
</tr>
<tr>
<td>Item.String. [2059]</td>
<td>Gets the Instance [2052] with the specified instance path. (Inherited from InstanceCollection.T. [2057].)</td>
</tr>
<tr>
<td>Item.Int32.</td>
<td>Gets or sets the element at the specified index. (Inherited from List.IMember [2065]..)</td>
</tr>
<tr>
<td>Mode [2060]</td>
<td>Gets the InstanceCollectionMode [2075]. (Inherited from InstanceCollection.T. [2057].)</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection.T. (Inherited from ICollection.IMember [2065]..)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T. (Inherited from ICollection.IMember [2065]..)</td>
</tr>
<tr>
<td>Contains(String) [2061]</td>
<td>Determines whether this collection contains an instance with the specified instance path. (Inherited from InstanceCollection.T. [2057].)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection.T contains a specific value. (Inherited from ICollection.IMember [2065]..)</td>
</tr>
<tr>
<td>ContainsName [2062]</td>
<td>Determines whether this collection contains an instance with the specified instance name. (Inherited from InstanceCollection.T. [2057].)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection.T to an Array, starting at a particular Array index. (Inherited from ICollection.IMember [2065]..)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IMember [2065]..)</td>
</tr>
<tr>
<td>GetInstance [2062]</td>
<td>Gets the Instance [2052] by instance path. (Inherited from InstanceCollection.T. [2057].)</td>
</tr>
<tr>
<td>GetInstanceByName [2063]</td>
<td>Gets the Instance [2052] by instance name. (Inherited from InstanceCollection.T. [2057].)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the List.T. (Inherited from List.IMember [2065]..)</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the List.T at the specified index. (Inherited from List.IMember [2065]..)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection.T. (Inherited from ICollection.IMember [2065]..)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the List.T item at the specified index. (Inherited from List.IMember [2065]..)</td>
</tr>
<tr>
<td>TryGetInstance [2063]</td>
<td>Tries to get the specified instance. (Inherited from InstanceCollection.T. [2057].)</td>
</tr>
<tr>
<td>TryGetInstanceByName [2064]</td>
<td>Tries to get the specified instance by name. (Inherited from InstanceCollection.T. [2057].)</td>
</tr>
<tr>
<td>TryGetMember [2071]</td>
<td>Gets the member with the specified name from the collection.</td>
</tr>
</tbody>
</table>
6.11.59.1 IMemberCollection Properties

The IMemberCollection type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection. (Inherited from ICollection.IMember..)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection is read-only. (Inherited from ICollection.IMember..)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the Instance with the specified instance path. (Inherited from InstanceCollection..)</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the element at the specified index. (Inherited from List.IMember..)</td>
</tr>
<tr>
<td>Mode</td>
<td>Gets the InstanceCollectionMode. (Inherited from InstanceCollection..)</td>
</tr>
</tbody>
</table>

Reference

IMemberCollection Interface IMemberCollection Interface

TwinCAT.TypeSystem Namespace

6.11.59.2 IMemberCollection Methods

The IMemberCollection type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection. (Inherited from ICollection.IMember..)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection. (Inherited from ICollection.IMember..)</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether this collection contains an instance with the specified instance path. (Inherited from InstanceCollection..)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection contains a specific value. (Inherited from ICollection.IMember..)</td>
</tr>
<tr>
<td>ContainsName</td>
<td>Determines whether this collection contains an instance with the specified instance name. (Inherited from InstanceCollection..)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection to an Array, starting at a particular Array index. (Inherited from ICollection.IMember..)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IMember..)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>GetInstance</strong></td>
<td>Gets the <code>IInstance</code> by instance path. (Inherited from <code>InstanceCollection&lt;T, InstancePath&gt;</code>.)</td>
</tr>
<tr>
<td><strong>GetInstanceByName</strong></td>
<td>Gets the <code>IInstance</code> by instance name. (Inherited from <code>InstanceCollection&lt;T, InstancePath&gt;</code>.)</td>
</tr>
<tr>
<td><strong>IndexOf</strong></td>
<td>Determines the index of a specific item in the <code>IList&lt;T&gt;</code>. (Inherited from <code>IList&lt;IMember&gt;</code>.)</td>
</tr>
<tr>
<td><strong>Insert</strong></td>
<td>Inserts an item to the <code>IList&lt;T&gt;</code> at the specified index. (Inherited from <code>IList&lt;IMember&gt;</code>.)</td>
</tr>
<tr>
<td><strong>Remove</strong></td>
<td>Removes the first occurrence of a specific object from the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;IMember&gt;</code>.)</td>
</tr>
<tr>
<td><strong>RemoveAt</strong></td>
<td>Removes the <code>IList&lt;T&gt;</code> item at the specified index. (Inherited from <code>IList&lt;IMember&gt;</code>.)</td>
</tr>
<tr>
<td><strong>TryGetInstance</strong></td>
<td>Tries to get the specified instance. (Inherited from <code>InstanceCollection&lt;T, InstancePath&gt;</code>.)</td>
</tr>
<tr>
<td><strong>TryGetInstanceByName</strong></td>
<td>Tries to get the specified instance by name. (Inherited from <code>InstanceCollection&lt;T, InstancePath&gt;</code>.)</td>
</tr>
<tr>
<td><strong>TryGetMember</strong></td>
<td>Gets the member with the specified name from the collection.</td>
</tr>
</tbody>
</table>

**Reference**

`IMemberCollection Interface [2068]`

`TwinCAT.TypeSystem Namespace [1622]`

### 6.11.59.2.1 IMemberCollection.TryGetMember Method

Gets the member with the specified name from the collection.

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd72ca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
bool TryGetMember(
    string memberName,
    out TwinCAT.TypeSystem.IMember symbol
)
```

**Parameters**

- **memberName**  
  Type: `System.String`  
  Name of the member.

- **symbol**  
  Type: `TwinCAT.TypeSystem.IMember [2065]`.  
  The symbol.

**Return Value**

Type: Boolean  
true if found, false otherwise.
Reference

INamespaceCollection Interface [► 2068]

TwinCAT.TypeSystem Namespace [► 1622]

6.11.60 INamespaceCollection Interface

Interface INamespaceCollection

Namespace: TwinCAT.TypeSystem [► 1622]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C# public interface INamespaceCollection : INamespaceCollection<IDataType>, ICollection<INamespace<IDataType>>, IEnumerable<INamespace<IDataType>>, IEnumerable

The INamespaceCollection type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection&lt;T&gt;. (Inherited from ICollection&lt;INamespace&lt;T&gt;, IDatatype&gt;)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection&lt;T&gt; is read-only. (Inherited from ICollection&lt;INamespace&lt;T&gt;, IDatatype&gt;)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection&lt;T&gt;. (Inherited from ICollection&lt;INamespace&lt;T&gt;, IDatatype&gt;)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection&lt;T&gt;. (Inherited from ICollection&lt;INamespace&lt;T&gt;, IDatatype&gt;)</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the ICollection&lt;T&gt; contains a specific value. (Inherited from ICollection&lt;INamespace&lt;T&gt;, IDatatype&gt;)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection&lt;T&gt; to an Array, starting at a particular Array index. (Inherited from ICollection&lt;INamespace&lt;T&gt;, IDatatype&gt;)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable&lt;INamespace&lt;T&gt;, IDatatype&gt;)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection&lt;T&gt;. (Inherited from ICollection&lt;INamespace&lt;T&gt;, IDatatype&gt;)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [► 1622]
6.11.60.1 INamespaceCollection Properties

The INamespaceCollection [2072] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T. (Inherited from INamespace [2458].IDataType [1986]...)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T. is read-only. (Inherited from INamespace [2458].IDataType [1986]...)</td>
</tr>
</tbody>
</table>

Reference

INamespaceCollection Interface [2072]

TwinCAT.TypeSystem Namespace [1622]

6.11.60.2 INamespaceCollection Methods

The INamespaceCollection [2072] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection.T. (Inherited from INamespace [2458].IDataType [1986]...)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T. (Inherited from INamespace [2458].IDataType [1986]...)</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the ICollection.T. contains a specific value. (Inherited from INamespace [2458].IDataType [1986]...)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from INamespace [2458].IDataType [1986]...)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable [2458].IDataType [1986]...)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection.T. (Inherited from INamespace [2458].IDataType [1986]...)</td>
</tr>
</tbody>
</table>

Reference

INamespaceCollection Interface [2072]

TwinCAT.TypeSystem Namespace [1622]

6.11.61 INamespaceCollection.T. Interface

Interface INamespaceCollection

Namespace: TwinCAT.TypeSystem [1622]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
public interface INamespaceCollection<T> : ICollection<INamespace<T>>,
    IEnumerable<INamespace<T>>, IEnumerable
    where T : class, IDataType
```

Type Parameters

T

The `INamespaceCollection<T>` type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;INamespace&lt;T&gt;&gt;</code>. <code>T</code>)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the <code>ICollection&lt;T&gt;</code> is read-only. (Inherited from <code>ICollection&lt;INamespace&lt;T&gt;&gt;</code>. <code>T</code>)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;INamespace&lt;T&gt;&gt;</code>. <code>T</code>)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;INamespace&lt;T&gt;&gt;</code>. <code>T</code>)</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the <code>ICollection&lt;T&gt;</code> contains a specific value. (Inherited from <code>ICollection&lt;INamespace&lt;T&gt;&gt;</code>. <code>T</code>)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the <code>ICollection&lt;T&gt;</code> to an <code>Array</code>, starting at a particular <code>Array</code> index. (Inherited from <code>ICollection&lt;INamespace&lt;T&gt;&gt;</code>. <code>T</code>)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from <code>IEnumerable&lt;INamespace&lt;T&gt;&gt;</code>. <code>T</code>)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;INamespace&lt;T&gt;&gt;</code>. <code>T</code>)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]

6.11.61.1 `INamespaceCollection<T>` Properties

The `INamespaceCollection<T>` generic type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;INamespace&lt;T&gt;&gt;</code>. <code>T</code>)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the <code>ICollection&lt;T&gt;</code> is read-only. (Inherited from <code>ICollection&lt;INamespace&lt;T&gt;&gt;</code>. <code>T</code>)</td>
</tr>
</tbody>
</table>
6.11.61.2 INamespaceCollection<T>. Methods

The INamespaceCollection<T> generic type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection&lt;T&gt;, (Inherited from ICollection&lt;T&gt;...)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection&lt;T&gt;, (Inherited from ICollection&lt;T&gt;...)</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the ICollection&lt;T&gt; contains a specific value, (Inherited from ICollection&lt;T&gt;...)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection&lt;T&gt; to an Array, starting at a particular Array index, (Inherited from ICollection&lt;T&gt;...)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection, (Inherited from IEnumerable&lt;T&gt;...)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection&lt;T&gt;, (Inherited from ICollection&lt;T&gt;...)</td>
</tr>
</tbody>
</table>

**Reference**

INamespaceCollection<T>. Interface [› 2073]
TwinCAT.TypeSystem Namespace [› 1622]

---

6.11.62 InstanceCollectionMode Enumeration

**Enum InstanceCollectionMode**

- **Namespace:** TwinCAT.TypeSystem [› 1622]
- **Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public enum InstanceCollectionMode
```

**Members**

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Names</td>
<td>0</td>
<td>InstanceCollection&lt;T&gt; is organized with InstanceNames instead of Instance Paths</td>
</tr>
<tr>
<td>Path</td>
<td>1</td>
<td>InstanceCollection&lt;T&gt; is organized with InstancePaths in a flat list</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PathHierarchy</td>
<td>2</td>
<td>InstanceCollection&lt;T&gt; is organized with InstancePaths in a Hierarchy (Only Root objects appearing)</td>
</tr>
</tbody>
</table>

#### Reference

TwinCAT.TypeSystem Namespace [› 1622]

#### 6.11.63 InsufficientAccessRightsException Class

Insufficient rights for access

#### Inheritance Hierarchy

```
System.Object
    System.Exception
        TwinCAT.AdsException [› 57]
            TwinCAT.TypeSystem.SymbolException [› 2401]
                TwinCAT.TypeSystem.InsufficientAccessRightsException
```

**Namespace:** TwinCAT.TypeSystem [› 1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```c#
[SerializableAttribute]
public sealed class InsufficientAccessRightsException : SymbolException
```

The `InsufficientAccessRightsException` type exposes the following members.

#### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>InsufficientAccessRightsException</code> [› 2077]</td>
<td>Initializes a new instance of the <code>InsufficientAccessRightsException</code> class.</td>
</tr>
</tbody>
</table>

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the instance path. (Inherited from <code>SymbolException</code> [› 2401].)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>Symbol</strong></td>
<td>Gets the symbol. (Inherited from <code>SymbolException</code>.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>When overridden in a derived class, returns the <code>Exception</code> that is the root cause of one or more subsequent exceptions. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>When overridden in a derived class, sets the <code>SerializationInfo</code> with information about the exception. (Inherited from <code>SymbolException</code>)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]

6.11.63.1 **InsufficientAccessRightsException Constructor**

Initializes a new instance of the `InsufficientAccessRightsException` class.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public InsufficientAccessRightsException(
    IValueSymbol symbol,
    SymbolAccessRights requested
)
```

**Parameters**

- `symbol`: Type: `TwinCAT.TypeSystem.IValueSymbol` [2254]
  The symbol.
- `requested`: Type: `TwinCAT.TypeSystem.SymbolAccessRights` [2396]
  The requested.

### Reference

`InsufficientAccessRightsException Class` [2076]
6.11.63.2 InsufficientAccessRightsException Properties

The InsufficientAccessRightsException [2076] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the instance path. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

InsufficientAccessRightsException Class [2076]

TwinCAT.TypeSystem Namespace [1622]

6.11.63.3 InsufficientAccessRightsException Methods

The InsufficientAccessRightsException [2076] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

TwinCAT.TypeSystem Namespace [1622]
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Reference

- InsufficientAccessRightsException Class [› 2076]
- TwinCAT.TypeSystem Namespace [› 1622]

### 6.11.64 IOversamplingArrayInstance Interface

#### Interface IOversamplingArrayInstance

**Namespace:** TwinCAT.TypeSystem [› 1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

C#

```csharp
public interface IOversamplingArrayInstance : IArrayInstance,
    ISymbol, IAttributedInstance, IInstance, IBitSize
```

The IOversamplingArrayInstance type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance[› 1980].)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType[› 1986] in bits. (Inherited from IBitSize[› 1982].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize[› 1982].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol[› 2176].)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the Instance[› 2052] (Inherited from IInstance[› 2052].)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType[› 1986] of the Instance[› 2052]. (Inherited from IInstance[› 2052].)</td>
</tr>
<tr>
<td>Dimensions</td>
<td>Gets the dimensions as read only collection. (Inherited from IArrayInstance[› 1964].)</td>
</tr>
<tr>
<td>Elements</td>
<td>Gets the contained Array Elements as read only collection. (Inherited from IArrayInstance[› 1964].)</td>
</tr>
<tr>
<td>ElementType</td>
<td>Gets the type of the contained elements. (Inherited from IArrayInstance[› 1964].)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.) (Inherited from IInstance[› 2052].)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from IInstance[› 2052].)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize[› 1982].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the size of the object is byte aligned (BitSize % 8 == 0) (Inherited from IBitSize)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this ISymbol is persistent. (Inherited from ISymbol)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the IInstance represents a Pointer type (Pointer TO) (Inherited from IInstance)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the IInstance represents a Reference type (REFERENCE TO) (Inherited from IInstance)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this IInstance is static. (Inherited from IInstance)</td>
</tr>
<tr>
<td>Item</td>
<td>Gets the ISymbol with the specified indices. (Inherited from IArrayInstance)</td>
</tr>
<tr>
<td>OversamplingElement</td>
<td>Gets the oversampling element.</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol (Inherited from ISymbol)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType (Inherited from IBitSize)</td>
</tr>
<tr>
<td>SubSymbols</td>
<td>Gets the SubSymbols of the ISymbol (Inherited from ISymbol)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType that is used for this Instance (Inherited from IInstance)</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance)</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryGetElement(List&lt;Int32, ISymbol&gt;)</td>
<td>Tries to get the array element with the specified indices (jagged array support). (Inherited from IArrayInstance)</td>
</tr>
<tr>
<td>TryGetElement(Int32, ISymbol)</td>
<td>Tries to get the array element with specified indices (only first level on jagged arrays) (Inherited from IArrayInstance)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace

TwinCAT.TypeSystem.IArrayInstance
### IOversamplingArrayInstance Properties

The IOversamplingArrayInstance type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance[1980].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType in bits. (Inherited from IBitSize[1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize[1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol[2176].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the Instance (Inherited from Instance[2052].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType of the Instance (Inherited from Instance[2052].)</td>
</tr>
<tr>
<td>Dimensions [1967]</td>
<td>Gets the dimensions as read only collection. (Inherited from IArrayInstance[1964].)</td>
</tr>
<tr>
<td>Elements [1968]</td>
<td>Gets the contained Array Elements as read only collection. (Inherited from IArrayInstance[1964].)</td>
</tr>
<tr>
<td>ElementType [1968]</td>
<td>Gets the type of the contained elements. (Inherited from IArrayInstance[1964].)</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from Instance[2052].)</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from Instance[2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize[1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize[1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol[2176].)</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this Symbol is persistent. (Inherited from ISymbol[2176].)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the Instance represents a Pointer type (Pointer TO) (Inherited from Instance[2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol[2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol[2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol[2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the Instance represents a Reference type (REFERENCE TO) (Inherited from Instance[2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this Instance is static. (Inherited from Instance[2052].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Item [1969]</td>
<td>Gets the ISymbol [2176] with the specified indices. (Inherited from IArrayInstance [1964].)</td>
</tr>
<tr>
<td>OversamplingElement [2082]</td>
<td>Gets the oversampling element.</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

Reference

IOversamplingArrayInstance Interface [2079]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.64.1.1  IOversamplingArrayInstance.OversamplingElement Property

Gets the oversampling element.

*Namespace: TwinCAT.TypeSystem [1622]*

*Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14*

**Syntax**

**C#**

```csharp
ISymbol OversamplingElement { get; }
```

**Property Value**

Type: ISymbol [2176]

The oversampling element.

Reference

IOversamplingArrayInstance Interface [2079]

TwinCAT.TypeSystem Namespace [1622]
6.11.64.2 IOversamplingArrayInstance Methods

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryGetElement(IList&lt;Int32&gt;, ISymbol) [➤ 1970]</td>
<td>Tries to get the array element with the specified indices (jagged array support). (Inherited from IArrayInstance [➤ 1964].)</td>
</tr>
<tr>
<td>TryGetElement(Int32, ISymbol) [➤ 1971]</td>
<td>Tries to get the array element with specified indices (only first level on jagged arrays) (Inherited from IArrayInstance [➤ 1964].)</td>
</tr>
</tbody>
</table>

Reference

IOversamplingArrayInstance Interface [➤ 2079]
TwinCAT.TypeSystem Namespace [➤ 1622]

6.11.65 IPointerInstance Interface

Interface representing an instance of an IPointerType [➤ 2086]

Namespace: TwinCAT.TypeSystem [➤ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IPointerInstance : ISymbol, IAttributedInstance, IInstance, IBitSize
```

The IPointerInstance type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [➤ 1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [➤ 1980].)</td>
</tr>
<tr>
<td>BitSize [➤ 1984]</td>
<td>Gets the size of the IDataType [➤ 1986] in bits. (Inherited from IBitSize [➤ 1982].)</td>
</tr>
<tr>
<td>ByteSize [➤ 1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [➤ 1982].)</td>
</tr>
<tr>
<td>Category [➤ 2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [➤ 2176].)</td>
</tr>
<tr>
<td>Comment [➤ 2053]</td>
<td>Gets the comment of the IInstance [➤ 2052] (Inherited from IInstance [➤ 2052].)</td>
</tr>
<tr>
<td>DataType [➤ 2054]</td>
<td>Gets the IDataType [➤ 1986] of the IInstance [➤ 2052]. (Inherited from IInstance [➤ 2052].)</td>
</tr>
<tr>
<td>InstanceName [➤ 2054]</td>
<td>Gets the name of the instance (without periods .) (Inherited from IInstance [➤ 2052].)</td>
</tr>
</tbody>
</table>
## IPointerInstance Properties

The `IPointerInstance` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from <code>IAttributedInstance</code> [1980])</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the <code>IDataType</code> in bits. (Inherited from <code>IBitSize</code> [1982])</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the <code>ISymbol</code> [2176] (Inherited from <code>ISymbol</code> [2176]).</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the <code>DataType</code> [1986] that is used for this <code>IInstance</code> [2052]. (Inherited from <code>IInstance</code> [2052]).</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from <code>IAtributedInstance</code> [1980]).</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the IInstance [2052] (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this IInstance [2052] is static. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Reference [2086]</td>
<td>Gets the resolved reference of Pointer / Reference</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the IDataType [1986] that is used for this IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

**Reference**

IPointerInstance Interface [2083]

TwinCAT.TypeSystem Namespace [1622]
6.11.65.1.1 IPointerInstance.Reference Property

Gets the resolved reference of Pointer / Reference

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
ISymbol Reference { get; }
```

**Property Value**

Type: `ISymbol`

The reference symbol or NULL if PVOID Pointer.

**Reference**

IPointerInstance Interface

TwinCAT.TypeSystem Namespace

6.11.66 IPointerType Interface

Interface representing a pointer type

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public interface IPointerType : IDataType, IBitSize
```

The IPointerType type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1987]</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [1988]</td>
<td>Gets the Data Type category (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Comment [1988]</td>
<td>Gets the comment behind the variable declaration. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>FullName [1989]</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from IDataType [1986].)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### 6.11.66.1 IPointerType Properties

The `IPointerType` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the <code>IDataType</code> (Inherited from <code>IDataType</code>).</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>IDataType</code> in bits. (Inherited from <code>IBitSize</code>.</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from <code>IBitSize</code>.</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from <code>IDataType</code>).</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment behind the variable declaration. (Inherited from <code>IDataType</code>).</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the <code>IDataType</code> (Namespace + Name) (Inherited from <code>IDataType</code>).</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Id [1989]</td>
<td>Gets the ID of the DataType (Inherited from [IDataType [1986].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from [IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from [IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainer [1990]</td>
<td>Gets a value indicating whether this [DataType [1986] is a container type (Inherited from [IDataType [1986].)</td>
</tr>
<tr>
<td>IsPointer [1990]</td>
<td>Gets a value indicating whether this [DataType [1986] is a pointer type (Inherited from [IDataType [1986].)</td>
</tr>
<tr>
<td>IsPrimitive [1991]</td>
<td>Gets a value indicating whether this [DataType [1986] is primitive (Inherited from [IDataType [1986].)</td>
</tr>
<tr>
<td>IsReference [1991]</td>
<td>Gets a value indicating whether this [DataType [1986] is a reference type (Inherited from [IDataType [1986].)</td>
</tr>
<tr>
<td>Name [1992]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from [IDataType [1986].)</td>
</tr>
<tr>
<td>Namespace [1992]</td>
<td>Gets the namespace string within the [DataType [1986] exists. (Inherited from [IDataType [1986].)</td>
</tr>
<tr>
<td>ReferencedType [2088]</td>
<td>Gets the the referenced type.</td>
</tr>
<tr>
<td>ReferenceTypeName [2089]</td>
<td>Gets the name of the referenced datatype</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from [IBitSize [1982].)</td>
</tr>
</tbody>
</table>

**Reference**

[IPointerType Interface [2086]]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.66.1.1 IPointerType.ReferencedType Property

Gets the the referenced type.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
IDataType ReferencedType { get; }
```

**Property Value**

Type: [IDataType [1986]]

The datatype of the reference or null if not resolved.
6.11.66.1.2 IPointerType. ReferenceTypeName Property

Gets the name of the referenced datatype

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294.Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
string ReferenceTypeName { get; }
```

**Property Value**

Type: `String`

The name of the reference datatype.

---

6.11.67 IPrimitiveType Interface

Interface IPrimitiveType

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294.Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public interface IPrimitiveType : IDataType, IBitSize
```

The IPrimitiveType type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType (Inherited from IDataType.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from BitSize.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from IDataType.)</td>
</tr>
</tbody>
</table>
### IPrimitiveType Properties

The IPrimitiveType type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType (Inherited from IDataType.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from IDataType.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment behind the variable declaration. (Inherited from IDataType.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the <a href="#">IDataType</a> (Namespace + Name) (Inherited from <a href="#">IDataType</a>.)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the DataType (Inherited from <a href="#">IDataType</a>.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a fullDataType but instead of some sort of bit mapping (Inherited from <a href="#">IBitSize</a>.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <a href="#">IBitSize</a>.)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this <a href="#">DataType</a> is a container type (Inherited from <a href="#">IDataType</a>.)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this <a href="#">DataType</a> is a pointer type (Inherited from <a href="#">IDataType</a>.)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this <a href="#">DataType</a> is primitive (Inherited from <a href="#">IDataType</a>.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this <a href="#">DataType</a> is a reference type (Inherited from <a href="#">IDataType</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from <a href="#">IDataType</a>.)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the <a href="#">DataType</a> exists. (Inherited from <a href="#">IDataType</a>.)</td>
</tr>
<tr>
<td>PrimitiveFlags</td>
<td>Indicates types of different PrimitiveTypes with flags.</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType (Inherited from <a href="#">IBitSize</a>.)</td>
</tr>
</tbody>
</table>

**Reference**

[IPrimitiveType Interface](#)[2289]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.67.1.1 IPrimitiveType.PrimitiveFlags Property

Indicates types of different PrimitiveTypes with flags.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

`PrimitiveTypeFlags PrimitiveFlags { get; }`

**Property Value**

Type: PrimitiveTypeFlags [2289]

**Reference**

[IPrimitiveType Interface](#)[2289]
TwinCAT.TypeSystem Namespace [\p 1622]

### 6.11.68 IProcessImageAddress Interface

Interface describing a Process Image Address

**Namespace:** TwinCAT.TypeSystem [\p 1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public interface IProcessImageAddress : IBitSize
```

The `IProcessImageAddress` type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitSize [\p 1984]</td>
<td>Gets the size of the <code>IDataType \[\p 1986\]</code> in bits. (Inherited from <code>IBitSize \[\p 1982\].</code>)</td>
</tr>
<tr>
<td>ByteSize [\p 1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from <code>IBitSize \[\p 1982\].</code>)</td>
</tr>
<tr>
<td>IndexGroup [\p 2093]</td>
<td>Gets the index group of the Symbol</td>
</tr>
<tr>
<td>IndexOffset [\p 2093]</td>
<td>Gets the index offset of the Symbol</td>
</tr>
<tr>
<td>IsBitType [\p 1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full <code>DataType</code> but instead of some sort of bit mapping (Inherited from <code>IBitSize \[\p 1982\].</code>)</td>
</tr>
<tr>
<td>IsByteAligned [\p 1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <code>IBitSize \[\p 1982\].</code>)</td>
</tr>
<tr>
<td>IsVirtual [\p 2094]</td>
<td>Gets a value indicating whether this instance is virtual.</td>
</tr>
<tr>
<td>Size [\p 1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on <code>IsBitType \[\p 1984\]</code> (Inherited from <code>IBitSize \[\p 1982\].</code>)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [\p 1622]

### 6.11.68.1 IProcessImageAddress Properties

The `IProcessImageAddress \[\p 2092\]` type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitSize [\p 1984]</td>
<td>Gets the size of the <code>IDataType \[\p 1986\]</code> in bits. (Inherited from <code>IBitSize \[\p 1982\].</code>)</td>
</tr>
<tr>
<td>ByteSize [\p 1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from <code>IBitSize \[\p 1982\].</code>)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IndexGroup</td>
<td>Gets the index group of the Symbol</td>
</tr>
<tr>
<td>IndexOffset</td>
<td>Gets the index offset of the Symbol</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize)</td>
</tr>
<tr>
<td>IsVirtual</td>
<td>Gets a value indicating whether this instance is virtual.</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType (Inherited from IBitSize)</td>
</tr>
</tbody>
</table>

**Reference**

IProcessImageAddress Interface [2092]

TwinCAT.TypeSystem Namespace [1622]

**6.11.68.1.1 IProcessImageAddress.IndexGroup Property**

Gets the index group of the Symbol

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
uint IndexGroup { get; }
```

**Property Value**

Type: UInt32
The index group.

**Reference**

IProcessImageAddress Interface [2092]

TwinCAT.TypeSystem Namespace [1622]

**6.11.68.1.2 IProcessImageAddress.IndexOffset Property**

Gets the index offset of the Symbol

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
uint IndexOffset { get; }
```

Property Value

Type: UInt32
The index offset.

Reference

IProcessImageAddress Interface [2092]
TwinCAT.TypeSystem Namespace [1622]

6.11.68.1.3 IProcessImageAddress.IsVirtual Property

Gets a value indicating whether this instance is virtual.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
bool IsVirtual { get; }
```

Property Value

Type: Boolean
true if this instance is virtual; otherwise, false.

Remarks

Virtual symbols are are only organizational elements within the Symbols Hierarchy and cannot be accessed seperately by IndexGroup/IndexOffset, Value Read/Writes, notifications or handles.

Reference

IProcessImageAddress Interface [2092]
TwinCAT.TypeSystem Namespace [1622]

6.11.69 IReferenceInstance Interface

Interface representing an instance of an IReferenceType [2099]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
The IReferenceInstance type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the Instance [2052] (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the Instance [2052]. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods .) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the Instance [2052] represents a Pointer type (Pointer TO) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE TO) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this Instance [2052] is static. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>ReferencedType [2097]</td>
<td>Gets the referenced type</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

**Name** | **Description**
---|---
ResolvedByteSize [2098] | Get the ByteSize of the (completely) resolved Symbol
ResolvedCategory [2098] | Gets the Category of the (completely) resolved Symbol.
ResolvedType [2099] | Gets the (completely) resolved type
Size [1985] | Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)
SubSymbols [2182] | Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)
TypeName [2056] | Gets the name of the DataType [1986] that is used for this IInstance [2052], (Inherited from IInstance [2052].)
ValueEncoding [1982] | Gets the value encoding. (Inherited from IAttributedInstance [1980].)

**Reference**

TwinCAT.TypeSystem Namespace [1622]

### 6.11.69.1 IReferenceInstance Properties

The IReferenceInstance [2094] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the IInstance [2052] (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this ISymbol is persistent. (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the IInstance represents a Pointer type (Pointer TO) (Inherited from Instance.)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the Instance represents a Reference type (REFERENCE TO) (Inherited from Instance.)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this Instance is static. (Inherited from Instance.)</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>ReferencedType</td>
<td>Gets the referenced type</td>
</tr>
<tr>
<td>ResolvedByteSize</td>
<td>Get the ByteSize of the (completely) resolved Symbol</td>
</tr>
<tr>
<td>ResolvedCategory</td>
<td>Gets the Category of the (completely) resolved Symbol.</td>
</tr>
<tr>
<td>ResolvedType</td>
<td>Gets the (completely) resolved type</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>SubSymbols</td>
<td>Gets the SubSymbols of the ISymbol (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType that is used for this Instance (Inherited from Instance.)</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance.)</td>
</tr>
</tbody>
</table>

**Reference**

IReferenceInstance Interface [2094]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.69.1.1 IReferenceInstance.ReferencedType Property

Gets the referenced type

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

IDataType ReferencedType { get; }
### Property Value

**Type:** `IDataType`  
The type of the referenced type

**Remarks**

This is no complete resolution, only the next level. The referenced type can be a reference again.

**Reference**

[IReferenceInstance Interface](http://example.com)  
[TwinCAT.TypeSystem Namespace](http://example.com)

#### 6.11.69.1.2 IReferenceInstance.ResolvedByteSize Property

Get the ByteSize of the (completely) resolved Symbol

**Namespace:**  **TwinCAT.TypeSystem**  
**Assembly:**  **TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version:**  
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
int ResolvedByteSize { get; }
```

**Property Value**

**Type:** `Int32`  
The byte size of the resolved type.

**Reference**

[IReferenceInstance Interface](http://example.com)  
[TwinCAT.TypeSystem Namespace](http://example.com)

#### 6.11.69.1.3 IReferenceInstance.ResolvedCategory Property

Gets the Category of the (completely) resolved Symbol.

**Namespace:**  **TwinCAT.TypeSystem**  
**Assembly:**  **TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version:**  
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
DataTypeCategory ResolvedCategory { get; }
```

**Property Value**

**Type:** `DataTypeCategory`  
The resolved category.
6.11.69.1.4  IReferenceInstance.ResolvedType Property

Gets the (completely) resolved type

Namespace:  TwinCAT.TypeSystem [1622]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  
IDataType ResolvedType { get; }  

Property Value

Type:  |DataType [1986]
The type of the resolved symbol

Reference

IReferenceInstance Interface [2094]
TwinCAT.TypeSystem Namespace [1622]

6.11.70   IReferenceType Interface

Interface representing a reference/pointer type

Namespace:  TwinCAT.TypeSystem [1622]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  
public interface IReferenceType : IDataType, IBitSize

The IReferenceType type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes  (Inherited from</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category  (Inherited from</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Comment [1988]</td>
<td>Gets the comment behind the variable declaration. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>FullName [1989]</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Id [1989]</td>
<td>Gets the ID of the DataType (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a container type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPointer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a pointer type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPrimitive [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is primitive (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsReference [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is a reference type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Name [1992]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Namespace [1992]</td>
<td>Gets the namespace string within the IDataType [1986] exists. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>ReferencedType [2102]</td>
<td>Gets the the referenced type.</td>
</tr>
<tr>
<td>ReferencedTypeName [2102]</td>
<td>Gets the name of the referenced type.</td>
</tr>
<tr>
<td>ResolvedByteSize [2102]</td>
<td>Get the ByteSize of the (completely) resolved Symbol</td>
</tr>
<tr>
<td>ResolvedCategory [2103]</td>
<td>Gets the Category of the (completely) resolved Symbol.</td>
</tr>
<tr>
<td>ResolvedType [2103]</td>
<td>Gets the (completely) resolved type</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

### 6.11.70.1 IReferenceType Properties

The IReferenceType [2099] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1987]</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [1988]</td>
<td>Gets the Data Type category (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Comment [1988]</td>
<td>Gets the comment behind the variable declaration. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>FullName [1989]</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Id [1989]</td>
<td>Gets the ID of the DataType (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a container type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPointer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a pointer type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPrimitive [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is primitive (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsReference [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is a reference type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Name [1992]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Namespace [1992]</td>
<td>Gets the namespace string within the IDataType [1986] exists. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>ReferencedType [2102]</td>
<td>Gets the the referenced type.</td>
</tr>
<tr>
<td>ReferencedTypeName [2102]</td>
<td>Gets the name of the referenced type.</td>
</tr>
<tr>
<td>ResolvedByteSize [2102]</td>
<td>Get the ByteSize of the (completely) resolved Symbol</td>
</tr>
<tr>
<td>ResolvedCategory [2103]</td>
<td>Gets the Category of the (completely) resolved Symbol.</td>
</tr>
<tr>
<td>ResolvedType [2103]</td>
<td>Gets the (completely) resolved type</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
</tbody>
</table>

Reference

IReferenceType Interface [2099]

TwinCAT.TypeSystem Namespace [1622]
### 6.11.70.1.1 IReferenceType.ReferencedType Property

Gets the the referenced type.

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
IDataType ReferencedType { get; }

**Property Value**

Type: IDataType [1986]  
The data type of the referenced type or null if not resolved.

**Reference**

IReferenceType Interface [2099]  
TwinCAT.TypeSystem Namespace [1622]

### 6.11.70.1.2 IReferenceType.ReferencedTypeName Property

Gets the name of the referenced type.

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
string ReferencedTypeName { get; }

**Property Value**

Type: String  
The name of the referenced type.

**Reference**

IReferenceType Interface [2099]  
TwinCAT.TypeSystem Namespace [1622]

### 6.11.70.1.3 IReferenceType.ResolvedByteSize Property

Get the ByteSize of the (completely) resolved Symbol

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
int ResolvedByteSize { get; }
```

Property Value

Type: `Int32`
The size of the resolved byte.

Reference

`IReferenceType Interface` [2099]
`TwinCAT.TypeSystem Namespace` [1622]

### 6.11.70.1.4 `IReferenceType.ResolvedCategory` Property

Gets the Category of the (completely) resolved Symbol.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
DataTypeCategory ResolvedCategory { get; }
```

Property Value

Type: `DataTypeCategory` [1649]
The resolved category.

Reference

`IReferenceType Interface` [2099]
`TwinCAT.TypeSystem Namespace` [1622]

### 6.11.70.1.5 `IReferenceType.ResolvedType` Property

Gets the (completely) resolved type

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
IDataType ResolvedType { get; }
```

Property Value

Type: `IDataType` [1986]
The type of the resolved symbol
**Reference**

IReferenceType Interface [2099]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.71 IRpcCallableInstance Interface

Interface for an RPC callable PLC Method (Remote procedure call)

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public interface IRpcCallableInstance
```

The IRpcCallableInstance type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RpcMethods</td>
<td>Gets the Method descriptions for the IRpcCallableType [2121]</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvokeRpcMethod(String, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, Object, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, Object, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>InvokeRpcMethodA sync(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, Object, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, Object, Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object)</td>
<td>Invokes the specified RpcMethod of the IRpcCallableInstance.</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

### 6.11.71.1 IRpcCallableInstance Properties

The IRpcCallableInstance [2104] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RpcMethods</td>
<td>Gets the Method descriptions for the IRpcCallableType [2121]</td>
</tr>
</tbody>
</table>

**Reference**

IRpcCallableInstance Interface [2104]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.71.1.1 IRpcCallableInstance.RpcMethods Property

Gets the Method descriptions for the IRpcCallableType [2121]

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

**C#**

```csharp
IRpcMethodCollection RpcMethods { get; }
```

**Property Value**

Type: `IRpcMethodCollection` [2127]

The methods.

**Reference**

- IRpcCallableInstance Interface [2104]
- TwinCAT.TypeSystem Namespace [1622]

### 6.11.71.2 IRpcCallableInstance Methods

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="2107" alt="InvokeRpcMethod(String, Object)" /></td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td><img src="2109" alt="InvokeRpcMethod(String, Object, Object)" /></td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td><img src="2110" alt="InvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object)" /></td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td><img src="2113" alt="InvokeRpcMethodAsync(String, Object, CancellationToken)" /></td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td><img src="2114" alt="InvokeRpcMethodAsync(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)" /></td>
<td>Invokes the specified RPC Method asynchronously</td>
</tr>
<tr>
<td><img src="2116" alt="TryInvokeRpcMethod(String, Object, Object)" /></td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TryInvokeRpcMetho(d(String, _Object, _Object, _Object))</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>TryInvokeRpcMetho(d(String, _Object, _AnyTypeSpecifier, _AnyTypeSpecifier, _Object, _Object))</td>
<td>Invokes the the specified RpcMethod of the _IRpcCallableInstance[2104].</td>
</tr>
</tbody>
</table>

**Reference**

IRpcCallableInstance Interface [2104]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.71.2.1 IRpcCallableInstance.InvokeRpcMethod Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvokeRpcMethod(String, _Object))</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, _Object, _Object)</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, _Object, _AnyTypeSpecifier, _AnyTypeSpecifier, _Object, _Object))</td>
<td>Invokes the specified RPC Method</td>
</tr>
</tbody>
</table>

**Reference**

IRpcCallableInstance Interface [2104]

TwinCAT.TypeSystem Namespace [1622]

**IRpcCallableInstance.InvokeRpcMethod Method (String, .Object.)**

Invokes the specified RPC Method

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
Object InvokeRpcMethod(
    string methodName,
    Object[] inParameters
)
```

Parameters

- **methodName**
  Type: `System.String`
  The method name.

- **inParameters**
  Type: `System.Object`
  The input parameters or NULL

Return Value

Type: `Object`
The return value of the Method (as object).

Remarks

This method only supports primitive data types as inParameters. Any available outparameters will be ignored. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```csharp
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program
            // Call a Method that has the following signature (within MAIN Program)
            /*
               {attribute 'TcRpcEnable'}
               METHOD PUBLIC M_Add : INT
               VAR_INPUT
               i1 : INT := 0;
               i2 : INT := 0;
               END_VAR
            */
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});
            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});
        }
    }
}
```
// Browsing RpcMethods
foreach(IRpcMethod method in main.RpcMethods)
{
    string methodName = method.Name;

    foreach(IRpcMethodParameter parameter in method.Parameters)
    {
        string parameterName = parameter.Name;
        string parameterType = parameter.TypeName;
    }
    }
}

Reference

IRpcCallableInstance Interface [2104]

InvokeRpcMethod Overload [2107]

TwinCAT.TypeSystem Namespace [1622]

IRpcCallableInstance.InvokeRpcMethod Method (String, .Object., .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

Object InvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    out Object[] outParameters
)

Parameters

methodName Type: System.String
The method name.
inParameters Type: System.Object
The input parameters or NULL
outParameters Type: System.Object..
The output parameters.

Return Value

Type: Object
The return value of the Method (as object).

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.
Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            IQueryable {attribute 'TcRpcEnable'}
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
                i1 : INT := 0;
                i2 : INT := 0;
            END_VAR
            */ /*
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});

            // Browsing RpcMethods
            foreach(IRpcMethod method in main.RpcMethods)
            {
                string methodName = method.Name;
                foreach(IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameterName = parameter.Name;
                    string parameterType = parameter.TypeName;
                }
            }
        }
    }
}

Reference

IRpcCallableInstance Interface [2104]

InvokeRpcMethod Overload [2107]

TwinCAT.TypeSystem Namespace [1622]

IRpcCallableInstance.InvokeRpcMethod Method
(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

Object InvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters
)

Parameters

methodName
Type: System.String
The method name.

inParameters
Type: System.Object
The parameters.

outSpecifiers
Type: TwinCAT.TypeSystem.AnyTypeSpecifier[]
The out specifiers (specifying the out types) or NULL.

retSpecifier
Type: TwinCAT.TypeSystem.AnyTypeSpecifier[]
The ret specifier (specifying the return value) or NULL.

outParameters
Type: System.Object
The out parameters.

Return Value

Type: Object
The return value of the Method (as object).

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

```csharp
// Call a Method that has the following signature (within MAIN Program)
/* {attribute 'TcRpcEnable'}
METHOD PUBLIC M_Add : INT
VAR_INPUT
    i1 : INT := 0;
    i2 : INT := 0;
END_VAR
*/

short result = (short)main.InvokeRpcMethod("M_Add", new object[] { (short)3, (short)4 });

// Call a Method that has no parameter and returns VOID
main.InvokeRpcMethod("M_Method1", new object[] {});

// Browsing RpcMethods
foreach(IRpcMethod method in main.RpcMethods)
{
    string methodName = method.Name;
    foreach(IRpcMethodParameter parameter in method.Parameters)
    {
        string parameterName = parameter.Name;
        string parameterType = parameter.TypeName;
    }
}
```

Reference

IRpcCallableInstance Interface [2104]

InvokeRpcMethod Overload [2107]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.71.2.2 IRpcCallableInstance.InvokeRpcMethodAsync Method

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![MethodName]</td>
<td>![MethodName]</td>
</tr>
<tr>
<td>![MethodName]</td>
<td>![MethodName]</td>
</tr>
</tbody>
</table>

Reference

IRpcCallableInstance Interface [2104]

TwinCAT.TypeSystem Namespace [1622]
IRpcCallableInstance.InvokeRpcMethodAsync Method (String, .Object., CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace:  TwinCAT.TypeSystem [1622]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Task<ResultRpcMethodAccess> InvokeRpcMethodAsync(
    string methodName,
    Object[] inParameters,
    CancellationToken cancel
)
```

Parameters

- **methodName**
  - Type: System.String
  - The method name.

- **inParameters**
  - Type: System.Object
  - The parameters.

- **cancel**
  - Type: System.Threading.CancellationToken
  - The cancellation token

Return Value

Type: Task<ResultRpcMethodAccess> [2571].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethodAccess [2571] results contains the return value (ReturnValue [2574]) together with the output parameters. The succeeded communication is indicated by the ErrorCode property (ErrorCode [2559]) after the communication.

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        }
    }
}

ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

// Get the Symbols (Dynamic Symbols)
IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

// Call a Method that has the following signature (within MAIN Program)
/*
   * [attribute 'TcRpcEnable']
   METHOD PUBLIC M_Add : INT
   VAR_INPUT
   i1 : INT := 0;
   i2 : INT := 0;
   END_VAR
*/
short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short)3, (short)4});

// Call a Method that has no parameter and returns VOID
main.InvokeRpcMethod("M_Method1", new object[]{});

// Browsing RpcMethods
foreach(IRpcMethod method in main.RpcMethods)
{
    string methodName = method.Name;
    foreach(IRpcMethodParameter parameter in method.Parameters)
    {
        string parameterName = parameter.Name;
        string parameterType = parameter.TypeName;
    }
}

Reference

IRpcCallableInstance Interface [2104]

InvokeRpcMethodAsync Overload [2112]

TwinCAT.TypeSystem Namespace [1622]

IRpcCallableInstance.InvokeRpcMethodAsync Method

(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

Task<ResultRpcMethodAccess> InvokeRpcMethodAsync(
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    CancellationToken cancel
)

Parameters

methodName Type: System.String
The method name.
inParameters  Type: System.Object
The parameters.

outSpecifiers Type: TwinCAT.TypeSystem.AnyTypeSpecifier[1633]
The out specifiers (specifying the out types) or NULL.

retSpecifier Type: TwinCAT.TypeSystem.AnyTypeSpecifier[1633]
The ret specifier (specifying the return value) or NULL.

cancel Type: System.Threading.CancellationToken
The cancellation token

Return Value
Type: Task.ResultRpcMethodAccess[2571].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethodAccess[2571] results contains the return value (ReturnValue[2574]) together with the output parameters. The succeeded communication is indicated by the ErrorCode property (ErrorCode[2559]) after the communication.

Remarks
The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples
The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program
            // Call a Method that has the following signature (within MAIN Program)
            /* {attribute 'TcRpcEnable'}
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
                i1 : INT := 0;
                i2 : INT := 0;
            END_VAR
            */

            short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short)3, (short)4});

            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});
        }
    }
}
// Browsing RpcMethods
foreach (IRpcMethod method in main.RpcMethods)
{
    string methodName = method.Name;
    foreach (IRpcMethodParameter parameter in method.Parameters)
    {
        string parameterName = parameter.Name;
        string parameterType = parameter.TypeName;
    }
}

Reference
IRpcCallableInstance Interface [2104]
InvokeRpcMethodAsync Overload [2112]
TwinCAT.TypeSystem Namespace [1622]

6.11.71.2.3 IRpcCallableInstance.TryInvokeRpcMethod Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![InvokeRpcMethod(String, Object, Object)](InvokeRpcMethod(String, Object, Object)) ![2116](InvokeRpcMethod(String, Object, Object))</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>![InvokeRpcMethod(String, Object, Object, Object)](InvokeRpcMethod(String, Object, Object, Object)) ![2118](InvokeRpcMethod(String, Object, Object, Object))</td>
<td>Invokes the specified RPC Method</td>
</tr>
<tr>
<td>![InvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object)](InvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object)) ![2120](InvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object))</td>
<td>Invokes the the specified RpcMethod of the IRpcCallableInstance <img src="IRpcCallableInstance" alt="2104" />.</td>
</tr>
</tbody>
</table>

Reference
IRpcCallableInstance Interface [2104]
TwinCAT.TypeSystem Namespace [1622]

IRpcCallableInstance.TryInvokeRpcMethod Method (String, Object, Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstracts (in TwinCAT.Ads.Abstracts.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
int TryInvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    out Object retval
)
```

Parameters

- **methodName**: Type: `System.String`. The method name.
- **inParameters**: Type: `System.Object`. The parameters.
- **retval**: Type: `System.Object`. The return value of the RPC method as object.

Return Value

- **Type**: `Int32`. The result value of the call (ErrorCode). 0 means Succeeded.

Remarks

Because this overload doesn't provide any `AnyTypeSpecifier` specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

**Dynamic Tree Mode**

```csharp
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);

            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program

            // Call a Method that has the following signature (within MAIN Program)
            /* {attribute 'TcRpcEnable'}
             METHOD PUBLIC M_Add : INT
             VAR_INPUT
             i1 : INT := 0;
             i2 : INT := 0;
             END_VAR
             */
```
short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short)3, (short)4});

// Call a Method that has no parameter and returns VOID
main.InvokeRpcMethod("M_Method1", new object[]{});

// Browsing RpcMethods
foreach(IRpcMethod method in main.RpcMethods)
{
    string methodName = method.Name;
    foreach(IRpcMethodParameter parameter in method.Parameters)
    {
        string parameterName = parameter.Name;
        string parameterType = parameter.TypeName;
    }
}

Reference

IRpcCallableInstance Interface [2104]

TryInvokeRpcMethod Overload [2116]

TwinCAT.TypeSystem Namespace [1622]


Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
int TryInvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    out Object[] outParameters,
    out Object retValue
)

Parameters

methodName Type: System.String
The method name.
inParameters Type: System.Object
The parameters.
outParameters Type: System.Object
The out parameters.
retValue Type: System.Object
The return value of the RPC method as object.

Return Value

Type: Int32
The result value of the call (ErrorCode). 0 means Succeeded.
Remarks

Because this overload doesn't provide any AnyTypeSpecifier [1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```csharp
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AdsAddress address = ArgParser.Parse(args);
        using (AdsClient client = new AdsClient())
        {
            //client.Synchronize = false;
            // Connect to the target device
            client.Connect(address);
            SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
            ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
            // Get the Symbols (Dynamic Symbols)
            IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Instance of the PLC Program
            // Call a Method that has the following signature (within MAIN Program)
            /*
             * {attribute 'TcRpcEnable'}
             * METHOD PUBLIC M_Add : INT
             * VAR_INPUT
             *     i1 : INT := 0;
             *     i2 : INT := 0;
             * END_VAR
             */
            short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});
            // Call a Method that has no parameter and returns VOID
            main.InvokeRpcMethod("M_Method1", new object[] {});
            // Browsing RpcMethods
            foreach(IRpcMethod method in main.RpcMethods)
            {
                string method_Name = method.Name;
                foreach(IRpcMethodParameter parameter in method.Parameters)
                {
                    string parameter_Name = parameter.Name;
                    string parameter_Type = parameter.TypeName;
                }
            }
        }
    }
}
```

Reference

<table>
<thead>
<tr>
<th>IRpcCallableInstance Interface [2104]</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryInvokeRpcMethod Overload [2116]</td>
</tr>
<tr>
<td>TwinCAT.TypeSystem Namespace [1622]</td>
</tr>
</tbody>
</table>
IRpcCallableInstance.TryInvokeRpcMethod Method
(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object., Object.)

Invokes the specified RpcMethod of the IRpcCallableInstance [2104].

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
int TryInvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

**methodName**
Type: System.String
Name of the method.

**inParameters**
Type: .System.Object.
The parameters.

**outSpecifiers**
Type: TwinCAT.TypeSystem.AnyTypeSpecifier [1633]
The out specifiers (specifying the out types) or NULL.

**retSpecifier**
Type: TwinCAT.TypeSystem.AnyTypeSpecifier [1633]
The ret specifier (specifying the return value) or NULL.

**outParameters**
Type: .System.Object.
The out parameters.

**retValue**
Type: System.Object.
The return value of the RPC method./>

Return Value

Type: Int32 AdsErrorCode.

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

**Dynamic Tree Mode**

class RpcCall1VirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    
}
using (AdsClient client = new AdsClient())
{
    //client.Synchronize = false;
    // Connect to the target device
    client.Connect(address);
    SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolLoadMode.VirtualTree);
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);
    // Get the Symbols (Dynamic Symbols)
    IRpcStructInstance main = (IRpcStructInstance)loader.Symbols("MAIN"); // Gets the MAIN Instance of the PLC Program
    short result = (short)main.InvokeRpcMethod("M_Add", new object[]{(short)3, (short)4});
    // Call a Method that has no parameter and returns VOID
    main.InvokeRpcMethod("M_Method1", new object[]{ });
    // Browsing RpcMethods
    foreach (IRpcMethod method in main.RpcMethods)
    {
        string methodName = method.Name;
        foreach (IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
    }
}

Reference
IRpcCallableInstance Interface [2104]
TryInvokeRpcMethod Overload [2116]
TwinCAT.TypeSystem Namespace [1622]

6.11.72 IRpcCallableType Interface

Interface representing an RPC callable IStructType [2162]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
public interface IRpcCallableType

The IRpcCallableType type exposes the following members.
## TwinCAT.Ads Namespaces

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RpcMethods</td>
<td>Gets the Method descriptions for the IRpcCallableType</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]

### 6.11.72.1 IRpcCallableType Properties

The IRpcCallableType [2121] type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RpcMethods</td>
<td>Gets the Method descriptions for the IRpcCallableType</td>
</tr>
</tbody>
</table>

### Reference

IRpcCallableType Interface [2121]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.72.1.1 IRpcCallableType.RpcMethods Property

Gets the Method descriptions for the IRpcCallableType [2121]

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
IRpcMethodCollection RpcMethods { get; }
```

### Property Value

Type: IRpcMethodCollection [2127]

The RPC methods.

### Remarks

The DataType (Structure) must be marked with the PlcAttribute 'TcRpcEnable' to enable RpcMethods, otherwise RpcMethods are not passed through to the ADS symbolic information.

### Reference

IRpcCallableType Interface [2121]

TwinCAT.TypeSystem Namespace [1622]
## IRpcMethod Interface

Interface describes an RPC Method

**Namespace:**  TwinCAT.TypeSystem

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public interface IRpcMethod
```

The IRpcMethod type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comment</td>
<td>Gets the Method comment.</td>
</tr>
<tr>
<td>InParameters</td>
<td>Gets the In-Parameters of the IRpcMethod</td>
</tr>
<tr>
<td>IsVoid</td>
<td>Gets a value indicating whether this IRpcMethod has no return parameter</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the method</td>
</tr>
<tr>
<td>OutParameters</td>
<td>Gets the Out-Parameters of the IRpcMethod</td>
</tr>
<tr>
<td>Parameters</td>
<td>Gets all parameters (In, Out and ref parameters) of the IRpcMethod</td>
</tr>
<tr>
<td>ReturnType</td>
<td>Gets the return type.</td>
</tr>
<tr>
<td>ReturnTypeSize</td>
<td>Gets the size of the return type in bytes.</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace

## IRpcMethod Properties

The IRpcMethod type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comment</td>
<td>Gets the Method comment.</td>
</tr>
<tr>
<td>InParameters</td>
<td>Gets the In-Parameters of the IRpcMethod</td>
</tr>
<tr>
<td>IsVoid</td>
<td>Gets a value indicating whether this IRpcMethod has no return parameter</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the method</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OutParameters</td>
<td>Gets the Out-Parameters of the IRpcMethod.</td>
</tr>
<tr>
<td>Parameters</td>
<td>Gets all parameters (In, Out and ref parameters) of the IRpcMethod.</td>
</tr>
<tr>
<td>ReturnType</td>
<td>Gets the return type.</td>
</tr>
<tr>
<td>ReturnTypeSize</td>
<td>Gets the size of the return type in bytes.</td>
</tr>
</tbody>
</table>

### Reference

IRpcMethod Interface [2123]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.73.1.1 IRpcMethod.Comment Property

Gets the Method comment.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
string Comment { get; }
```

**Property Value**

Type: `String`

The comment.

**Reference**

IRpcMethod Interface [2123]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.73.1.2 IRpcMethod.InParameters Property

Gets the In-Parameters of the IRpcMethod.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
IRpcMethodParameterCollection InParameters { get; }
```
Property Value
Type: IRpcMethodParameterCollection
The In- and Ref-Parameters

Reference
IRpcMethod Interface
TwinCAT.TypeSystem Namespace

6.11.73.1.3 IRpcMethod.IsVoid Property
Gets a value indicating whether this IRpcMethod has no return parameter

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
bool IsVoid { get; }

Property Value
Type: Boolean
true if this instance is void; otherwise, false.

Reference
IRpcMethod Interface
TwinCAT.TypeSystem Namespace

6.11.73.1.4 IRpcMethod.Name Property
Gets the name of the method

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
string Name { get; }

Property Value
Type: String
The name.

Reference
IRpcMethod Interface
TwinCAT.TypeSystem Namespace
6.11.73.1.5 IRpcMethod.OutParameters Property

Gets the Out-Parameters of the IRpcMethod [2123]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

IRpcMethodParameterCollection OutParameters { get; }

Property Value

Type: IRpcMethodParameterCollection [2137]
The out- and ref-parameters.

Remarks

The Out-Parameters doesn't include the Ref Parameters. These are included in the InParameters [2124] set.

Reference

IRpcMethod Interface [2123]
TwinCAT.TypeSystem Namespace [1622]

6.11.73.1.6 IRpcMethod.Parameters Property

Gets all parameters (In, Out and ref parameters) of the IRpcMethod [2123]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

IRpcMethodParameterCollection Parameters { get; }

Property Value

Type: IRpcMethodParameterCollection [2137]
The parameters.

Reference

IRpcMethod Interface [2123]
TwinCAT.TypeSystem Namespace [1622]

6.11.73.1.7 IRpcMethod.ReturnType Property

Gets the return type.
6.11.73.1.8 IRpcMethod.ReturnTypeSize Property

Gets the size of the return type in bytes.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
int ReturnTypeSize { get; }
```

**Property Value**

Type: Int32
The size of the return type.

**Reference**

IRpcMethod Interface

TwinCAT.TypeSystem Namespace

6.11.74 IRpcMethodCollection Interface

Interface for RPC Method collections.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public interface IRpcMethodCollection : IList<IRpcMethod>, ICollection<IRpcMethod>, IEnumerable<IRpcMethod>, IEnumerable
```
The IRpcMethodCollection type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection&lt;T&gt;. (Inherited from ICollection&lt;IRpcMethod&gt;[2123]..)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection&lt;T&gt; is read-only. (Inherited from ICollection&lt;IRpcMethod&gt;[2123]..)</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the element at the specified index. (Inherited from IList&lt;IRpcMethod&gt;[2123]..)</td>
</tr>
<tr>
<td>Item.String_[2129]</td>
<td>Gets the IRpcMethod[2123] with the specified method name.</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection&lt;T&gt;. (Inherited from ICollection&lt;IRpcMethod&gt;[2123]..)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection&lt;T&gt;. (Inherited from ICollection&lt;IRpcMethod&gt;[2123]..)</td>
</tr>
<tr>
<td>Contains(String)[2131]</td>
<td>Determines whether this collection contains the specified method name.</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection&lt;T&gt; contains a specific value. (Inherited from ICollection&lt;IRpcMethod&gt;[2123]..)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection&lt;T&gt; to an Array, starting at a particular Array index. (Inherited from ICollection&lt;IRpcMethod&gt;[2123]..)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable&lt;IRpcMethod&gt;[2123]..)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList&lt;T&gt;. (Inherited from IList&lt;IRpcMethod&gt;[2123]..)</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList&lt;T&gt; at the specified index. (Inherited from IList&lt;IRpcMethod&gt;[2123]..)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection&lt;T&gt;. (Inherited from ICollection&lt;IRpcMethod&gt;[2123]..)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IList&lt;T&gt; item at the specified index. (Inherited from IList&lt;IRpcMethod&gt;[2123]..)</td>
</tr>
<tr>
<td>TryGetMethod(Int32 , IRpcMethod) [2132]</td>
<td>Tries to get the specified method.</td>
</tr>
<tr>
<td>TryGetMethod(String , IRpcMethod) [2133]</td>
<td>Tries to get the specified method.</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]
6.11.74.1  IRpcMethodCollection Properties

The IRpcMethodCollection type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection. (Inherited from ICollection. IRpcMethod..)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection is read-only. (Inherited from ICollection. IRpcMethod..)</td>
</tr>
<tr>
<td>Item.Int32.</td>
<td>Gets or sets the element at the specified index. (Inherited from IList. IRpcMethod..)</td>
</tr>
<tr>
<td>Item.String.</td>
<td>Gets the IRpcMethod with the specified method name.</td>
</tr>
</tbody>
</table>

Reference

IRpcMethodCollection Interface

TwinCAT.TypeSystem Namespace

6.11.74.1.1  IRpcMethodCollection.Item Property

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32.</td>
<td>Gets or sets the element at the specified index. (Inherited from IList. IRpcMethod..)</td>
</tr>
<tr>
<td>Item.String.</td>
<td>Gets the IRpcMethod with the specified method name.</td>
</tr>
</tbody>
</table>

Reference

IRpcMethodCollection Interface

TwinCAT.TypeSystem Namespace

IRpcMethodCollection.Item Property (String)

Gets the IRpcMethod with the specified method name.

Namespace:  TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e7bac0ea15da1c14

Syntax

C#  
IRpcMethod this[string methodName] { get; }
Parameters

methodName  
Type: System.String
Name of the method.

Return Value

Type: IRpcMethod [2123]
RpcMethod.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyNotFoundException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

IRpcMethodCollection Interface [2127]

Item Overload [2129]

TwinCAT.TypeSystem Namespace [1622]

6.11.74.2  IRpcMethodCollection Methods

The IRpcMethodCollection [2127] type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection&lt;T&gt;. (Inherited from ICollection&lt;IRpcMethod&gt; [2123].)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection&lt;T&gt;. (Inherited from ICollection&lt;IRpcMethod&gt; [2123].)</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether this collection contains the specified method name.</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection&lt;T&gt; contains a specific value. (Inherited from ICollection&lt;IRpcMethod&gt; [2123].)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection&lt;T&gt; to an Array, starting at a particular Array index. (Inherited from ICollection&lt;IRpcMethod&gt; [2123].)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable&lt;IRpcMethod&gt; [2123].)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList&lt;T&gt;. (Inherited from IList&lt;IRpcMethod&gt; [2123].)</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList&lt;T&gt; at the specified index. (Inherited from IList&lt;IRpcMethod&gt; [2123].)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection&lt;T&gt;. (Inherited from ICollection&lt;IRpcMethod&gt; [2123].)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IList&lt;T&gt; item at the specified index. (Inherited from IList&lt;IRpcMethod&gt; [2123].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>TryGetMethod(Int32, IRpcMethod.) [2132]</td>
<td>Tries to get the specified method.</td>
</tr>
<tr>
<td>TryGetMethod(String, IRpcMethod.) [2133]</td>
<td>Tries to get the specified method.</td>
</tr>
</tbody>
</table>

**Reference**

IRpcMethodCollection Interface [2127]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.74.2.1 IRpcMethodCollection.Contains Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String) [2131]</td>
<td>Determines whether this collection contains the specified method name.</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection&lt;T&gt; contains a specific value. (Inherited from ICollection&lt;T&gt;.)</td>
</tr>
</tbody>
</table>

**Reference**

IRpcMethodCollection Interface [2127]
TwinCAT.TypeSystem Namespace [1622]

### IRpcMethodCollection.Contains Method (String)

Determines whether this collection contains the specified method name.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
bool Contains(
    string methodName
)
```

**Parameters**

- **methodName**
  - Type: System.String
  - Name of the method.
**Return Value**

Type: Boolean
true if contained.; otherwise, false.

**Reference**

IRpcMethodCollection Interface [2127]
Contains Overload [2131]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.74.2.2 IRpcMethodCollection.TryGetMethod Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="2132" alt="TryGetMethod(Int32, IRpcMethod.)" /></td>
<td>Tries to get the specified method.</td>
</tr>
<tr>
<td><img src="2133" alt="TryGetMethod(String, IRpcMethod.)" /></td>
<td>Tries to get the specified method.</td>
</tr>
</tbody>
</table>

**Reference**

IRpcMethodCollection Interface [2127]
TwinCAT.TypeSystem Namespace [1622]

**IRpcMethodCollection.TryGetMethod Method (Int32, IRpcMethod.)**

Tries to get the specified method.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

*C#*

```csharp
bool TryGetMethod(
    int vTableIndex,
    out IRpcMethod method
)
```

**Parameters**

- **vTableIndex**
  Type: System.Int32
  vTableIndex.

- **method**
  Type: TwinCAT.TypeSystem.IRpcMethod [2123].
  The method if found, NULL otherwise.
**Return Value**

Type: Boolean
ture if found, false otherwise.

**Reference**

IRpcMethodCollection Interface [› 2127]

TryGetMethod Overload [› 2132]

TwinCAT.TypeSystem Namespace [› 1622]

**IRpcMethodCollection.TryGetMethod Method (String, IRpcMethod.)**

Tries to get the specified method.

**Namespace:** TwinCAT.TypeSystem [› 1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
bool TryGetMethod(  
    string methodName,  
    out IRpcMethod method  
)
```

**Parameters**

- **methodName**
  
  Type: System.String  
  Name of the method.

- **method**
  
  Type: TwinCAT.TypeSystem.IRpcMethod [› 2123].  
  The method if fund, NULL otherwise.

**Return Value**

Type: Boolean  
true if found, false otherwise.

**Reference**

IRpcMethodCollection Interface [› 2127]

TryGetMethod Overload [› 2132]

TwinCAT.TypeSystem Namespace [› 1622]

### 6.11.75 IRpcMethodParameter Interface

Interface IRpcMethodParameter

**Namespace:** TwinCAT.TypeSystem [› 1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

public interface IRpcMethodParameter

The IRpcMethodParameter type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HasLengthIsParameter [2135]</td>
<td>Gets a value indicating whether this instance has a related LengthIs Parameter.</td>
</tr>
<tr>
<td>LengthIsParameterIndex [2135]</td>
<td>Gets the index of the LengthIs parameter (within the MethodParameter List)</td>
</tr>
<tr>
<td>Name [2135]</td>
<td>Gets the Parameter Name</td>
</tr>
<tr>
<td>ParameterFlags [2136]</td>
<td>Gets the parameter flags.</td>
</tr>
<tr>
<td>Size [2136]</td>
<td>Gets the size of the IRpcMethodParameter</td>
</tr>
<tr>
<td>TypeName [2137]</td>
<td>Gets the Data type of the Parameter</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]

6.11.75.1 IRpcMethodParameter Properties

The IRpcMethodParameter [2133] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HasLengthIsParameter [2135]</td>
<td>Gets a value indicating whether this instance has a related LengthIs Parameter.</td>
</tr>
<tr>
<td>LengthIsParameterIndex [2135]</td>
<td>Gets the index of the LengthIs parameter (within the MethodParameter List)</td>
</tr>
<tr>
<td>Name [2135]</td>
<td>Gets the Parameter Name</td>
</tr>
<tr>
<td>ParameterFlags [2136]</td>
<td>Gets the parameter flags.</td>
</tr>
<tr>
<td>Size [2136]</td>
<td>Gets the size of the IRpcMethodParameter</td>
</tr>
<tr>
<td>TypeName [2137]</td>
<td>Gets the Data type of the Parameter</td>
</tr>
</tbody>
</table>

Reference

IRpcMethodParameter Interface [2133]

TwinCAT.TypeSystem Namespace [1622]
6.11.75.1.1 IRpcMethodParameter.HasLengthIsParameter Property

Gets a value indicating whether this instance has a related LengthIs Parameter.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
bool HasLengthIsParameter { get; }
```

**Property Value**

Type: Boolean

true if this instance has a LengthIs parameter; otherwise, false.

**Reference**

IRpcMethodParameter Interface

TwinCAT.TypeSystem Namespace

6.11.75.1.2 IRpcMethodParameter.LengthIsParameterIndex Property

Gets the index of the LengthIs parameter (within the MethodParameter List)

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
int LengthIsParameterIndex { get; }
```

**Property Value**

Type: Int32

The index of the length is parameter.

**Remarks**

This field references to the Parameter that defines the length for this generic one. Equally to the marshalling attributes of COM (sizeof, length) this enables to transport parameter of type (PVOID)

**Reference**

IRpcMethodParameter Interface

TwinCAT.TypeSystem Namespace

6.11.75.1.3 IRpcMethodParameter.Name Property

Gets the Parameter Name
namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

string Name { get; }

Property Value

Type: String
The name.

Reference

IRpcMethodParameter Interface [2133]
TwinCAT.TypeSystem Namespace [1622]

6.11.75.1.4 IRpcMethodParameter.ParameterFlags Property

Gets the parameter flags.

namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

MethodParamFlags ParameterFlags { get; }

Property Value

Type: MethodParamFlags [2288]
The parameter flags.

Reference

IRpcMethodParameter Interface [2133]
TwinCAT.TypeSystem Namespace [1622]

6.11.75.1.5 IRpcMethodParameter.Size Property

Gets the size of the IRpcMethodParameter [2133]

namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

int Size { get; }
6.11.75.1.6 IRpcMethodParameter.TypeName Property

Gets the Data type of the Parameter

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
string TypeName { get; }

Property Value

Type: String
The type.

Reference

IRpcMethodParameter Interface [2133]
TwinCAT.TypeSystem Namespace [1622]

6.11.76 IRpcMethodParameterCollection Interface

Interface IRpcMethodParameterCollection

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public interface IRpcMethodParameterCollection : IList<IRpcMethodParameter>, ICollection<IRpcMethodParameter>, IEnumerable<IRpcMethodParameter>, IEnumerable

The IRpcMethodParameterCollection type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T. (Inherited from ICollection&lt;IRpcMethodParameter&gt; [2133]..)</td>
</tr>
</tbody>
</table>
TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IRpcMethodParameter [2133].)</td>
</tr>
<tr>
<td>Item</td>
<td>Gets or sets the element at the specified index. (Inherited from IList.IRpcMethodParameter [2133].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection.T. (Inherited from ICollection.IRpcMethodParameter [2133].)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T. (Inherited from ICollection.IRpcMethodParameter [2133].)</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IRpcMethodParameter [2133].)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IRpcMethodParameter [2133].)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IRpcMethodParameter [2133].)</td>
</tr>
<tr>
<td>GetLengthIsParameter</td>
<td>Gets the corresponding LengthIs parameter.</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList.T. (Inherited from IList.IRpcMethodParameter [2133].)</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList.T. at the specified index. (Inherited from IList.IRpcMethodParameter [2133].)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection.T. (Inherited from ICollection.IRpcMethodParameter [2133].)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IList.T. item at the specified index. (Inherited from IList.IRpcMethodParameter [2133].)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]

6.11.76.1 IRpcMethodParameterCollection Properties

The IRpcMethodParameterCollection [2137] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T. (Inherited from ICollection.IRpcMethodParameter [2133].)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IRpcMethodParameter [2133].)</td>
</tr>
<tr>
<td>Item</td>
<td>Gets or sets the element at the specified index. (Inherited from IList.IRpcMethodParameter [2133].)</td>
</tr>
</tbody>
</table>

Reference

IRpcMethodParameterCollection Interface [2137]
## 6.11.76.2 IRpcMethodParameterCollection Methods

The `IRpcMethodParameterCollection` type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the <code>ICollection&lt;T&gt;</code> (Inherited from <code>ICollection&lt;IRpcMethodParameter&gt;</code>..)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the <code>ICollection&lt;T&gt;</code> (Inherited from <code>ICollection&lt;IRpcMethodParameter&gt;</code>..)</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the <code>ICollection&lt;T&gt;</code> contains a specific value. (Inherited from <code>ICollection&lt;IRpcMethodParameter&gt;</code>..)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the <code>ICollection&lt;T&gt;</code> to an <code>Array</code>, starting at a particular <code>Array</code> index. (Inherited from <code>ICollection&lt;IRpcMethodParameter&gt;</code>..)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from <code>IEnumerable&lt;IRpcMethodParameter&gt;</code>..)</td>
</tr>
<tr>
<td>GetLengthIsParameter</td>
<td>Gets the corresponding LengthIs parameter.</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the <code>IList&lt;T&gt;</code> (Inherited from <code>IList&lt;IRpcMethodParameter&gt;</code>..)</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the <code>IList&lt;T&gt;</code> at the specified index. (Inherited from <code>IList&lt;IRpcMethodParameter&gt;</code>..)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the <code>ICollection&lt;T&gt;</code> (Inherited from <code>ICollection&lt;IRpcMethodParameter&gt;</code>..)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the <code>IList&lt;T&gt;</code> item at the specified index. (Inherited from <code>IList&lt;IRpcMethodParameter&gt;</code>..)</td>
</tr>
</tbody>
</table>

### Reference

`IRpcMethodParameterCollection Interface` [2137]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.76.2.1 IRpcMethodParameterCollection.GetLengthIsParameter Method

Gets the corresponding LengthIs parameter.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
IRpcMethodParameter GetLengthIsParameter(
    IRpcMethodParameter parameter
)
```
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| parameter | Type: TwinCAT>TypeSystem.IRpcMethodParameter\[2133\]  
The value parameter |

Return Value

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRpcMethodParameter[2133]</td>
<td>The LengthIs Parameter</td>
</tr>
</tbody>
</table>

Reference

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRpcMethodParameterCollection Interface[2137]</td>
<td></td>
</tr>
<tr>
<td>TwinCAT&gt;TypeSystem Namespace[1622]</td>
<td></td>
</tr>
<tr>
<td>IRpcMethodParameter.LengthIsParameterIndex[2135]</td>
<td></td>
</tr>
<tr>
<td>IRpcMethodParameter.HasLengthIsParameter[2135]</td>
<td></td>
</tr>
</tbody>
</table>

6.11.77 IRpcStructInstance Interface

Interface IRpcStructInstance

Namespace: TwinCAT>TypeSystem\[1622\]  
Assembly: TwinCAT>Ads>Abstractions (in TwinCAT>Ads>Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IRpcStructInstance : IStructInstance,  
ISymbol, IAttributedInstance, IInstance, IBitSize, IRpcCallableInstance
```

The IRpcStructInstance type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes[1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance[1980].)</td>
</tr>
<tr>
<td>BitSize[1984]</td>
<td>Gets the size of the IDataType[1986] in bits. (Inherited from IBitSize[1982].)</td>
</tr>
<tr>
<td>ByteSize[1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize[1982].)</td>
</tr>
<tr>
<td>Category[2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol[2176].)</td>
</tr>
<tr>
<td>Comment[2053]</td>
<td>Gets the comment of the IInstance[2052] (Inherited from IInstance[2052].)</td>
</tr>
<tr>
<td>DataType[2054]</td>
<td>Gets the IDataType[1986] of the IInstance[2052]. (Inherited from IInstance[2052].)</td>
</tr>
<tr>
<td>HasRpcMethods[2161]</td>
<td>Gets a value indicating whether this instance has RPC methods (Inherited from IStructInstance[2158].)</td>
</tr>
<tr>
<td>InstanceName[2054]</td>
<td>Gets the name of the instance (without periods .) (Inherited from IInstance[2052].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)). (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0). (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this [ISymbol [2176] is persistent. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the Instance [2052] represents a Pointer type (Pointer TO) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE TO) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this Instance [2052] is static. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>MemberInstances [2162]</td>
<td>Gets the member instances of the Struct Instance [2158]. (Inherited from StructInstance [2158].)</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>RpcMethods [2105]</td>
<td>Gets the Method descriptions for the IRpcCallableType [2121] (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052]. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvokeRpcMethod(S string, Object) [2107]</td>
<td>Invokes the specified RPC Method (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>InvokeRpcMethod(S string, Object, Object) [2109]</td>
<td>Invokes the specified RPC Method (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, Object, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, Object, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, Object, Object, Object)</td>
<td>Invokes the specified RPC Method (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object)</td>
<td>Invokes the the specified RpcMethod of the IRpcCallableInstance [2104]. (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]
TwinCAT.TypeSystem.IStructInstance [2158]
TwinCAT.TypeSystem.IRpcCallableInstance [2104]

**6.11.77.1 IRpcStructInstance Properties**

The IRpcStructInstance [2140] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the IInstance [2052] (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>HasRpcMethods [2161]</td>
<td>Gets a value indicating whether this instance has RPC methods (Inherited from IStructInstance [2158].)</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this IInstance [2052] is static. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>MemberInstances [2162]</td>
<td>Gets the member instances of the Struct Instance [2158]. (Inherited from IStructInstance [2158].)</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>RpcMethods [2105]</td>
<td>Gets the Method descriptions for the IRpcCallableType [2121] (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### Name | Description
--- | ---
ValueEncoding | Gets the value encoding. (Inherited from IAttributedInstance [1980].)

### Reference
- IRpcStructInstance Interface [2140]
- TwinCAT.TypeSystem Namespace [1622]

### 6.11.77.2 IRpcStructInstance Methods

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvokeRpcMethod(String, Object.)</td>
<td>Invokes the specified RPC Method (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, Object., Object.)</td>
<td>Invokes the specified RPC Method (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>InvokeRpcMethod(String, Object., AnyTypeSpecifier., AnyTypeSpecifier, Object.)</td>
<td>Invokes the specified RPC Method (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, Object., CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>InvokeRpcMethodAsync(String, Object., AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken)</td>
<td>Invokes the specified RPC Method asynchronously (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, Object., Object.)</td>
<td>Invokes the specified RPC Method (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
<tr>
<td>TryInvokeRpcMethod(String, Object., Object.)</td>
<td>Invokes the specified RPC Method (Inherited from IRpcCallableInstance [2104].)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### TryInvokeRpcMethod

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryInvokeRpcMethod(String, Object, AnyTypeSpecifier, AnyTypeSpecifier, Object, Object)</td>
<td>Invokes the the specified RpcMethod of the IRpcCallableInstance. (Inherited from IRpcCallableInstance.)</td>
</tr>
</tbody>
</table>

### Reference

- IRpcStructInstance Interface [2140]
- TwinCAT.TypeSystem Namespace [1622]

### 6.11.78 IStringInstance Interface

Interface IStringInstance

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)  
Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public interface IStringInstance : ISymbol, IAttributedInstance, IInstance, IBitSize
```

The IStringInstance type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the IInstance [2052] (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>DataType [2053]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods .) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsFixedSize [2148]</td>
<td>Gets a value indicating whether this instance is a string of static length</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE</td>
</tr>
<tr>
<td></td>
<td>TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this IInstance [2052] is static. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from IInstance [2052].)</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsByteAligned [</td>
</tr>
<tr>
<td></td>
<td>1985] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052].</td>
</tr>
<tr>
<td></td>
<td>(Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]
TwinCAT.TypeSystem.ISymbol [2176]
TwinCAT.TypeSystem.IStringType [2155]

### 6.11.78.1 IStringInstance Properties

The IStringInstance [2145] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [</td>
</tr>
<tr>
<td></td>
<td>1982].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the Instance</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from Instance</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from Instance</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol</td>
</tr>
<tr>
<td>IsFixedSize</td>
<td>Gets a value indicating whether this instance is a string of static length</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this ISymbol</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the Instance</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the Instance</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this Instance</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol (Inherited from ISymbol</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType</td>
</tr>
<tr>
<td>SubSymbols</td>
<td>Gets the SubSymbols of the ISymbol</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the IDataType</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance</td>
</tr>
</tbody>
</table>

**Reference**

IStringInstance Interface | [2145]
6.11.78.1.1  IString Instance.IsFixedLength Property

Gets a value indicating whether this instance is a string of static length

**Namespace:**  TwinCAT.TypeSystem

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
bool IsFixedLength { get; }
```

**Property Value**

Type: Boolean  

true if this instance is of static length; otherwise, false.

**Reference**

IString Instance Interface  

IString Marshaler Interface

6.11.79  IStringMarshaler Interface

Common interface for marshalling ADS string values.

**Namespace:**  TwinCAT.TypeSystem

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public interface IStringMarshaler
```

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marshal(String, Span)</td>
<td>Gets the marshal size of the string.</td>
</tr>
<tr>
<td>Marshal(IString Type, String, Span)</td>
<td>Gets the marshal size of the specified string type.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MarshalSize(.Encoding, Int32)</td>
<td>Gets the marshal size of the string given by its length.</td>
</tr>
<tr>
<td>Marshal(ReadOnlySpan, Void, Byte)</td>
<td></td>
</tr>
<tr>
<td>Marshal(IStringType, String, Span)</td>
<td></td>
</tr>
<tr>
<td>MarshalSize(String)</td>
<td>Gets the marshal size of the string.</td>
</tr>
<tr>
<td>MarshalSize(IStringType)</td>
<td>Gets the marshal size of the specified string type.</td>
</tr>
<tr>
<td>MarshalSize(Encoding, Int32)</td>
<td>Gets the marshal size of the string given by its length.</td>
</tr>
<tr>
<td>Unmarshal(ReadOnlySpan, Void, Byte)</td>
<td></td>
</tr>
<tr>
<td>Unmarshal(IStringType, ReadOnlySpan, Void)</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

### 6.11.79.1 IStringMarshaler Methods

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marshal(String, Span)</td>
<td></td>
</tr>
<tr>
<td>Marshal(IStringType, String, Span)</td>
<td></td>
</tr>
<tr>
<td>MarshalSize(String)</td>
<td>Gets the marshal size of the string.</td>
</tr>
<tr>
<td>MarshalSize(IStringType)</td>
<td>Gets the marshal size of the specified string type.</td>
</tr>
<tr>
<td>MarshalSize(Encoding, Int32)</td>
<td>Gets the marshal size of the string given by its length.</td>
</tr>
<tr>
<td>Unmarshal(ReadOnlySpan, Void, Byte)</td>
<td></td>
</tr>
<tr>
<td>Unmarshal(IStringType, ReadOnlySpan, Void)</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

IStringMarshalAser Interface [2148]

TwinCAT.TypeSystem Namespace [1622]
6.11.79.1.1  IStringMarshaler.Marshal Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marshal(String, Span)</td>
<td></td>
</tr>
<tr>
<td>Marshal(ISourceType, String, Span)</td>
<td></td>
</tr>
</tbody>
</table>

Reference

IStringMarshaler Interface [p. 2148]
TwinCAT.TypeSystem Namespace [p. 1622]

IStringMarshaler.Marshal Method (String, Span)`1

Namespace:  TwinCAT.TypeSystem [p. 1622]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
int Marshal(
    string value,
    Span destination
)
```

Parameters

value            Type: System.String
destination      Type: Span

Return Value

Type: Int32

Reference

IStringMarshaler Interface [p. 2148]
Marshal Overload [p. 2150]
TwinCAT.TypeSystem Namespace [p. 1622]

IStringMarshaler.Marshal Method (ISourceType, String, Span)`1

Namespace:  TwinCAT.TypeSystem [p. 1622]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
int Marshal(
    IStringType type,
    string value,
    Span destination
)
```

Parameters

type Type: TwinCAT.TypeSystem.IStringType

value Type: System.String

destination Type: Span

Return Value

Type: Int32

Reference

IStringMarshaler Interface

Marshal Overload

TwinCAT.TypeSystem Namespace

6.11.79.1.2 IStringMarshaler.MarshalSize Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MarshalSize(String)</td>
<td>Gets the marshal size of the string.</td>
</tr>
<tr>
<td>MarshalSize(IStringType)</td>
<td>Gets the marshal size of the specified string type.</td>
</tr>
<tr>
<td>MarshalSize(Encoding, Int32)</td>
<td>Gets the marshal size of the string given by its length.</td>
</tr>
</tbody>
</table>

Reference

IStringMarshaler Interface

TwinCAT.TypeSystem Namespace

IStringMarshaler.MarshalSize Method (String)

Gets the marshal size of the string.
Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
int MarshalSize(
    string value
)
```

Parameters

value
Type: System.String
The string value.

Return Value

Type: Int32
Marshalling size of the string.

Reference

IStringMarshaler Interface [2148]
MarshalSize Overload [2151]
TwinCAT.TypeSystem Namespace [1622]

IStringMarshaler.MarshalSize Method (IStringType)

Gets the marshal size of the specified string type.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
int MarshalSize(
    IStringType stringType
)
```

Parameters

stringType
Type: TwinCAT.TypeSystem.IStringType [2155]
Type of the string.

Return Value

Type: Int32
Marshalling size of the string

Reference

IStringMarshaler Interface [2148]
MarshalSize Overload [2151]
IStringMarshaler.MarshalSize Method (Encoding, Int32)

Gets the marshal size of the string given by its length.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
int MarshalSize(
    Encoding encoding,
    int strLen
)
```

**Parameters**

- `encoding`: Type: System.Text.Encoding
  The encoding.
- `strLen`: Type: System.Int32
  Length of the string.

**Return Value**

Type: Int32
Marshalling size of the string.

**Reference**

IStringMarshaler Interface

MarshalSize Overload

TwinCAT.TypeSystem Namespace

### 6.11.79.1.3 IStringMarshaler.Unmarshal Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Unmarshal(ReadOnlySpan, Void, Byte)](Unmarshal(ReadOnlySpan, Void, Byte)[2154])</td>
<td></td>
</tr>
<tr>
<td>![Unmarshal(IStringType, ReadOnlySpan, Void)](Unmarshal(IStringType, ReadOnlySpan, Void)[2154])</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

IStringMarshaler Interface

TwinCAT.TypeSystem Namespace
IStringMarshaler.Unmarshal Method (ReadOnlySpan`1, Void, Byte)

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
int Unmarshal(
    ReadOnlySpan source,
    void encoding,
    byte value
)
```

**Parameters**

- `source` Type: `ReadOnlySpan`
- `encoding` Type: `System.Void`
- `value` Type: `System.Byte`

**Return Value**

Type: `Int32`

**Reference**

- [IStringMarshaler Interface](#)
- [Unmarshal Overload](#)
- [TwinCAT.TypeSystem Namespace](#)

IStringMarshaler.Unmarshal Method (IStringType, ReadOnlySpan`1, Void)

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
int Unmarshal(
    IStringType stringType,
    ReadOnlySpan source,
    void value
)
```

**Parameters**

- `stringType` Type: `TwinCAT.TypeSystem.IStringType`
- `source` Type: `ReadOnlySpan`
value Type: System.Void

Return Value
Type: Int32

Reference
IStringMarshaler Interface [ ], 2148
Unmarshal Overload [ ], 2153
TwinCAT.TypeSystem Namespace [ ], 1622

6.11.80 IStringType Interface

Interface representing a string [IDataType ], 1986

Namespace: TwinCAT.TypeSystem [ ], 1622
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
public interface IStringType : IDataType, IBitSize

The IStringType type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the [IDataType ], 1986 (Inherited from [IDataType ], 1986).</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the [IDataType ], 1986 in bits. (Inherited from [IBitSize ], 1982).</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from [IBitSize ], 1982).</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from [IDataType ], 1986).</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment behind the variable declaration. (Inherited from [IDataType ], 1986).</td>
</tr>
<tr>
<td>Encoding</td>
<td>Gets the encoding of the String (Encoding.Default (STRING) or Encoding.UNICODE (WSTRING)).</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the [IDataType ], 1986 (Namespace + Name) (Inherited from [IDataType ], 1986).</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the DataType (Inherited from [IDataType ], 1986).</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from [IBitSize ], 1982).</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from [IBitSize ], 1982).</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this [IDataType] is a container type (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>IsFixedLength</td>
<td>Gets a value indicating whether the string is of fixed length.</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this [IDataType] is a pointer type (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this [IDataType] is primitive (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this [IDataType] is a reference type (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>Length</td>
<td>Gets the number of characters within the string (when fixed length).</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the [IDataType] exists. (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType (Inherited from [IBitSize].)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

**6.11.80.1 IStringType Properties**

The IStringType [2155] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the [IDataType] (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the [IDataType] in bits. (Inherited from [IBitSize].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from [IBitSize].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment behind the variable declaration. (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>Encoding</td>
<td>Gets the encoding of the String (Encoding.Default (STRING) or Encoding.UNICODE (WSTRING))</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the [IDataType] (Namespace + Name) (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the Data Type (Inherited from [IDataType].)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from [IBitSize].)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from [IBitSize].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsContainer [1990]</td>
<td>Gets a value indicating whether this [DataType [1986] is a container type</td>
</tr>
<tr>
<td></td>
<td>(Inherited from [DataType [1986].)</td>
</tr>
<tr>
<td>IsFixedLength [2157]</td>
<td>Gets a value indicating whether the string is of fixed length.</td>
</tr>
<tr>
<td>IsPointer [1990]</td>
<td>Gets a value indicating whether this [DataType [1986] is a pointer type</td>
</tr>
<tr>
<td></td>
<td>(Inherited from [DataType [1986].)</td>
</tr>
<tr>
<td>IsPrimitive [1991]</td>
<td>Gets a value indicating whether this [DataType [1986] is primitive</td>
</tr>
<tr>
<td></td>
<td>(Inherited from [DataType [1986].)</td>
</tr>
<tr>
<td>IsReference [1991]</td>
<td>Gets a value indicating whether this [DataType [1986] is a reference type</td>
</tr>
<tr>
<td></td>
<td>(Inherited from [DataType [1986].)</td>
</tr>
<tr>
<td>Length [2158]</td>
<td>Gets the number of characters within the string (when fixed length).</td>
</tr>
<tr>
<td>Name [1992]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from</td>
</tr>
<tr>
<td></td>
<td>[DataType [1986].)</td>
</tr>
<tr>
<td>Namespace [1992]</td>
<td>Gets the namespace string within the [DataType [1986] exists. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from [DataType [1986].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984]</td>
</tr>
<tr>
<td></td>
<td>(Inherited from [BitSize [1982].)</td>
</tr>
</tbody>
</table>

**Reference**

IStringType Interface [2155]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.80.1.1 IStringType.Encoding Property

Gets the encoding of the String (Encoding.Default (STRING) or Encoding.UNICODE (WSTRING))

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
Encoding Encoding { get; }
```

**Property Value**

Type: Encoding

The encoding.

**Reference**

IStringType Interface [2155]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.80.1.2 IStringType.IsFixedLength Property

Gets a value indicating whether the string is of fixed length.
**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
bool IsFixedLength { get; }
```

### Property Value

**Type:** Boolean

true if this instance is fixed length; otherwise, false.

### Reference

IStringType Interface

TwinCAT.TypeSystem Namespace

#### 6.11.80.1.3 IStringType.Length Property

Gets the number of characters within the string (when fixed length).

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
int Length { get; }
```

### Property Value

**Type:** Int32

The length if fixed length, otherwise -1

### Reference

IStringType Interface

TwinCAT.TypeSystem Namespace

#### 6.11.81 IStructInstance Interface

Interface representing an instance of a IStructType

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public interface IStructInstance : ISymbol, IAttributedInstance, IInstance, IBitSize
```
The IStructInstance type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the DataType in bits. (Inherited from IBitSize)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from IBitSize)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the Instance (Inherited from Instance)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the DataType of the Instance. (Inherited from IInstance)</td>
</tr>
<tr>
<td>HasRpcMethods</td>
<td>Gets a value indicating whether this instance has RPC methods</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods .) (Inherited from Instance)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from Instance)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this ISymbol is persistent. (Inherited from ISymbol)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the Instance represents a Pointer type (Pointer TO) (Inherited from Instance)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the Instance represents a Reference type (REFERENCE TO) (Inherited from Instance)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this Instance is static. (Inherited from Instance)</td>
</tr>
<tr>
<td>MemberInstances</td>
<td>Gets the member instances of the Struct Instance.</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol (Inherited from ISymbol)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType (Inherited from IBitSize)</td>
</tr>
</tbody>
</table>
## IStructInstance Properties

The `IStructInstance` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from <code>IAtributedInstance</code>)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>IDataType</code> in bits. (Inherited from <code>IBitSize</code>)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in bytes. (Inherited from <code>IBitSize</code>)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Symbol/Datatype Category (Inherited from <code>ISymbol</code>)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the <code>IInstance</code> (Inherited from <code>IInstance</code>)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the <code>IDataType</code> of the <code>IInstance</code>. (Inherited from <code>IInstance</code>)</td>
</tr>
<tr>
<td>HasRpcMethods</td>
<td>Gets a value indicating whether this instance has RPC methods</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods .) (Inherited from <code>IInstance</code>)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from <code>IInstance</code>)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full <code>DataType</code> but instead of some sort of bit mapping (Inherited from <code>IBitSize</code>)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <code>IBitSize</code>)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from <code>ISymbol</code>)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this <code>ISymbol</code> is persistent. (Inherited from <code>ISymbol</code>)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the <code>IInstance</code> represents a Pointer type (Pointer TO) (Inherited from <code>IInstance</code>)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from <code>ISymbol</code>)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE TO) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this Instance [2052] is static. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>MemberInstances [2162]</td>
<td>Gets the member instances of the Struct Instance [2158].</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052]. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from AttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

Reference

IStructInstance Interface [2158]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.81.1.1 IStructInstance.HasRpcMethods Property

Ggets a value indicating whether this instance has RPC methods

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
bool HasRpcMethods { get; }
```

**Property Value**

Type: Boolean
ture if this instance has RPC methods; otherwise, false.

**Remarks**

If the struct instance supports RPC Methods, then the instance class is also supporting IRpcStructInstance [2140]:

Reference

IStructInstance Interface [2158]
6.11.81.1.2 IStructInstance.MemberInstances Property

Gets the member instances of the Struct Instance.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  

```
ISymbolCollection<ISymbol> MemberInstances { get; }
```

Property Value

Type: ISymbolCollection | ISymbol. The member instances.

Reference

IStructInstance Interface

TwinCAT.TypeSystem Namespace

6.11.82 IStructType Interface

Interface representing Struct data types

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  

```
public interface IStructType : IDataType, IBitSize
```

The IStructType type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllMembers</td>
<td>Gets all members (down the derivation hierarchy)</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType (Inherited from IDataType)</td>
</tr>
<tr>
<td>BaseType</td>
<td>Gets the structs Base Type (Null if not derived).</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### IStructType Properties

The `IStructType` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllMembers</td>
<td>Gets all members (down the derivation hierarchy)</td>
</tr>
</tbody>
</table>

---

Reference

TwinCAT.TypeSystem Namespace [1622]

### 6.11.82.1 IStructType Properties

The `IStructType` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1987]</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>BaseType [2165]</td>
<td>Gets the structs Base Type (Null if not derived).</td>
</tr>
<tr>
<td>BaseTypeName [2165]</td>
<td>Gets the the Name of the Base class (if derived)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [1988]</td>
<td>Gets the Data Type category (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Comment [1988]</td>
<td>Gets the comment behind the variable declaration. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>FullName [1989]</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>HasRpcMethods [2166]</td>
<td>Gets a value indicating whether this instance has RPC methods (Struct types only)</td>
</tr>
<tr>
<td>Id [1989]</td>
<td>Gets the ID of the DataType (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainer [1990]</td>
<td>Gets a value indicating whether this IDatatype [1986] is a container type (Inherited from IDatatype [1986].)</td>
</tr>
<tr>
<td>IsPointer [1990]</td>
<td>Gets a value indicating whether this IDatatype [1986] is a pointer type (Inherited from IDatatype [1986].)</td>
</tr>
<tr>
<td>IsPrimitive [1991]</td>
<td>Gets a value indicating whether this IDatatype [1986] is primitive (Inherited from IDatatype [1986].)</td>
</tr>
<tr>
<td>IsReference [1991]</td>
<td>Gets a value indicating whether this IDatatype [1986] is a reference type (Inherited from IDatatype [1986].)</td>
</tr>
<tr>
<td>Members [2166]</td>
<td>Gets a readonly collection of the Members [2065] of the IStructType [2162].</td>
</tr>
<tr>
<td>Name [1992]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from IDatatype [1986].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
</tbody>
</table>

**Reference**

IStructType Interface [2162]

TwinCAT.TypeSystem Namespace [1622]
6.11.82.1.1 **IStructType.AllMembers Property**

Gets all members (down the derivation hierarchy)

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
IMemberCollection AllMembers { get; }
```

**Property Value**

Type: `IMemberCollection`  
All members.

**Reference**

`IStructType Interface`  
`TwinCAT.TypeSystem Namespace`

---

6.11.82.1.2 **IStructType.BaseType Property**

Gets the structs Base Type (Null if not derived).

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
IDataType BaseType { get; }
```

**Property Value**

Type: `IDataType`

**Reference**

`IStructType Interface`  
`TwinCAT.TypeSystem Namespace`

---

6.11.82.1.3 **IStructType.BaseTypeName Property**

Gets the the Name of the Base class (if derived)

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
string BaseTypeName { get; }
```

Property Value

Type: String
Empty if not derived.

Reference

[StructType Interface [2162]]

[TwinCAT.TypeSystem Namespace [1622]]

6.11.82.1.4 IStructType.HasRpcMethods Property

Gets a value indicating whether this instance has RPC methods (Struct types only)

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
bool HasRpcMethods { get; }
```

Property Value

Type: Boolean
true if this instance has RPC methods; otherwise, false.

Remarks

The DataType (Structure) must be marked with the PlcAttribute 'TcRpcEnable' to enable RpcMethods, otherwise RpcMethods are not passed through to the ADS symbolic information.

Reference

[StructType Interface [2162]]

[TwinCAT.TypeSystem Namespace [1622]]

6.11.82.1.5 IStructType.Members Property

Gets a readonly collection of the Members [2065] of the IStructType [2162].

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
IMemberCollection Members { get; }
```
Property Value

Type: IMemberCollection [2068]
The members as readonly collection.

Remarks

If the IStructType [2162] is derived, only the extended members are returned. To get all supported members down the inheritance chain, use the AllMembers [2165] property.

Reference

IStructType Interface [2162]
TwinCAT.TypeSystem Namespace [1622]

6.11.83 IStructValue Interface

Interface IStructValue

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C# public interface IStructValue : IValue

The IStructValue type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age [2228]</td>
<td>Gets the age of the value (last successful read of the value) (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>CachedRaw [2228]</td>
<td>Gets the cached Raw internal Data. (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>DataType [2228]</td>
<td>Gets the data type bound to this IValue [2226] (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>IsPrimitive [2229]</td>
<td>Gets a value indicating whether this IValue [2226] is a primitive value. (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>Symbol [2229]</td>
<td>Gets the symbol bound to this IValue [2226]. (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>TimeStamp [2230]</td>
<td>Gets the Time stamp of the last successful read of the Value (local user time, UTC) (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>UpdateMode [2230]</td>
<td>Gets the update mode (not implemented yet) (Inherited from IValue [2226].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read [2231]</td>
<td>Reads the value (via ADS) (Inherited from IValue [2226].)</td>
</tr>
<tr>
<td>ReadAsync [2231]</td>
<td>Reads the value (via ADS) (Inherited from IValue [2226].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResolveValue</td>
<td>Resolves the Value object to its primitive value. (Inherited from IValue)</td>
</tr>
<tr>
<td>TryGetMemberValue</td>
<td>Tries to get a property/Member value.</td>
</tr>
<tr>
<td>TryResolveValue</td>
<td>Tries to resolves the Value object to its primitive value. (Inherited from IValue)</td>
</tr>
<tr>
<td>TrySetMemberValue</td>
<td>Tries to Set a Member/Property Value</td>
</tr>
<tr>
<td>Write</td>
<td>Writes the value (via ADS) (Inherited from IValue)</td>
</tr>
<tr>
<td>WriteAsync</td>
<td>Writes the value (via ADS) (Inherited from IValue)</td>
</tr>
</tbody>
</table>

**Reference**

[TwinCAT.TypeSystem Namespace](#) [1622]

[TwinCAT.TypeSystem.IValue](#) [2226]

### 6.11.83.1 IStructValue Properties

The IStructValue type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Gets the age of the value (last successful read of the value) (Inherited from IValue)</td>
</tr>
<tr>
<td>CachedRaw</td>
<td>Gets the cached Raw internal Data. (Inherited from IValue)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the data type bound to this IValue (Inherited from IValue)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IValue is a primitive value. (Inherited from IValue)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol bound to this IValue. (Inherited from IValue)</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Gets the Time stamp of the last successful read of the Value (local user time, UTC) (Inherited from IValue)</td>
</tr>
<tr>
<td>UpdateMode</td>
<td>Gets the update mode (not implemented yet) (Inherited from IValue)</td>
</tr>
</tbody>
</table>

**Reference**

IStructValue Interface [2167]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.83.2 IStructValue Methods

The IStructValue type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Reads the value (via ADS) (Inherited from IValue.2226.)</td>
</tr>
<tr>
<td>ReadAsync</td>
<td>Reads the value (via ADS) (Inherited from IValue.2226.)</td>
</tr>
<tr>
<td>ResolveValue</td>
<td>Resolves the Value object to its primitive value. (Inherited from IValue.2226.)</td>
</tr>
<tr>
<td>TryGetMemberValue</td>
<td>Tries to get a property/Member value.</td>
</tr>
<tr>
<td>TryResolveValue</td>
<td>Tries to resolves the Value object to its primitive value. (Inherited from IValue.2226.)</td>
</tr>
<tr>
<td>TrySetMemberValue</td>
<td>Tries to Set a Member/Property Value</td>
</tr>
<tr>
<td>Write</td>
<td>Writes the value (via ADS) (Inherited from IValue.2226.)</td>
</tr>
<tr>
<td>WriteAsync</td>
<td>Writes the value (via ADS) (Inherited from IValue.2226.)</td>
</tr>
</tbody>
</table>

## Reference

<table>
<thead>
<tr>
<th>Reference</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IStructValue Interface.2167</td>
<td></td>
</tr>
<tr>
<td>TwinCAT.TypeSystem Namespace.1622</td>
<td></td>
</tr>
</tbody>
</table>

### 6.11.83.2.1 IStructValue.TryGetMemberValue Method

Tries to get a property/Member value.

**Namespace:** TwinCAT.TypeSystem.1622  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)  
**Version:** 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
bool TryGetMemberValue(
    string name,
    out Object value
)
```

**Parameters**

| name | Type: System.String  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The name of the member</td>
</tr>
</tbody>
</table>
| value| Type: System.Object  
|      | The value. |

**Return Value**

Type: Boolean  
true if succeeded, otherwise false otherwise.

**Reference**

<table>
<thead>
<tr>
<th>Reference</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IStructValue Interface.2167</td>
<td></td>
</tr>
</tbody>
</table>
6.11.83.2.2 IStructValue.TrySetMemberValue Method

Tries to Set a Member/Property Value

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
bool TrySetMemberValue(
    string name,
    Object value
)
```

Parameters

name  
Type: System.String  
The name of the member

value  
Type: System.Object  
The value.

Return Value

Type: Boolean  
true if succeeded, otherwise false otherwise.

Reference

IStructValue Interface [2167]

TwinCAT.TypeSystem Namespace [1622]

6.11.84 ISubRangeType Interface

Interface representing a SubRange type

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface ISubRangeType : IDataType,
    IBitSize
```

The ISubRangeType type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType (Inherited from IDataType.)</td>
</tr>
</tbody>
</table>
### ISubRangeType Properties

The `ISubRangeType` type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the <code>IDataType</code> (Inherited from <code>IDataType</code>).</td>
</tr>
<tr>
<td>BaseType</td>
<td>Gets the the base type of the <code>ISubRangeType</code>.</td>
</tr>
<tr>
<td>BaseTypeName</td>
<td>Gets the name of the base type.</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>IDataType</code> in bits. (Inherited from <code>IBitSize</code>).</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from</td>
</tr>
<tr>
<td></td>
<td><code>IBitSize</code>).</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from <code>IDataType</code>).</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment behind the variable declaration. (Inherited from <code>IDataType</code>).</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the <code>IDataType</code> (Namespace + Name) (Inherited from</td>
</tr>
<tr>
<td></td>
<td><code>IDataType</code>).</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the Data Type (Inherited from <code>IDataType</code>).</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full</td>
</tr>
<tr>
<td></td>
<td><code>IDataType</code> but instead of some sort of bit mapping (Inherited from <code>IBitSize</code>).</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>IBitSize</code>).</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a container type</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>IDataType</code>).</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a pointer type</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>IDataType</code>).</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this <code>IDataType</code> is primitive</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>IDataType</code>).</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this <code>IDataType</code> is a reference type</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>IDataType</code>).</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from <code>IDataType</code>).</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the <code>IDataType</code> exists. (Inherited from</td>
</tr>
<tr>
<td></td>
<td><code>IDataType</code>).</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on <code>IsBitType</code> (Inherited from <code>IBitSize</code>).</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BaseTypeNamn</td>
<td>Gets the name of the base type.</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from IBitSize.</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from IDataType.</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment behind the variable declaration. (Inherited from</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the IDataType (Namespace + Name) (Inherited from</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the Data Type (Inherited from IDataType.</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this IDataType is a container type</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this IDataType is a pointer type</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IDataType is primitive</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this IDataType is a reference type</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the IDataType exists. (Inherited from</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType.</td>
</tr>
</tbody>
</table>

Reference

ISubRangeType Interface  

TwinCAT.TypeSystem Namespace  

6.11.84.1.1 ISubRangeType.BaseType Property

Gets the base type of the ISubRangeType.

Namespace: TwinCAT.TypeSystem  
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

IDataType BaseType { get; }

Property Value

Type: IDataType
The base datatype or null if not resolved.

Reference

ISubRangeType Interface
TwinCAT.TypeSystem Namespace

6.11.84.1.2 ISubRangeType.BaseTypeName Property

Gets the name of the base type.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

string BaseTypeName { get; }

Property Value

Type: String
The name of the base type.

Reference

ISubRangeType Interface
TwinCAT.TypeSystem Namespace

6.11.85 ISubRangeType.T Interface

Interface representing a SubRange type

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public interface ISubRangeType<T> : ISubRangeType,
IDataType, IBitSize
where T : struct, new()
Type Parameters

T

The ISubRangeType.T. type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1987]</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>BaseType [2172]</td>
<td>Gets the base type of the ISubRangeType [2170] (Inherited from ISubRangeType [2170].)</td>
</tr>
<tr>
<td>BaseTypeName [2173]</td>
<td>Gets the name of the base type. (Inherited from ISubRangeType [2170].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [1988]</td>
<td>Gets the Data Type category (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Comment [1988]</td>
<td>Gets the comment behind the variable declaration. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>FullName [1989]</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Id [1989]</td>
<td>Gets the ID of the DataType (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a container type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPointer [1990]</td>
<td>Gets a value indicating whether this IDataType [1986] is a pointer type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPrimitive [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is primitive (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsReference [1991]</td>
<td>Gets a value indicating whether this IDataType [1986] is a reference type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>LowerBound [2176]</td>
<td>Gets the lower bound.</td>
</tr>
<tr>
<td>Name [1992]</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Namespace [1992]</td>
<td>Gets the namespace string within the IDataType [1986] exists. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>UpperBound [2176]</td>
<td>Gets the upper bound.</td>
</tr>
</tbody>
</table>
6.11.85.1  ISubRangeType.T. Properties

The ISubRangeType.T. [2173] generic type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType [1986] (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>BaseType</td>
<td>Gets the the base type of the ISubRangeType [2170] (Inherited from ISubRangeType [2170].)</td>
</tr>
<tr>
<td>BaseTypeName</td>
<td>Gets the name of the base type. (Inherited from ISubRangeType [2170].)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment behind the variable declaration. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the IDataType [1986] (Namespace + Name) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the DataType (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this IDataType [1986] is a container type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this IDataType [1986] is a pointer type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IDataType [1986] is primitive (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this IDataType [1986] is a reference type (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>LowerBound</td>
<td>Gets the lower bound.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the IDataType [1986] exists. (Inherited from IDataType [1986].)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>UpperBound</td>
<td>Gets the upper bound.</td>
</tr>
</tbody>
</table>
6.11.85.1.1 ISubRangeType.T..LowerBound Property

Gets the lower bound.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
T LowerBound { get; }
```

**Property Value**

Type: T

The lower bound.

6.11.85.1.2 ISubRangeType.T..UpperBound Property

Gets the upper bound.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
T UpperBound { get; }
```

**Property Value**

Type: T

The upper bound.

6.11.86 ISymbol Interface

Interface specifying Symbols ( 
TwinCAT.Ads Namespaces

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface ISymbol : IAttributedInstance,
    IInstance, IBitSize
```

The ISymbol type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Symbol/Datatype Category</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the IInstance (Inherited from IInstance.)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType of the IInstance. (Inherited from IInstance.)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods .) (Inherited from IInstance.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods .)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full Data-</td>
</tr>
<tr>
<td></td>
<td>Type but instead of some sort of bit mapping. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type.</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this ISymbol is persistent.</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the IInstance represents a Pointer type (Pointer TO) (Inherited from IInstance.)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type.</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only.</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive.</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the IInstance represents a Reference type (Reference TO) (Inherited from IInstance.)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this IInstance is static. (Inherited from IInstance.)</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984]</td>
</tr>
<tr>
<td></td>
<td>(Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052].</td>
</tr>
<tr>
<td></td>
<td>(Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

### 6.11.86.1 ISymbol Properties

The ISymbol [2176] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods (.) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full</td>
</tr>
<tr>
<td></td>
<td>DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is a container type.</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent.</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type.</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only.</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive.</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the Instance represents a Reference type (REFERENCE TO)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this Instance is static.</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType</td>
</tr>
<tr>
<td>SubSymbols</td>
<td>Gets the SubSymbols of the ISymbol</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the name of the DataType that is used for this Instance</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding.</td>
</tr>
</tbody>
</table>

### Reference

**ISymbol Interface**

**TwinCAT.TypeSystem Namespace**

### 6.11.86.1.1 ISymbol.Category Property

Gets the Symbol/Datatype Category

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
DataTypeCategory Category { get; }
```

**Property Value**

Type: `DataTypeCategory`

The category.

### Reference

**ISymbol Interface**

**TwinCAT.TypeSystem Namespace**

### 6.11.86.1.2 ISymbol.IsContainerType Property

Gets a value indicating whether this Symbol is a container type.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
bool IsContainerType { get; }
```

**Property Value**

Type: `Boolean`

true if this instance is container type; otherwise, false.

**Reference**

`ISymbol Interface [2176]`

`TwinCAT.TypeSystem Namespace [1622]`

### 6.11.86.1.3 ISymbol.IsPersistent Property

Gets a value indicating whether this `ISymbol` is persistent.

**Namespace:** `TwinCAT.TypeSystem [1622]`

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
bool IsPersistent { get; }
```

**Property Value**

Type: `Boolean`

true if this instance is persistent; otherwise, false.

**Reference**

`ISymbol Interface [2176]`

`TwinCAT.TypeSystem Namespace [1622]`

### 6.11.86.1.4 ISymbol.IsPrimitiveType Property

Gets a value indicating whether this instance is a primitive type.

**Namespace:** `TwinCAT.TypeSystem [1622]`

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
bool IsPrimitiveType { get; }
```

**Property Value**

Type: `Boolean`

true if this instance is primitive type; otherwise, false.
6.11.86.1.5  ISymbol.IsReadOnly Property

Indicates that this instance is read only.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
bool IsReadOnly { get; }
```

**Property Value**

Type: Boolean

**Remarks**

Actually, this Flag is restricted to TcCOM-Objects readonly Parameters. Within the PLC this is used for the ApplicationName and ProjectName of PLC instances. Write-Access on these Modules will create an DeviceAccessDenied error.

**Reference**

ISymbol Interface

TwinCAT.TypeSystem Namespace

6.11.86.1.6  ISymbol.IsRecursive Property

Gets a value indicating whether this instance is recursive.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
bool IsRecursive { get; }
```

**Property Value**

Type: Boolean

true if this instance is recursive; otherwise, false.
6.11.86.1.7  ISymbol.Parent Property

Gets the parent Symbol

Namespace:  TwinCAT.TypeSystem
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

ISymbol Parent { get; }

Property Value

Type:  ISymbol
The parent.

Reference

ISymbol Interface
TwinCAT.TypeSystem Namespace

6.11.86.1.8  ISymbol.SubSymbols Property

Gets the SubSymbols of the ISymbol

Namespace:  TwinCAT.TypeSystem
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

ISymbolCollection<ISymbol> SubSymbols { get; }

Property Value

Type:  ISymbolCollection, ISymbol

Remarks

Used for Array, Struct, Pointer and Reference instances. Otherwise empty

Reference

ISymbol Interface
TwinCAT.TypeSystem Namespace

6.11.87  ISymbolCollection Interface

Interface ISymbolCollection Implements the IInstanceCollection.T.
Syntax

C#

```csharp
public interface ISymbolCollection : ISymbolCollection<ISymbol>,
  IInstanceCollection<ISymbol>, IList<ISymbol>, ICollection<ISymbol>,
  IEnumerable<ISymbol>, IEnumerable
```

The ISymbolCollection type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T.</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T is read-only.</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the IInstance with the specified instance path.</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Mode</td>
<td>Gets the InstanceCollectionMode.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection.T.</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T.</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether this collection contains an instance with the specified</td>
</tr>
<tr>
<td></td>
<td>instance path.</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection.T contains a specific value.</td>
</tr>
<tr>
<td>ContainsName</td>
<td>Determines whether this collection contains an instance with the specified</td>
</tr>
<tr>
<td></td>
<td>instance name.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection.T to an Array, starting at a particular</td>
</tr>
<tr>
<td></td>
<td>Array index.</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetInstance</td>
<td>Gets the IInstance by instance path.</td>
</tr>
<tr>
<td>GetInstanceByName</td>
<td>Gets the IInstance by instance name.</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList.T.</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList.T at the specified index.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection.T.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from ICollection.ISymbol[2176]..)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IList.T. item at the specified index. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>IList.ISymbol[2176]..)</td>
</tr>
<tr>
<td>TryGetInstance[2063]</td>
<td>Tries to get the specified instance. (Inherited from InstanceCollection.T.</td>
</tr>
<tr>
<td></td>
<td>[2057].)</td>
</tr>
<tr>
<td>TryGetInstanceByName[2064]</td>
<td>Tries to get the specified instance by name. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>InstanceCollection.T.[2057].)</td>
</tr>
</tbody>
</table>

Reference
TwinCAT.TypeSystem Namespace [1622]
TwinCAT.TypeSystem.InstanceCollection.T.[2057]

6.11.87.1 ISymbolCollection Properties
The ISymbolCollection[2182] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>ICollection.ISymbol[2176]..)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T. is read-only. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from ICollection.ISymbol[2176]..)</td>
</tr>
<tr>
<td>Item.String.[2059]</td>
<td>Gets the Instance[2052] with the specified instance path. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>InstanceCollection.T.[2057].)</td>
</tr>
<tr>
<td>Item.Int32.</td>
<td>Gets or sets the element at the specified index. (Inherited from IList.ISymbol[2176]..)</td>
</tr>
<tr>
<td>Mode.[2060]</td>
<td>Gets the InstanceCollectionMode[2075]. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>InstanceCollection.T.[2057].)</td>
</tr>
</tbody>
</table>

Reference
ISymbolCollection Interface [2182]
TwinCAT.TypeSystem Namespace[1622]

6.11.87.2 ISymbolCollection Methods
The ISymbolCollection[2182] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection.T. (Inherited from ICollection.ISymbol[2176]..)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>ICollection.ISymbol[2176]..)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether this collection contains an instance with the specified</td>
</tr>
<tr>
<td>[2061]</td>
<td>instance path. (Inherited from IInstanceCollection&lt;T&gt; [2057].)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection&lt;T&gt; contains a specific value. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from ICollection&lt;ISymbol&gt; [2176].)</td>
</tr>
<tr>
<td>ContainsName</td>
<td>Determines whether this collection contains an instance with the specified</td>
</tr>
<tr>
<td>[2062]</td>
<td>instance name. (Inherited from IInstanceCollection&lt;T&gt; [2057].)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection&lt;T&gt; to an Array, starting at a</td>
</tr>
<tr>
<td></td>
<td>particular Array index. (Inherited from ICollection&lt;ISymbol&gt; [2176].)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>IList&lt;ISymbol&gt; [2176].)</td>
</tr>
<tr>
<td>GetInstance</td>
<td>Gets the IInstance by instance path. (Inherited from</td>
</tr>
<tr>
<td>[2062]</td>
<td>IInstanceCollection&lt;T&gt; [2057].)</td>
</tr>
<tr>
<td>GetInstanceByName</td>
<td>Gets the IInstance by instance name. (Inherited from IInstanceCollection&lt;T&gt;</td>
</tr>
<tr>
<td>[2063]</td>
<td>[2057].)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList&lt;T&gt;. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>IList&lt;ISymbol&gt; [2176].)</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList&lt;T&gt; at the specified index. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>IList&lt;ISymbol&gt; [2176].)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from ICollection&lt;ISymbol&gt; [2176].)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IList&lt;T&gt; item at the specified index. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>IList&lt;ISymbol&gt; [2176].)</td>
</tr>
<tr>
<td>TryGetInstance</td>
<td>Tries to get the specified instance. (Inherited from IInstanceCollection&lt;T&gt;</td>
</tr>
<tr>
<td>[2063]</td>
<td>[2057].)</td>
</tr>
<tr>
<td>TryGetInstanceByName</td>
<td>Tries to get the specified instance by name. (Inherited from</td>
</tr>
<tr>
<td>[2064]</td>
<td>IInstanceCollection&lt;T&gt; [2057].)</td>
</tr>
</tbody>
</table>

Reference

ISymbolCollection Interface [2182]

TwinCAT.TypeSystem Namespace [1622]

6.11.88 ISymbolCollection<T>. Interface

Interface ISymbolCollection

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0a15da1c14

Syntax

C#

```csharp
public interface ISymbolCollection<T> : IInstanceCollection<T>,
  IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable<T>
where T : class, ISymbol
```

Type Parameters

T
The `ISymbolCollection<T>` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the <code>ICollection&lt;T&gt;</code> is read-only. (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td><code>Item[String]</code></td>
<td>Gets the <code>IInstance</code> with the specified instance path. (Inherited from <code>IInstanceCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>Item.Int32.</td>
<td>Gets or sets the element at the specified index. (Inherited from <code>IList&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>Mode</td>
<td>Gets the <code>InstanceCollectionMode</code>. (Inherited from <code>IInstanceCollection&lt;T&gt;</code>)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the <code>ICollection&lt;T&gt;</code> (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the <code>ICollection&lt;T&gt;</code> (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether this collection contains an instance with the specified instance path. (Inherited from <code>IInstanceCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the <code>ICollection&lt;T&gt;</code> contains a specific value. (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>ContainsName</td>
<td>Determines whether this collection contains an instance with the specified instance name. (Inherited from <code>IInstanceCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the <code>ICollection&lt;T&gt;</code> to an <code>Array</code>, starting at a particular <code>Array</code> index. (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from <code>IEnumerable&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>GetInstance</td>
<td>Gets the <code>IInstance</code> by instance path. (Inherited from <code>IInstanceCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>GetInstanceByName</td>
<td>Gets the <code>IInstance</code> by instance name. (Inherited from <code>IInstanceCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the <code>IList&lt;T&gt;</code>. (Inherited from <code>IList&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the <code>IList&lt;T&gt;</code> at the specified index. (Inherited from <code>IList&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the <code>ICollection&lt;T&gt;</code> (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the <code>IList&lt;T&gt;</code> item at the specified index. (Inherited from <code>IList&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>TryGetInstance</td>
<td>Tries to get the specified instance. (Inherited from <code>IInstanceCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>TryGetInstanceByName</td>
<td>Tries to get the specified instance by name. (Inherited from <code>IInstanceCollection&lt;T&gt;</code>)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]
6.11.88.1  ISymbolCollection.T. Properties

The ISymbolCollection.T. generic type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.T..)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.T..)</td>
</tr>
<tr>
<td>Item.String [2059]</td>
<td>Gets the Instance [2052] with the specified instance path. (Inherited from InstanceCollection.T..)</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the element at the specified index. (Inherited from IList.T..)</td>
</tr>
<tr>
<td>Mode [2060]</td>
<td>Gets the InstanceCollectionMode [2075]. (Inherited from InstanceCollection.T..)</td>
</tr>
</tbody>
</table>

### Reference

ISymbolCollection.T. Interface [2185]

TwinCAT.TypeSystem Namespace [1622]

6.11.88.2  ISymbolCollection.T. Methods

The ISymbolCollection.T. generic type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection.T.. (Inherited from ICollection.T..)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T.. (Inherited from ICollection.T..)</td>
</tr>
<tr>
<td>Contains(String) [2061]</td>
<td>Determines whether this collection contains an instance with the specified instance path. (Inherited from InstanceCollection.T..)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.T..)</td>
</tr>
<tr>
<td>ContainsName [2062]</td>
<td>Determines whether this collection contains an instance with the specified instance name. (Inherited from InstanceCollection.T..)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.T..)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.T..)</td>
</tr>
<tr>
<td>GetInstance [2062]</td>
<td>Gets the Instance [2052] by instance path. (Inherited from InstanceCollection.T..)</td>
</tr>
<tr>
<td>GetInstanceByName [2063]</td>
<td>Gets the Instance [2052] by instance name. (Inherited from InstanceCollection.T..)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IndexOf</strong></td>
<td>Determines the index of a specific item in the <code>IList&lt;T&gt;</code> (Inherited from <code>IList&lt;T&gt;</code>)</td>
</tr>
<tr>
<td><strong>Insert</strong></td>
<td>Inserts an item to the <code>IList&lt;T&gt;</code> at the specified index. (Inherited from <code>IList&lt;T&gt;</code>)</td>
</tr>
<tr>
<td><strong>Remove</strong></td>
<td>Removes the first occurrence of a specific object from the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td><strong>RemoveAt</strong></td>
<td>Removes the <code>IList&lt;T&gt;</code> item at the specified index. (Inherited from <code>IList&lt;T&gt;</code>)</td>
</tr>
<tr>
<td><strong>TryGetInstance</strong></td>
<td>Tries to get the specified instance. (Inherited from <code>IInstanceCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td><strong>TryGetInstanceByName</strong></td>
<td>Tries to get the specified instance by name. (Inherited from <code>IInstanceCollection&lt;T&gt;</code>)</td>
</tr>
</tbody>
</table>

### Reference

- `ISymbolCollection<T> Interface` ![2185]
- `TwinCAT.TypeSystem Namespace` ![1622]

### 6.11.89 ISymbolFactory Interface

Symbol Factory Interface

**Namespace:** TwinCAT.TypeSystem ![1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public interface ISymbolFactory
```

The ISymbolFactory type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FactoryServices</strong></td>
<td>Gets the factory services.</td>
</tr>
<tr>
<td><strong>HasInvalidCharacters</strong></td>
<td>Gets a value indicating whether <code>ISymbol</code> has invalid characters</td>
</tr>
<tr>
<td><strong>InvalidCharacters</strong></td>
<td>Gets the invalid characters that are not allowed to appear within the Instance Name</td>
</tr>
</tbody>
</table>

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CreateArrayElement</strong></td>
<td>Creates a single Array Element</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateArrayElementInstances</td>
<td>Creates all Element Instances of the specified array parent symbol.</td>
</tr>
<tr>
<td>CreateFieldInstance</td>
<td>Creates a single Instance member on a struct parent.</td>
</tr>
<tr>
<td>CreateFieldInstanceMemberCollection</td>
<td>Creates the Member Instances collection for the specified parent instance.</td>
</tr>
<tr>
<td>CreateInstance</td>
<td>Creates the Symbol with the specified resolver.</td>
</tr>
<tr>
<td>CreateInstanceAsync</td>
<td>Creates the Symbol with the specified resolver.</td>
</tr>
<tr>
<td>CreateReferenceInstance</td>
<td>Creates the dereferenced Pointer instance.</td>
</tr>
<tr>
<td>CreateVirtualStruct</td>
<td>Creates the virtual structure.</td>
</tr>
<tr>
<td>Initialize</td>
<td>Initializes the ISymbolFactory.</td>
</tr>
<tr>
<td>SetInvalidCharacters</td>
<td>Sets the invalid characters.</td>
</tr>
</tbody>
</table>

#### Reference

- TwinCAT.TypeSystem Namespace [1622]

### 6.11.89.1 ISymbolFactory Properties

The ISymbolFactory [2188] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FactoryServices</td>
<td>Gets the factory services.</td>
</tr>
<tr>
<td>HasInvalidCharacters</td>
<td>Gets a value indicating whether ISymbols have invalid characters.</td>
</tr>
<tr>
<td>InvalidCharacters</td>
<td>Gets the invalid characters that are not allowed to appear within the Instance Name</td>
</tr>
</tbody>
</table>

#### Reference

- ISymbolFactory Interface [2188]
  - TwinCAT.TypeSystem Namespace [1622]

### 6.11.89.1.1 ISymbolFactory.FactoryServices Property

Gets the factory services.
**TwinCAT.Ads Namespaces**

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
ISymbolFactoryServices FactoryServices { get; }
```

### Property Value

Type: ISymbolFactoryServices
The factory services.

### Reference

- ISymbolFactory Interface
- TwinCAT.TypeSystem Namespace

### 6.11.89.1.2 ISymbolFactory.HasInvalidCharacters Property

Gets a value indicating whether |Symbol|s have invalid characters

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
bool HasInvalidCharacters { get; }
```

### Property Value

Type: Boolean
true if this instance has invalid characters; otherwise, false.

### Reference

- ISymbolFactory Interface
- TwinCAT.TypeSystem Namespace
- ISymbolFactory.InvalidCharacters

### 6.11.89.1.3 ISymbolFactory.InvalidCharacters Property

Gets the invalid characters that are not allowed to appear within the Instance Name

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#
char[] InvalidCharacters { get; }

Property Value

Type: Char
The forbidden characters.

Reference

ISymbolFactory Interface [2188]
TwinCAT.TypeSystem Namespace [1622]
ISymbolFactory.SetInvalidCharacters(Char) [2197]
ISymbolFactory.HasInvalidCharacters [2190]

6.11.89.2 ISymbolFactory Methods

The ISymbolFactory [2188] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateArrayElement</td>
<td>Creates a single Array Element</td>
</tr>
<tr>
<td>CreateArrayElementInstances</td>
<td>Creates all Element Instances of the specified array parent symbol.</td>
</tr>
<tr>
<td>CreateFieldInstance</td>
<td>Creates a single Instance member on a struct parent</td>
</tr>
<tr>
<td>CreateFieldInstances</td>
<td>Creates the Member Instances collection for the specified parent instance</td>
</tr>
<tr>
<td>CreateInstance</td>
<td>Creates the Symbol with the specified resolver</td>
</tr>
<tr>
<td>CreateInstanceAsync</td>
<td>Creates the Symbol with the specified resolver</td>
</tr>
<tr>
<td>CreateReferenceInstance</td>
<td>Creates the dereferenced Pointer instance</td>
</tr>
<tr>
<td>CreateVirtualStruct</td>
<td>Creates the virtual structure.</td>
</tr>
<tr>
<td>Initialize</td>
<td>Initializes the the ISymbolFactory [2188].</td>
</tr>
<tr>
<td>SetInvalidCharacters</td>
<td>Sets the invalid characters.</td>
</tr>
</tbody>
</table>

Reference

ISymbolFactory Interface [2188]
TwinCAT.TypeSystem Namespace [1622]

6.11.89.2.1 ISymbolFactory.CreateArrayElement Method

Creates a single Array Element

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
ISymbol CreateArrayElement(  
    IArrayType arrayType,  
    int[] currentIndex,  
    ISymbol parent  
)

**Parameters**

- `arrayType`: Type: TwinCAT.TypeSystem.IArrayType [1971]  
  Resolved Array type.
- `currentIndex`: Type: System.Int32  
  Array Index of the Element
- `parent`: Type: TwinCAT.TypeSystem.ISymbol [2176]  
  Array Instance

**Return Value**

Type: ISymbol [2176]  
Array element

**Reference**

ISymbolFactory Interface [2188]  
TwinCAT.TypeSystem Namespace [1622]

6.11.89.2.2 ISymbolFactory.CreateArrayElementInstances Method

Creates all Element Instances of the specified array parent symbol.

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
ISymbolCollection<ISymbol> CreateArrayElementInstances(  
    ISymbol parentInstance,  
    IArrayType arrayType  
)
Parameters

parentInstance
Type: TwinCAT.TypeSystem.ISymbol
The parent instance.

arrayType
Type: TwinCAT.TypeSystem.IArrayType
Resolved array type.

Return Value

Type: ISymbolCollection|SymbolCollection.

Reference

ISymbolFactory Interface
TwinCAT.TypeSystem Namespace

6.11.89.2.3 ISymbolFactory.CreateFieldInstance Method

Creates a single Instance member on a struct parent

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

ISymbol CreateFieldInstance(
    IFIELD field,
    ISymbol parent
)

Parameters

field
Type: TwinCAT.TypeSystem.IField
Field

parent
Type: TwinCAT.TypeSystem.ISymbol
Parent Struct/Alias/Union

Return Value

Type: ISymbol
Instance member

Remarks

Because the Alias type can act like a struct, the parent can be an IAliasInstance also.

Reference

ISymbolFactory Interface
TwinCAT.TypeSystem Namespace
6.11.89.2.4 ISymbolFactory.CreateFieldInstances Method

Creates the Member Instances collection for the specified parent instance

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
ISymbolCollection<ISymbol> CreateFieldInstances(
    ISymbol parentInstance,
    IDataType parentType
)
```

**Parameters**

- **parentInstance**  
  Type: TwinCAT.TypeSystem.ISymbol [2176]  
  The parent instance.

- **parentType**  
  Type: TwinCAT.TypeSystem.IDataType [1986]  
  Parent Type (Struct/Alias/Union).

**Return Value**

Type: ISymbolCollection [2185] ISymbol [2176].  
SymbolCollection.

**Reference**

ISymbolFactory Interface [2188]  
TwinCAT.TypeSystem Namespace [1622]

6.11.89.2.5 ISymbolFactory.CreateInstance Method

Creates the Symbol with the specified resolver

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
ISymbol CreateInstance(
    ISymbolInfo entry,
    ISymbol parent
)
```

**Parameters**

- **entry**  
  Type: TwinCAT.TypeSystem.ISymbolInfo [2199]  
  Symbol Entry.

- **parent**  
  Type: TwinCAT.TypeSystem.ISymbol [2176]  
  The parent.
### Return Value

Type: ISymbol

Symbol instance.

### Reference

ISymbolFactory Interface

TwinCAT.TypeSystem Namespace

### 6.11.89.2.6 ISymbolFactory.CreateInstanceAsync Method

Creates the Symbol with the specified resolver

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
Task<ResultValue<ISymbol>> CreateInstanceAsync(
    ISymbolInfo entry,
    ISymbol parent,
    CancellationToken cancel
)
```

**Parameters**

- **entry**  
  Type: TwinCAT.TypeSystem.ISymbolInfo
  Symbol Entry.

- **parent**  
  Type: TwinCAT.TypeSystem.ISymbol
  The parent.

- **cancel**  
  Type: System.Threading.CancellationToken
  The cancellation token.

**Return Value**

Type: Task<ResultValue<ISymbol>>

Task<ResultValue<ISymbol>>.

**Reference**

ISymbolFactory Interface

TwinCAT.TypeSystem Namespace

### 6.11.89.2.7 ISymbolFactory.CreateReferenceInstance Method

Creates the dereferenced Pointer instance

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
### Syntax

**C#**

```csharp
ISymbol CreateReferenceInstance(
    IPointerType type,
    ISymbol parent
)
```

#### Parameters

- **type**  
  Type: `TwinCAT.TypeSystem.IPointerType`  
  Reference/Pointer type.

- **parent**  
  Type: `TwinCAT.TypeSystem.ISymbol`  
  Parent Instance of the reference

#### Return Value

Type: `ISymbol`  
Reference/Pointer instance.

### Reference

`ISymbolFactory Interface`  
`TwinCAT.TypeSystem Namespace`

### 6.11.89.2.8  `ISymbolFactory.CreateVirtualStruct Method`

Creates the virtual structure.

**Namespace:** `TwinCAT.TypeSystem`  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
ISymbol CreateVirtualStruct(
    string instanceName,
    string instancePath,
    ISymbol parent
)
```

#### Parameters

- **instanceName**  
  Type: `System.String`  
  Name of the instance.

- **instancePath**  
  Type: `System.String`  
  The instance path.

- **parent**  
  Type: `TwinCAT.TypeSystem.ISymbol`  
  The parent.

#### Return Value

Type: `ISymbol`  
Virtual struct instance
6.11.89.2.9  ISymbolFactory.Initialize Method

Initializes the ISymbolFactory.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
void Initialize(
    ISymbolFactoryServices services
)
```

**Parameters**

services  Type: ISymbolFactoryServices  
The services.

**Reference**

ISymbolFactory Interface

TwinCAT.TypeSystem Namespace

6.11.89.2.10  ISymbolFactory.SetInvalidCharacters Method

Sets the invalid characters.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
void SetInvalidCharacters(
    char[] invalidChars
)
```

**Parameters**

invalidChars  Type: System.Char  
The character.

**Reference**

ISymbolFactory Interface

TwinCAT.TypeSystem Namespace
6.11.90 ISymbolFactoryServiceProvider Interface

Symbol Value Access interface

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

class ISymbolFactoryServiceProvider

The ISymbolFactoryServiceProvider type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FactoryServices</td>
<td>Gets the factory services.</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

6.11.90.1 ISymbolFactoryServiceProvider Properties

The ISymbolFactoryServiceProvider [2198] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FactoryServices</td>
<td>Gets the factory services.</td>
</tr>
</tbody>
</table>

**Reference**

ISymbolFactoryServiceProvider Interface [2198]

TwinCAT.TypeSystem Namespace [1622]

6.11.90.1.1 ISymbolFactoryServiceProvider.FactoryServices Property

Gets the factory services.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

ISymbolFactoryServiceProvider.FactoryServices { get; }
Property Value
Type: ISymbolFactoryServices
The factory services.

Reference
ISymbolFactoryServicesProvider Interface [2198]
TwinCAT.TypeSystem Namespace [1622]

6.11.91 ISymbolInfo Interface

Interface ISymbolInfo

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public interface ISymbolInfo

The ISymbolInfo type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstancePath</td>
<td>Gets the Symbol Path</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the data type Name</td>
</tr>
</tbody>
</table>

Reference
TwinCAT.TypeSystem Namespace [1622]

6.11.91.1 ISymbolInfo Properties

The ISymbolInfo [2199] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstancePath</td>
<td>Gets the Symbol Path</td>
</tr>
<tr>
<td>TypeName</td>
<td>Gets the data type Name</td>
</tr>
</tbody>
</table>

Reference
ISymbolInfo Interface [2199]
TwinCAT.TypeSystem Namespace [1622]
### 6.11.91.1  ISymbolInfo.InstancePath Property

 Gets the Symbol Path

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
string InstancePath { get; }
```

**Property Value**

Type: `String`

The path.

**Reference**

ISymbolInfo Interface

TwinCAT.TypeSystem Namespace

### 6.11.91.2  ISymbolInfo.TypeName Property

 Gets the data type Name

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
string TypeName { get; }
```

**Property Value**

Type: `String`

The type of the data.

**Reference**

ISymbolInfo Interface

TwinCAT.TypeSystem Namespace

### 6.11.92  ISymbolLoader Interface

 Symbol Loader interface

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

public interface ISymbolLoader : ISymbolProvider, ISymbolServer

The ISymbolLoader type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BuildInTypes</td>
<td>Gets the build in types.</td>
</tr>
<tr>
<td>DataTypes       [2206]</td>
<td>Gets the data types (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>DefaultValueEncoding [2206]</td>
<td>Gets the default value encoding. (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>RootNamespaceName [2204]</td>
<td>Gets the name of the root namespace (Inherited from ISymbolProvider [2203].)</td>
</tr>
<tr>
<td>Settings [2202]</td>
<td>Gets or sets the access Method</td>
</tr>
<tr>
<td>Symbols [2207]</td>
<td>Gets the symbols. (Inherited from ISymbolServer [2205].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDataTypesAsync [2208]</td>
<td>Gets the data types asynchronously. (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>GetSymbolsAsync [2208]</td>
<td>Gets the symbols asynchronously (Inherited from ISymbolServer [2205].)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]

6.11.92.1 ISymbolLoader Properties

The ISymbolLoader [2200] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BuildInTypes</td>
<td>Gets the build in types.</td>
</tr>
<tr>
<td>DataTypes       [2206]</td>
<td>Gets the data types (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>DefaultValueEncoding [2206]</td>
<td>Gets the default value encoding. (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>RootNamespaceName [2204]</td>
<td>Gets the name of the root namespace (Inherited from ISymbolProvider [2203].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Settings [2202]</td>
<td>Gets or sets the access Method</td>
</tr>
<tr>
<td>Symbols [2207]</td>
<td>Gets the symbols. (Inherited from ISymbolServer [2205].)</td>
</tr>
</tbody>
</table>

### Reference

ISymbolLoader Interface [2200]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.92.1.1 ISymbolLoader.BuildInTypes Property

Gets the build in types.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
IDataTypeCollection BuildInTypes { get; }

**Property Value**

Type: IDataTypeCollection [1993]

The build in types.

**Reference**

ISymbolLoader Interface [2200]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.92.1.2 ISymbolLoader.Settings Property

Gets or sets the access Method

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
ISymbolLoaderSettings Settings { get; }

**Property Value**

Type: ISymbolLoaderSettings [99]

The access method.

**Reference**

ISymbolLoader Interface [2200]
TwinCAT.TypeSystem Namespace [1622]

6.11.92.2 ISymbolLoader Methods

The ISymbolLoader [2200] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDataTypesAsync</td>
<td>Gets the data types asynchronously. (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>GetSymbolsAsync</td>
<td>Gets the symbols asynchronously (Inherited from ISymbolServer [2205].)</td>
</tr>
</tbody>
</table>

Reference

ISymbolLoader Interface [2200]

TwinCAT.TypeSystem Namespace [1622]

6.11.93 ISymbolProvider Interface

Symbol Provider interface.

Namespace:  TwinCAT.TypeSystem [1622]  
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface ISymbolProvider : ISymbolServer
```

The ISymbolProvider type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypes [2206]</td>
<td>Gets the data types (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>DefaultValueEncoding</td>
<td>Gets the default value encoding. (Inherited from ISymbolServer [2205].)</td>
</tr>
<tr>
<td>RootNamespaceName</td>
<td>Gets the name of the root namespace</td>
</tr>
<tr>
<td>Symbols [2207]</td>
<td>Gets the symbols. (Inherited from ISymbolServer [2205].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDataTypesAsync</td>
<td>Gets the data types asynchronously. (Inherited from ISymbolServer [2205].)</td>
</tr>
</tbody>
</table>

### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSymbolsAsync ([2208])</td>
<td>Gets the symbols asynchronously (Inherited from ISymbolServer ([2205]).)</td>
</tr>
</tbody>
</table>

#### Reference

- [TwinCAT.TypeSystem Namespace](#) [1622]

#### 6.11.93.1 ISymbolProvider Properties

The ISymbolProvider ([2203]) type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypes ([2206])</td>
<td>Gets the data types (Inherited from ISymbolServer ([2205]).)</td>
</tr>
<tr>
<td>DefaultValueEncoding ([2206])</td>
<td>Gets the default value encoding. (Inherited from ISymbolServer ([2205]).)</td>
</tr>
<tr>
<td>RootNamespaceName ([2204])</td>
<td>Gets the name of the root namespace</td>
</tr>
<tr>
<td>Symbols ([2207])</td>
<td>Gets the symbols. (Inherited from ISymbolServer ([2205]).)</td>
</tr>
</tbody>
</table>

#### Reference

- [ISymbolProvider Interface](#) [2203]
- [TwinCAT.TypeSystem Namespace](#) [1622]

#### 6.11.93.1.1 ISymbolProvider.RootNamespaceName Property

Gets the name of the root namespace

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
string RootNamespaceName { get; }
```

**Property Value**

**Type:** String  
The namespace.

#### Reference

- [ISymbolProvider Interface](#) [2203]
- [TwinCAT.TypeSystem Namespace](#) [1622]
6.11.93.2 ISymbolProvider Methods

The ISymbolProvider type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDataTypesAsync</td>
<td>Gets the data types asynchronously. (Inherited from ISymbolServer.)</td>
</tr>
<tr>
<td>GetSymbolsAsync</td>
<td>Gets the symbols asynchronously (Inherited from ISymbolServer.)</td>
</tr>
</tbody>
</table>

### Reference

ISymbolProvider Interface

TwinCAT.TypeSystem Namespace

6.11.94 ISymbolServer Interface

Symbol Server Interface

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public interface ISymbolServer
```

The ISymbolServer type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypes</td>
<td>Gets the data types</td>
</tr>
<tr>
<td>DefaultValueEncoding</td>
<td>Gets the default value encoding.</td>
</tr>
<tr>
<td>Symbols</td>
<td>Gets the symbols.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDataTypesAsync</td>
<td>Gets the data types asynchronously.</td>
</tr>
<tr>
<td>GetSymbolsAsync</td>
<td>Gets the symbols asynchronously.</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace
6.11.94.1  **ISymbolServer Properties**

The **ISymbolServer** type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypes</td>
<td>Gets the data types</td>
</tr>
<tr>
<td>DefaultValueEncoding</td>
<td>Gets the default value encoding.</td>
</tr>
<tr>
<td>Symbols</td>
<td>Gets the symbols</td>
</tr>
</tbody>
</table>

### Reference

- **ISymbolServer Interface**
- **TwinCAT.TypeSystem Namespace**

#### 6.11.94.1.1  **ISymbolServer.DataTypes Property**

Gets the data types

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
IDataTypeCollection<IDataType> DataTypes { get; }
```

**Property Value**

Type: `IDataTypeCollection` `IDataType`.

The data types.

**Remarks**

This property reads the DataTypes synchronously, if the data is not available yet. For performance reasons, the asynchronous counterpart **GetDataTypesAsync(CancellationToken)** should be preferred for the first call.

**Reference**

- **ISymbolServer Interface**
- **TwinCAT.TypeSystem Namespace**
- **ISymbolServer.GetDataTypesAsync(CancellationToken)**

#### 6.11.94.1.2  **ISymbolServer.DefaultValueEncoding Property**

Gets the default value encoding.
Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
Encoding DefaultValueEncoding { get; }
```

**Property Value**

Type: `Encoding`

The default value encoding.

**Reference**

[ISymbolServer Interface](#)

TwinCAT.TypeSystem Namespace

---

6.11.94.1.3 **ISymbolServer.Symbols Property**

Gets the symbols.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
ISymbolCollection<ISymbol> Symbols { get; }
```

**Property Value**

Type: `ISymbolCollection`, `ISymbol`

The symbols.

**Remarks**

This property reads the Symbol information synchronously, if the data is not available yet. For performance reasons, the asynchronous counterpart `GetSymbolsAsync(CancellationToken)` should be preferred for the first call.

**Reference**

[ISymbolServer Interface](#)

TwinCAT.TypeSystem Namespace

[ISymbolServer.GetSymbolsAsync(CancellationToken)](#)

---

6.11.94.2 **ISymbolServer Methods**

The `ISymbolServer` type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDataTypesAsync</td>
<td>Gets the data types asynchronously.</td>
</tr>
<tr>
<td>GetSymbolsAsync</td>
<td>Gets the symbols asynchronously</td>
</tr>
</tbody>
</table>

### Reference

- `ISymbolServer Interface [1.205]`
- `TwinCAT.TypeSystem Namespace [1.162]`

#### 6.11.94.2.1 ISymbolServer.GetDataTypesAsync Method

Gets the data types asynchronously.

- **Namespace:** TwinCAT.TypeSystem [1.162]
- **Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.R.90bb1a4b36095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
Task<ResultDataTypes> GetDataTypesAsync(
    CancellationToken cancel
)
```

**Parameters**

- **cancel**
  - Type: `System.Threading.CancellationToken`
  - The cancellation token.

**Return Value**

- **Type:** `Task<ResultDataTypes>`
  - A task that represents the asynchronous 'GetDataTypes' operation. The `ResultDataTypes` parameter contains the data types (DataTypes [2349]) and the `ErrorCode` [992] after execution.

### Reference

- `ISymbolServer Interface [1.205]`
- `TwinCAT.TypeSystem Namespace [1.162]`
- `ISymbolServer.DataTypes [1.2206]`

#### 6.11.94.2.2 ISymbolServer.GetSymbolsAsync Method

Gets the symbols asynchronously.

- **Namespace:** TwinCAT.TypeSystem [1.162]
- **Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.R.90bb1a4b36095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
Task<ResultSymbols> GetSymbolsAsync(
    CancellationToken cancel
)
```

Parameters

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultSymbols>
A task that represents the asynchronous 'GetDataTypes' operation. The ResultSymbols parameter contains the data types (Symbols) and the ErrorCode after execution.

Reference

.ISymbolServer Interface
.TwinCAT.TypeSystem Namespace
.ISymbolServer.Symbols

6.11.95 ITypeAttribute Interface

Interface for ADS attributes

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface ITypeAttribute
```

The ITypeAttribute type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name [2210]</td>
<td>Name of the Attribute</td>
</tr>
<tr>
<td>Value [2210]</td>
<td>Gets the value of the attribute</td>
</tr>
</tbody>
</table>

Reference

.TwinCAT.TypeSystem Namespace

6.11.95.1 ITypeAttribute Properties

The ITypeAttribute type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name [2210]</td>
<td>Name of the Attribute</td>
</tr>
<tr>
<td>Value [2210]</td>
<td>Gets the value of the attribute</td>
</tr>
</tbody>
</table>

Reference

ITypeAttribute Interface [2209]

TwinCAT.TypeSystem Namespace [1622]

6.11.95.1.1 ITypeAttribute.Name Property

Name of the Attribute

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
string Name { get; }
```

Property Value

Type: String
The name.

Reference

ITypeAttribute Interface [2209]

TwinCAT.TypeSystem Namespace [1622]

6.11.95.1.2 ITypeAttribute.Value Property

Gets the value of the attribute

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
string Value { get; }
```

Property Value

Type: String
The value.
TwinCAT.Ads Namespaces

Reference

ITypeAttribute Interface [2209]

TwinCAT.TypeSystem Namespace [1622]

6.11.96 ITypeAttributeCollection Interface

Interface ITypeAttributeCollection

Namespace: TwinCAT.TypeSystem [1622]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface ITypeAttributeCollection : IList<ITypeAttribute>, ICollection<ITypeAttribute>, IEnumerable<ITypeAttribute>, IEnumerable
```

The ITypeAttributeCollection type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection[T]. (Inherited from ICollection[ITypeAttribute] [2209]..)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection[T] is read-only. (Inherited from ICollection[ITypeAttribute] [2209]..)</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the element at the specified index. (Inherited from IList[ITypeAttribute] [2209]..)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the String with the specified name.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection[T]. (Inherited from ICollection[ITypeAttribute] [2209]..)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection[T]. (Inherited from ICollection[ITypeAttribute] [2209]..)</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether this ITypeAttributeCollection contains the specified attribute.</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the ICollection[T] contains a specific value. (Inherited from ICollection[ITypeAttribute] [2209]..)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the ICollection[T] to an Array, starting at a particular Array index. (Inherited from ICollection[ITypeAttribute] [2209]..)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable[ITypeAttribute] [2209]..)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList[T]. (Inherited from IList[ITypeAttribute] [2209]..)</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList[T] at the specified index. (Inherited from IList[ITypeAttribute] [2209]..)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### TwinCAT.TypeSystem Namespace

#### 6.11.96.1 ITypeAttributeCollection Properties

The `ITypeAttributeCollection` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>ICollection</code>. (Inherited from <code>ICollection</code>. <code>ITypeAttribute</code>..)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the <code>ICollection</code> is read-only. (Inherited from <code>ICollection</code>. <code>ITypeAttribute</code>..)</td>
</tr>
<tr>
<td>Item[Int32]</td>
<td>Gets or sets the element at the specified index. (Inherited from <code>IList</code>. <code>ITypeAttribute</code>..)</td>
</tr>
<tr>
<td>Item[String]</td>
<td>Gets the <code>String</code> with the specified name.</td>
</tr>
</tbody>
</table>

**Reference**

- [ITypeAttributeCollection Interface](#)
- [TwinCAT.TypeSystem Namespace](#)

#### 6.11.96.1.1 ITypeAttributeCollection.Item Property

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item[Int32]</td>
<td>Gets or sets the element at the specified index. (Inherited from <code>IList</code>. <code>ITypeAttribute</code>..)</td>
</tr>
<tr>
<td>Item[String]</td>
<td>Gets the <code>String</code> with the specified name.</td>
</tr>
</tbody>
</table>

**Reference**

- [ITypeAttributeCollection Interface](#)
- [TwinCAT.TypeSystem Namespace](#)
**ITypeAttributeCollection.Item Property (String)**

Gets the `String` with the specified name.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
string this[string name] { get; }
```

**Parameters**

`name`  
Type: `System.String`  
The name.

**Return Value**

Type: `String`  
`System.String`.

**Reference**

ITypeAttributeCollection Interface

Item Overload

TwinCAT.TypeSystem Namespace

**6.11.96.2 ITypeAttributeCollection Methods**

The `ITypeAttributeCollection` type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;ITypeAttribute&gt;</code>)</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;ITypeAttribute&gt;</code>)</td>
</tr>
<tr>
<td>Contains(String) [2214]</td>
<td>Determines whether this <code>ITypeAttributeCollection</code> contains the specified attribute.</td>
</tr>
<tr>
<td>Contains(T) [2214]</td>
<td>Determines whether the <code>ICollection&lt;T&gt;</code> contains a specific value. (Inherited from <code>ICollection&lt;ITypeAttribute&gt;</code>)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the elements of the <code>ICollection&lt;T&gt;</code> to an <code>Array</code>, starting at a particular <code>Array</code> index. (Inherited from <code>ICollection&lt;ITypeAttribute&gt;</code>)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from <code>IEnumerable&lt;ITypeAttribute&gt;</code>)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the <code>IList&lt;T&gt;</code>. (Inherited from <code>IList&lt;ITypeAttribute&gt;</code>)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert</td>
<td>Inserts an item to the <code>IList&lt;T&gt;</code> at the specified index. (Inherited from <code>IList&lt;ITypeAttribute&gt;</code>)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the <code>ICollection&lt;T&gt;</code>. (Inherited from <code>ICollection&lt;ITypeAttribute&gt;</code>)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the <code>IList&lt;T&gt;</code> item at the specified index. (Inherited from <code>IList&lt;ITypeAttribute&gt;</code>)</td>
</tr>
<tr>
<td>TryGetAttribute</td>
<td>Tries to get the specified <code>ITypeAttribute</code>. (Inherited from <code>ITypeAttributeCollection</code>)</td>
</tr>
<tr>
<td>TryGetValue</td>
<td>Tries to get the specified Attribute value.</td>
</tr>
</tbody>
</table>

**Reference**

`ITypeAttributeCollection Interface` [2211]

`TwinCAT.TypeSystem Namespace` [1622]

### 6.11.96.2.1 `ITypeAttributeCollection.Contains Method` (String)

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String)</td>
<td>Determines whether this <code>ITypeAttributeCollection</code> contains the specified attribute. (Inherited from <code>ICollection&lt;ITypeAttribute&gt;</code>)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether the <code>ICollection&lt;T&gt;</code> contains a specific value. (Inherited from <code>ICollection&lt;ITypeAttribute&gt;</code>)</td>
</tr>
</tbody>
</table>

**Reference**

`ITypeAttributeCollection Interface` [2211]

`TwinCAT.TypeSystem Namespace` [1622]

### `ITypeAttributeCollection.Contains Method (String)`

Determines whether this `ITypeAttributeCollection` contains the specified attribute.

**Namespace:** `TwinCAT.TypeSystem` [1622]

**Assembly:** `TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)` Version: 5.0.294+Branch.releases-S.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
bool Contains(
    string name
)
```

**Parameters**

- **name**
  - Type: `System.String`
  - The name.
Return Value

Type: Boolean
ture if [contains] [the specified name]; otherwise, false.

Reference

ITypeAttributeCollection Interface [2211]
Contains Overload [2214]
TwinCAT.TypeSystem Namespace [1622]

6.11.96.2.2 ITypeAttributeCollection.TryGetAttribute Method

Tries to get the specified ITypeAttribute [2209]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
bool TryGetAttribute(
    string name,
    out ITypeAttribute attribute
)
```

Parameters

- name
  Type: System.String
  The name of the ITypeAttribute [2209].

- attribute
  Type: TwinCAT.TypeSystem.ITypeAttribute [2209]
  The attribute.

Return Value

Type: Boolean
ture if found, false otherwise.

Reference

ITypeAttributeCollection Interface [2211]
TwinCAT.TypeSystem Namespace [1622]

6.11.96.2.3 ITypeAttributeCollection.TryGetValue Method

Tries to get the specified Attribute value.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#
bool TryGetValue(
    string name,
    out string value
)

Parameters

name Type: System.String
The name.

value Type: System.String
The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

ITypeAttributeCollection Interface [2211]
TwinCAT.TypeSystem Namespace [1622]

6.11.97 ITypeMarshaler Interface

Interface ITypeMarshaler

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public interface ITypeMarshaler

The ITypeMarshaler type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanMarshal(Object)</td>
<td>Determines whether ADS can marshal the specified value</td>
</tr>
<tr>
<td>CanMarshal(Type)</td>
<td>Determines whether ADS can marshal the specified managed data type.</td>
</tr>
<tr>
<td>Marshal</td>
<td>Gets the byte size of the value when marshalled.</td>
</tr>
<tr>
<td>MarshalSize</td>
<td></td>
</tr>
<tr>
<td>Unmarshal</td>
<td></td>
</tr>
</tbody>
</table>
Remarks
The ITypeMarshaler is the common base interface for marshalling classes. It supports marshalling / unmarshalling of primitive managed values.

Reference
TwinCAT.TypeSystem Namespace [1622]

6.11.97.1 ITypeMarshaler Methods
The ITypeMarshaler [2216] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanMarshal(Object)</td>
<td>Determines whether ADS can marshal the specified value</td>
</tr>
<tr>
<td>CanMarshal(Type)</td>
<td>Determines whether ADS can marshal the specified managed data type.</td>
</tr>
<tr>
<td>Marshal</td>
<td></td>
</tr>
<tr>
<td>MarshalSize</td>
<td>Gets the byte size of the value when marshalled.</td>
</tr>
<tr>
<td>Unmarshal</td>
<td></td>
</tr>
</tbody>
</table>

Reference
ITypeMarshaler Interface [2216]
TwinCAT.TypeSystem Namespace [1622]

6.11.97.1.1 ITypeMarshaler.CanMarshal Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanMarshal(Object)</td>
<td>Determines whether ADS can marshal the specified value</td>
</tr>
<tr>
<td>CanMarshal(Type)</td>
<td>Determines whether ADS can marshal the specified managed data type.</td>
</tr>
</tbody>
</table>

Reference
ITypeMarshaler Interface [2216]
TwinCAT.TypeSystem Namespace [1622]

ITypeMarshaler.CanMarshal Method (Object)
Determines whether ADS can marshal the specified value

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
bool CanMarshal(
    Object value
)
```

Parameters

value Type: `System.Object`
The value.

Return Value

Type: `Boolean`
true if this instance can marshal the specified value; otherwise, false.

Reference

`ITypeMarshaler Interface [2216]`

`CanMarshal Overload [2217]`

`TwinCAT.TypeSystem Namespace [1622]`

`ITypeMarshaler.CanMarshal Method (Type)`

Determines whether ADS can marshal the specified managed data type.

Namespace: `TwinCAT.TypeSystem [1622]`
Assembly: `TwinCAT.Ads.Abstractions` (in `TwinCAT.Ads.Abstractions.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
bool CanMarshal(
    Type type
)
```

Parameters

type Type: `System.Type`
The Managed data type.

Return Value

Type: `Boolean`
true if this instance can marshal the specified managed type; otherwise, false.

Reference

`ITypeMarshaler Interface [2216]`

`CanMarshal Overload [2217]`

`TwinCAT.TypeSystem Namespace [1622]`
6.11.97.1.2 ITypeMarshaler.Marshal Method

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
int Marshal(
    Object value,
    Encoding encoding,
    Span destination
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>System.Object</td>
<td>The value.</td>
</tr>
<tr>
<td>encoding</td>
<td>System.Text.Encoding</td>
<td>The encoding.</td>
</tr>
<tr>
<td>destination</td>
<td>Span</td>
<td></td>
</tr>
</tbody>
</table>

Return Value

Type: Int32

Reference

ITypeMarshaler Interface
TwinCAT.TypeSystem Namespace

6.11.97.1.3 IType Marshaler.MarshalSize Method

Gets the byte size of the value when marshalled.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
int MarshalSize(
    Object value,
    Encoding encoding
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>System.Object</td>
<td>The value.</td>
</tr>
<tr>
<td>encoding</td>
<td>System.Text.Encoding</td>
<td>The encoding.</td>
</tr>
</tbody>
</table>
Return Value
Type: Int32
The marshal size of the value.

Reference
ITypeMarshaler Interface [2216]
TwinCAT.TypeSystem Namespace [1622]

6.11.97.1.4 ITypeMarshaler.Unmarshal Method

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#

```csharp
int Unmarshal(
    Type type,
    ReadOnlySpan source,
    void encoding,
    byte value
)
```

Parameters
- type Type: System.Type
- source Type: ReadOnlySpan
- encoding Type: System.Void
- value Type: System.Byte

Return Value
Type: Int32

Reference
ITypeMarshaler Interface [2216]
TwinCAT.TypeSystem Namespace [1622]

6.11.98 IUnionInstance Interface

Interface for an Instance of the IUnionType [2224].

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public interface IUnionInstance : ISymbol,
IAttributedInstance, IInstance, IBitSize
```

The `IUnionInstance` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from <code>IAttributedInstance</code>)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the <code>IDataType</code> in bits. (Inherited from <code>IBitSize</code>)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the <code>IInstance</code> in Bytes (Inherited from <code>IBitSize</code>)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Symbol/Datatype Category (Inherited from <code>ISymbol</code>)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the <code>Instance</code> (Inherited from <code>IInstance</code>)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the <code>IDataType</code> of the <code>IInstance</code> (Inherited from <code>IInstance</code>)</td>
</tr>
<tr>
<td>FieldInstances</td>
<td>Gets the field instances of the <code>Union</code></td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods .) (Inherited from <code>IInstance</code>)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods .)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full <code>DataType</code> but instead of some sort of bit mapping (Inherited from <code>IBitSize</code>)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <code>IBitSize</code>)</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from <code>ISymbol</code>)</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this <code>IInstance</code> is persistent. (Inherited from <code>ISymbol</code>)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the <code>Instance</code> represents a Pointer type (Pointer TO) (Inherited from <code>IInstance</code>)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from <code>ISymbol</code>)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only. (Inherited from <code>ISymbol</code>)</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from <code>ISymbol</code>)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the <code>Instance</code> represents a Reference type (REFERENCE TO) (Inherited from <code>IInstance</code>)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this <code>Instance</code> is static. (Inherited from <code>IInstance</code>)</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol (Inherited from <code>ISymbol</code>)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984]</td>
</tr>
<tr>
<td></td>
<td>(Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>typeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052].</td>
</tr>
<tr>
<td></td>
<td>(Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

### Reference

- TwinCAT.TypeSystem Namespace [1622]
- TwinCAT.TypeSystem.ISymbol [2176]

### 6.11.98.1 IUnionInstance Properties

The IUnionInstance [2220] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the IInstance [2052] (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>FieldInstances [2223]</td>
<td>Gets the field instances of the Union</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods .) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods .)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full</td>
</tr>
<tr>
<td></td>
<td>.DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [2052]).</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [2052]).</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this IInstance [2052] is static. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from BitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

### Reference

IUnionInstance Interface [2220]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.98.1.1 IUnionInstance.FieldInstances Property

Gets the field instances of the Union

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
ISymbolCollection<ISymbol> FieldInstances { get; }
```

**Property Value**

Type: ISymbolCollection [2185]. ISymbol [2176].
The field instances.

### Reference

IUnionInstance Interface [2220]

TwinCAT.TypeSystem Namespace [1622]
## 6.11.99 IUnionType Interface

Interface for an union data type.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public interface IUnionType : IDataType, IBitSize

The IUnionType type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType (Inherited from IDataType.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from IDataType.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment behind the variable declaration. (Inherited from IDataType.)</td>
</tr>
<tr>
<td>Fields</td>
<td>Gets a readonly collection of the Members of the IUnionType.</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the IDataType (Namespace + Name) (Inherited from IDataType.)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the Data Type (Inherited from IDataType.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this IDataType is a container type (Inherited from IDataType.)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this IDataType is a pointer type (Inherited from IDataType.)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IDataType is primitive (Inherited from IDataType.)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this IDataType is a reference type (Inherited from IDataType.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from IDataType.)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the IDataType exists. (Inherited from IDataType.)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType (Inherited from IBitSize.)</td>
</tr>
</tbody>
</table>
6.11.99.1 IUnionType Properties

The IUnionType type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the IDataType (Inherited from IDataType)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from IBitSize)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from IBitSize)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Data Type category (Inherited from IDataType)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment behind the variable declaration. (Inherited from IDataType)</td>
</tr>
<tr>
<td>Fields</td>
<td>Gets a readonly collection of the Members of the IUnionType</td>
</tr>
<tr>
<td>FullName</td>
<td>Gets the full name of the IDataType (Namespace + Name) (Inherited from IDataType)</td>
</tr>
<tr>
<td>Id</td>
<td>Gets the ID of the Data Type (Inherited from IDataType)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize)</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize)</td>
</tr>
<tr>
<td>IsContainer</td>
<td>Gets a value indicating whether this IDataType is a container type (Inherited from IDataType)</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Gets a value indicating whether this IDataType is a pointer type (Inherited from IDataType)</td>
</tr>
<tr>
<td>IsPrimitive</td>
<td>Gets a value indicating whether this IDataType is primitive (Inherited from IDataType)</td>
</tr>
<tr>
<td>IsReference</td>
<td>Gets a value indicating whether this IDataType is a reference type (Inherited from IDataType)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the Data Type (without namespace) (Inherited from IDataType)</td>
</tr>
<tr>
<td>Namespace</td>
<td>Gets the namespace string within the IDataType exists. (Inherited from IDataType)</td>
</tr>
<tr>
<td>Size</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType (Inherited from IBitSize)</td>
</tr>
</tbody>
</table>

Reference

IUnionType Interface [2224]
TwinCAT.TypeSystem Namespace [1622]

6.11.99.1.1 IUnionType.Fields Property

Gets a readonly collection of the Members [2040] of the IUnionType [2224].

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
IFieldCollection Fields { get; }
```

Property Value

Type: IFieldCollection [2042]
The members as readonly collection.

Remarks

If the IStructType [2162] is derived, only the extended members are returned. To get all supported members down the inheritance chain, use the AllMembers [2165] property.

Reference

IUnionType Interface [2224]
TwinCAT.TypeSystem Namespace [1622]

6.11.100 IValue Interface

Symbol Value Interface

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IValue
```

The IValue type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age [2228]</td>
<td>Gets the age of the value (last successful read of the value)</td>
</tr>
<tr>
<td>CachedRaw [2228]</td>
<td>Gets the cached Raw internal Data.</td>
</tr>
<tr>
<td>DataType [2228]</td>
<td>Gets the data type bound to this IValue</td>
</tr>
<tr>
<td>IsPrimitive [2229]</td>
<td>Gets a value indicating whether this IValue is a primitive value.</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### Symbol [2229]
- **Description**: Gets the symbol bound to this IValue.

### TimeStamp [2230]
- **Description**: Gets the Time stamp of the last successful read of the Value (local user time, UTC)

### UpdateMode [2230]
- **Description**: Gets the update mode (not implemented yet)

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read [2231]</td>
<td>Reads the value (via ADS)</td>
</tr>
<tr>
<td>ReadAsync [2231]</td>
<td>Reads the value (via ADS)</td>
</tr>
<tr>
<td>ResolveValue [2232]</td>
<td>Resolves the Value object to its primitive value.</td>
</tr>
<tr>
<td>TryResolveValue [2232]</td>
<td>Tries to resolves the Value object to its primitive value.</td>
</tr>
<tr>
<td>Write [2233]</td>
<td>Writes the value (via ADS)</td>
</tr>
<tr>
<td>WriteAsync [2233]</td>
<td>Writes the value (via ADS)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]

### 6.11.100.1 IValue Properties

The **IValue [2226]** type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age [2228]</td>
<td>Gets the age of the value (last successful read of the value)</td>
</tr>
<tr>
<td>CachedRaw [2228]</td>
<td>Gets the cached Raw internal Data.</td>
</tr>
<tr>
<td>DataType [2228]</td>
<td>Gets the data type bound to this IValue [2226]</td>
</tr>
<tr>
<td>IsPrimitive [2229]</td>
<td>Gets a value indicating whether this IValue [2226] is a primitive value.</td>
</tr>
<tr>
<td>Symbol [2229]</td>
<td>Gets the symbol bound to this IValue [2226].</td>
</tr>
<tr>
<td>TimeStamp [2230]</td>
<td>Gets the Time stamp of the last successful read of the Value (local user time, UTC)</td>
</tr>
<tr>
<td>UpdateMode [2230]</td>
<td>Gets the update mode (not implemented yet)</td>
</tr>
</tbody>
</table>

### Reference

IValue Interface [2226]

TwinCAT.TypeSystem Namespace [1622]
6.11.100.1.1  IValue.Age Property

Gets the age of the value (last successful read of the value)

**Namespace:**  TwinCAT.TypeSystem

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
TimeSpan Age { get; }
```

**Property Value**

Type:  `TimeSpan`

The age.

**Reference**

- IValue Interface
- TwinCAT.TypeSystem Namespace
- IValue.TimeStamp

6.11.100.1.2  IValue.CachedRaw Property

Gets the cached Raw internal Data.

**Namespace:**  TwinCAT.TypeSystem

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
byte[] CachedRaw { get; }
```

**Property Value**

Type:  `Byte`

The raw cached data.

**Reference**

- IValue Interface
- TwinCAT.TypeSystem Namespace

6.11.100.1.3  IValue.DataType Property

Gets the data type bound to this `IValue`

**Namespace:**  TwinCAT.TypeSystem

**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
## TwinCAT.Ads Namespaces

### Syntax

**C#**

```csharp
IDataType DataType { get; }
```

### Property Value

Type: `IDataType` [1986]

The type of the data.

### Reference

- `IValue Interface` [2226]
- `TwinCAT.TypeSystem Namespace` [1622]

### 6.11.100.1.4 IValue.IsPrimitive Property

Gets a value indicating whether this `IValue` [2226] is a primitive value.

**Namespace:** `TwinCAT.TypeSystem` [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
bool IsPrimitive { get; }
```

### Property Value

Type: `Boolean`

true if this instance is primitive; otherwise, false.

### Reference

- `IValue Interface` [2226]
- `TwinCAT.TypeSystem Namespace` [1622]

### 6.11.100.1.5 IValue.Symbol Property

Gets the symbol bound to this `IValue` [2226].

**Namespace:** `TwinCAT.TypeSystem` [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
ISymbol Symbol { get; }
```

### Property Value

Type: `ISymbol` [2176]

The symbol.
6.11.100.1.6 IValue.TimeStamp Property

Gets the Time stamp of the last successful read of the Value (local user time, UTC)

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
DateTimeOffset Timestamp { get; }
```

Property Value

Type: DateTimeOffset
The read time stamp.

Reference

IValue Interface [2226]
TwinCAT.TypeSystem Namespace [1622]

6.11.100.1.7 IValue.UpdateMode Property

Gets the update mode (not implemented yet)

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
ValueUpdateMode UpdateMode { get; }
```

Property Value

Type: ValueUpdateMode [150]
The update mode.

Reference

IValue Interface [2226]
TwinCAT.TypeSystem Namespace [1622]

6.11.100.2 IValue Methods

The [IValue [2226]] type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read [2231]</td>
<td>Reads the value (via ADS)</td>
</tr>
<tr>
<td>ReadAsync [2231]</td>
<td>Reads the value (via ADS)</td>
</tr>
<tr>
<td>ResolveValue [2232]</td>
<td>Resolves the Value object to its primitive value.</td>
</tr>
<tr>
<td>TryResolveValue [2232]</td>
<td>Tries to resolves the Value object to its primitive value.</td>
</tr>
<tr>
<td>Write [2233]</td>
<td>Writes the value (via ADS)</td>
</tr>
<tr>
<td>WriteAsync [2233]</td>
<td>Writes the value (via ADS)</td>
</tr>
</tbody>
</table>

### Reference

IValue Interface [2226]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.100.2.1 IValue.Read Method

Reads the value (via ADS)

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
void Read()
```

**Reference**

IValue Interface [2226]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.100.2.2 IValue.ReadAsync Method

Reads the value (via ADS)

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
Task<ResultAccess> ReadAsync(
    CancellationToken cancel)
```
TwinCAT.Ads Namespaces

Parameters

cancel  
Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task.ResultAccess [2556].
Task<ReadValueResult>.

Reference

IValue Interface [2226]

TwinCAT.TypeSystem Namespace [1622]

6.11.100.2.3  IValue.ResolveValue Method

Resolves the Value object to its primitive value.

Namespace:  TwinCAT.TypeSystem [1622]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  
Object ResolveValue(
    bool resolveEnumToPrimitive
)

Parameters

resolveEnumToPrimitive  
Type: System.Boolean
if set to true IEnumValue [2028]s are resolved to their primitives also.

Return Value

Type: Object
System.Object.

Remarks

If the value is not primitive, this method returns the IValue [2226] itself.

Reference

IValue Interface [2226]

TwinCAT.TypeSystem Namespace [1622]

6.11.100.2.4  IValue.TryResolveValue Method

Tries to resolves the Value object to its primitive value.

Namespace:  TwinCAT.TypeSystem [1622]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
bool TryResolveValue(
    bool resolveEnumToPrimitive,
    out Object value
)
```

Parameters

- `resolveEnumToPrimitive`: Type: System.Boolean
  If set to true, `IEnumValue` are resolved to their primitives also.
- `value`: Type: System.Object
  The value.

Return Value

Type: Boolean
true if value can be resolved, false otherwise.

Reference

IValue Interface

TwinCAT.TypeSystem Namespace

6.11.100.2.5 IValue.Write Method

Writes the value (via ADS)

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
void Write()
```

Reference

IValue Interface

TwinCAT.TypeSystem Namespace

6.11.100.2.6 IValue.WriteAsync Method

Writes the value (via ADS)

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
Task<ResultWriteAccess> WriteAsync(
    CancellationToken cancel)
```

Parameters

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultWriteAccess>.
Task<WriteValueResult>.

Reference

IValue Interface [2226]
TwinCAT.TypeSystem Namespace [1622]

6.11.101 IValueAccessorProvider Interface

Interface IValueAccessorProvider

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IValueAccessorProvider
```

The IValueAccessorProvider type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueAccessor [2235]</td>
<td>Gets the value accessor.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]

6.11.101.1 IValueAccessorProvider Properties

The IValueAccessorProvider [2234] type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueAccessor</td>
<td>Gets the value accessor.</td>
</tr>
</tbody>
</table>

Reference

IValueAccessorProvider Interface [2234]

TwinCAT.TypeSystem Namespace [1622]

6.11.101.1 IValueAccessorProvider.ValueAccessor Property

Gets the value accessor.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
IAccessorRawValue ValueAccessor { get; }
```

Property Value

Type: IAccessorRawValue [2544]
The value accessor.

Reference

IValueAccessorProvider Interface [2234]

TwinCAT.TypeSystem Namespace [1622]

6.11.102 IValueAnySymbol Interface

Interface IValueAnySymbol

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public interface IValueAnySymbol : ISymbol,
   IAttributedInstance, IInstance, IBitSize
```

The IValueAnySymbol type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the IInstance [2052] (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods .) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this IInstance [2052] is static. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAnyValue(Type)</td>
<td>Reads the value of this Value into a new created instance of the managed type</td>
</tr>
<tr>
<td>ReadAnyValue(Type, Int32)</td>
<td>Reads the value of this Value into a new created instance of the managed type</td>
</tr>
<tr>
<td>ReadAnyValueAsync</td>
<td>Reads the (AnyType) value asynchronously.</td>
</tr>
<tr>
<td>UpdateAnyValue(Object)</td>
<td>Reads the value of this Value into the specified managed value.</td>
</tr>
<tr>
<td>UpdateAnyValue(Object, Int32)</td>
<td>Reads the value of this Value into the specified managed value.</td>
</tr>
<tr>
<td>WriteAnyValue(Object)</td>
<td>Writes the value represented by the managed value to this Value.</td>
</tr>
<tr>
<td>WriteAnyValue(Object, Int32)</td>
<td>Writes the value represented by the managed value to this Value.</td>
</tr>
</tbody>
</table>

## Reference

TwinCAT.TypeSystem Namespace [1622]

TwinCAT.TypeSystem.ISymbol [2176]

### 6.11.102.1 IValueAnySymbol Properties

The IValueAnySymbol [2235] type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the Instance [2052] (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType [1986] of the Instance [2052]. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods .) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from Instance [2052].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the Instance [2052] represents a Pointer type (Pointer TO) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE TO) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this Instance [2052] is static. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052]. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

**Reference**

IValueAnySymbol Interface [2235]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.102.2 IValueAnySymbol Methods

The IValueAnySymbol [2235] type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAnyValue(Type) [2239]</td>
<td>Reads the value of this Value [2235] into a new created instance of the managed type</td>
</tr>
<tr>
<td>ReadAnyValue(Type, Int32) [2240]</td>
<td>Reads the value of this Value [2254] into a new created instance of the managed type</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAnyValueAsync</td>
<td>Reads the (AnyType) value asynchronously.</td>
</tr>
<tr>
<td>UpdateAnyValue(Object)</td>
<td>Reads the value of this Value into the specified managed value.</td>
</tr>
<tr>
<td>UpdateAnyValue(Object, Int32)</td>
<td>Reads the value of this Value into the specified managed value.</td>
</tr>
<tr>
<td>WriteAnyValue(Object)</td>
<td>Writes the value represented by the managed value to this Value.</td>
</tr>
<tr>
<td>WriteAnyValue(Object, Int32)</td>
<td>Writes the value represented by the managed value to this Value.</td>
</tr>
</tbody>
</table>

**Reference**

IValueAnySymbol Interface

TwinCAT.TypeSystem Namespace

### 6.11.102.2.1 IValueAnySymbol.ReadAnyValue Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadAnyValue(Type)</td>
<td>Reads the value of this Value into a new created instance of the managed type</td>
</tr>
<tr>
<td>ReadAnyValue(Type, Int32)</td>
<td>Reads the value of this Value into a new created instance of the managed type</td>
</tr>
</tbody>
</table>

**Reference**

IValueAnySymbol Interface

TwinCAT.TypeSystem Namespace

### IValueAnySymbol.ReadAnyValue Method (Type)

Reads the value of this Value into a new created instance of the managed type

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
Object ReadAnyValue(
    Type managedType
)
```
### Parameters

- **managedType**
  - **Type:** System.Type
  - The tp.

### Return Value

- **Type:** Object
  - Read value (System.Object).

### Reference

- IValueAnySymbol Interface [2235]
- ReadAnyValue Overload [2239]
- TwinCAT.TypeSystem Namespace [1622]
- IValueAnySymbol.WriteAnyValue(Object) [2243]
- IValueAnySymbol.UpdateAnyValue(Object) [2242]

### IValueAnySymbol.ReadAnyValue Method (Type, Int32)

Reads the value of this Value [2254] into a new created instance of the managed type

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstracts (in TwinCAT.Ads.Abstracts.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd6ca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
Object ReadAnyValue(
    Type managedType,
    int timeout
)
```

**Parameters**

- **managedType**
  - Type: System.Type
  - The tp.
- **timeout**
  - Type: System.Int32
  - The timeout in ms.

**Return Value**

- **Type:** Object
  - Read value (System.Object).

### Reference

- IValueAnySymbol Interface [2235]
- ReadAnyValue Overload [2239]
- TwinCAT.TypeSystem Namespace [1622]
- IValueAnySymbol.WriteAnyValue(Object) [2243]
6.11.102.2.2 **IValueAnySymbol.ReadAnyValueAsync Method**

Reads the (AnyType) value asynchronously.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
Task<ResultReadValueAccess> ReadAnyValueAsync(
    Type managedType,
    CancellationToken cancel
)
```

**Parameters**

- `managedType`  
  Type: `System.Type`  
  Managed type of the value to read.

- `cancel`  
  Type: `System.Threading.CancellationToken`  
  The cancellation token.

**Return Value**

Type: `Task<ResultReadValueAccess>` [2566].

A task object that is representing the asynchronous 'ReadAnyValue' operation. The result will be returned in a `ResultReadValueAccess` [2566], which contains the `Value` [2570] and the `ErrorCode` [2559].

**Reference**

- `IValueAnySymbol Interface` [2235]
- `TwinCAT.TypeSystem Namespace` [1622]

6.11.102.2.3 **IValueAnySymbol.UpdateAnyValue Method**

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>UpdateAnyValue(Object.)</code> [2242]</td>
<td>Reads the value of this <code>Value</code> [2254] into the specified managed value.</td>
</tr>
<tr>
<td><code>UpdateAnyValue(Object, Int32)</code> [2242]</td>
<td>Reads the value of this <code>Value</code> [2254] into the specified managed value.</td>
</tr>
</tbody>
</table>

**Reference**

- `IValueAnySymbol Interface` [2235]
- `TwinCAT.TypeSystem Namespace` [1622]
IValueAnySymbol.UpdateAnyValue Method (Object.)

Reads the value of this Value [2254] into the specified managed value.

Namespace:  TwinCAT.TypeSystem [1622]
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
void UpdateAnyValue(
    ref Object managedObject
)

Parameters

managedObject  Type: System.Object.
The managed object.

Return Value

Type:
Read value (System.Object).

Reference

IValueAnySymbol Interface [2235]
UpdateAnyValue Overload [2241]
TwinCAT.TypeSystem Namespace [1622]
IValueAnySymbol.ReadAnyValue(Type) [2239]
IValueAnySymbol.WriteAnyValue(Object) [2243]

IValueAnySymbol.UpdateAnyValue Method (Object., Int32)

Reads the value of this Value [2254] into the specified managed value.

Namespace:  TwinCAT.TypeSystem [1622]
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
void UpdateAnyValue(
    ref Object managedObject,
    int timeout
)

Parameters

managedObject  Type: System.Object.
The managed object.

timeout  Type: System.Int32
The timeout.
Return Value
Type: Read value (System.Object).

Reference
IValueAnySymbol Interface [2235]
UpdateAnyValue Overload [2241]
TwinCAT.TypeSystem Namespace [1622]
IValueAnySymbol.ReadAnyValue(Type) [2239]
IValueAnySymbol.WriteAnyValue(Object) [2243]

6.11.102.2.4 IValueAnySymbol.WriteAnyValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![WriteAnyValue(Object)]</td>
<td>Writes the value represented by the managed value to this Value [2254]</td>
</tr>
<tr>
<td>![WriteAnyValue(Object, Int32)]</td>
<td>Writes the value represented by the managed value to this Value [2254]</td>
</tr>
</tbody>
</table>

Reference
IValueAnySymbol Interface [2235]
TwinCAT.TypeSystem Namespace [1622]

IValueAnySymbol.WriteAnyValue Method (Object)

Writes the value represented by the managed value to this Value [2254]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  
void WriteAnyValue(
    Object managedValue
  )

Parameters

managedValue              Type: System.Object
                           The managed value.
IValueAnySymbol.WriteAnyValue Method (Object, Int32)

Writers the value represented by the managed value to this Value

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
void WriteAnyValue(
    Object managedValue,
    int timeout
)
```

**Parameters**

- `managedValue` Type: `System.Object`
  The managed value.
- `timeout` Type: `System.Int32`
  The timeout in ms.

**Reference**

IValueAnySymbol Interface [2235]

WriteAnyValue Overload [2243]

TwinCAT.TypeSystem Namespace [1622]

IValueAnySymbol.ReadAnyValue(Type) [2239]

IValueAnySymbol.UpdateAnyValue(Object) [2242]

6.11.103 IValueRawSymbol Interface

Interface IValueRawSymbol

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
The IValueRawSymbol type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the [DataType] in bits. (Inherited from IBitSize)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of the Type/Instance in Bytes (Inherited from BSize)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Symbol/Datatype Category</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the Instance</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the [DataType] of the Instance</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this [ValueSymbol] has a value</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance (without periods)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance (with periods)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full</td>
</tr>
<tr>
<td></td>
<td>DataType but instead of some sort of bit mapping</td>
</tr>
<tr>
<td>IsByteAligned</td>
<td>Indicates that the Size of the Object is Byte aligned</td>
</tr>
<tr>
<td>IsContainerType</td>
<td>Gets a value indicating whether this Symbol is a container type.</td>
</tr>
<tr>
<td>IsPersistent</td>
<td>Gets a value indicating whether this [Symbol] is persistent.</td>
</tr>
<tr>
<td>IsPointer</td>
<td>Indicates that the [Instance] represents a Pointer type (Pointer TO)</td>
</tr>
<tr>
<td>IsPrimitiveType</td>
<td>Gets a value indicating whether this instance is a primitive type.</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Indicates that this instance is read only.</td>
</tr>
<tr>
<td>IsRecursive</td>
<td>Gets a value indicating whether this instance is recursive.</td>
</tr>
<tr>
<td>IsReference</td>
<td>Indicates that the [Instance] represents a Reference type (REFERENCE TO)</td>
</tr>
<tr>
<td>IsStatic</td>
<td>Gets a value indicating whether this [Instance] is static.</td>
</tr>
<tr>
<td>Parent</td>
<td>Gets the parent Symbol</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>ValueAccessor [2248]</td>
<td>Gets the value accessor.</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadRawValue [2250]</td>
<td>Reads the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td>ReadRawValue(Int32) [2250]</td>
<td>Reads the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td>ReadRawValueAsync [2251]</td>
<td>Reads the raw value of the IValueSymbol [2254] (Ads Read / Write) asynchronously.</td>
</tr>
<tr>
<td>SetParent [2051]</td>
<td>Sets the parent of the Symbol (Inherited from IHierarchicalSymbol [2048].)</td>
</tr>
<tr>
<td>WriteRawValue(Bytes) [2252]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteRawValue(Bytes, Int32) [2252]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteRawValueAsync [2253]</td>
<td>Writes the raw value of the IValueSymbol [2254] (Ads Read / Write)</td>
</tr>
</tbody>
</table>

**Events**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged [2254]</td>
<td>Occurs when the RawValue of the IValueSymbol [2254] has changed.</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

TwinCAT.TypeSystem.IHierarchicalSymbol [2048]

6.11.103.1 IValueRawSymbol Properties

The IValueRawSymbol [2244] type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [► 1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [► 1980].)</td>
</tr>
<tr>
<td>BitSize [► 1984]</td>
<td>Gets the size of the IDataType [► 1986] in bits. (Inherited from IBitSize [► 1982].)</td>
</tr>
<tr>
<td>ByteSize [► 1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [► 1982].)</td>
</tr>
<tr>
<td>Category [► 2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [► 2176].)</td>
</tr>
<tr>
<td>Comment [► 2053]</td>
<td>Gets the comment of the IInstance [► 2052] (Inherited from IInstance [► 2052].)</td>
</tr>
<tr>
<td>DataType [► 2054]</td>
<td>Gets the IDataType [► 1986] of the IInstance [► 2052]. (Inherited from IInstance [► 2052].)</td>
</tr>
<tr>
<td>HasValue [► 2248]</td>
<td>Gets a value indicating whether this IValueSymbol [► 2254] has a value.</td>
</tr>
<tr>
<td>InstanceName [► 2054]</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from IInstance [► 2052].)</td>
</tr>
<tr>
<td>InstancePath [► 2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.) (Inherited from IInstance [► 2052].)</td>
</tr>
<tr>
<td>IsBitType [► 1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [► 1982].)</td>
</tr>
<tr>
<td>IsByteAligned [► 1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [► 1982].)</td>
</tr>
<tr>
<td>IsContainerType [► 2179]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol [► 2176].)</td>
</tr>
<tr>
<td>IsPersistent [► 2180]</td>
<td>Gets a value indicating whether this ISymbol [► 2176] is persistent. (Inherited from ISymbol [► 2176].)</td>
</tr>
<tr>
<td>IsPointer [► 2055]</td>
<td>Indicates that the IInstance [► 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [► 2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [► 2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [► 2176].)</td>
</tr>
<tr>
<td>IsReadOnly [► 2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [► 2176].)</td>
</tr>
<tr>
<td>IsRecursive [► 2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [► 2176].)</td>
</tr>
<tr>
<td>IsReference [► 2056]</td>
<td>Indicates that the IInstance [► 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [► 2052].)</td>
</tr>
<tr>
<td>IsStatic [► 2056]</td>
<td>Gets a value indicating whether this IInstance [► 2052] is static. (Inherited from IInstance [► 2052].)</td>
</tr>
<tr>
<td>Parent [► 2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [► 2176].)</td>
</tr>
<tr>
<td>Size [► 1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [► 1984] (Inherited from IBitSize [► 1982].)</td>
</tr>
<tr>
<td>SubSymbols [► 2182]</td>
<td>Gets the SubSymbols of the ISymbol [► 2176] (Inherited from ISymbol [► 2176].)</td>
</tr>
<tr>
<td>TypeName [► 2056]</td>
<td>Gets the name of the DataType [► 1986] that is used for this IInstance [► 2052]. (Inherited from IInstance [► 2052].)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueAccessor</td>
<td>Gets the value accessor.</td>
</tr>
<tr>
<td>ValueEncoding</td>
<td>Gets the value encoding. (Inherited from <code>IAttributedInstance</code>.)</td>
</tr>
</tbody>
</table>

### Reference

- `IValueRawSymbol Interface` ([2244]
- `TwinCAT.TypeSystem Namespace` ([1622]

### 6.11.103.1.1 `IValueRawSymbol.HasValue Property`

Gets a value indicating whether this `IValueSymbol` ([2254] has a value.

#### Syntax

**C#**

```csharp
bool HasValue { get; }
```

#### Property Value

**Type:** `Boolean`

- `true` if this instance has value; otherwise, `false`.

#### Remarks

A `VirtualSymbol` does not support values, but in terms of the `IValueSymbol` ([2254] definition, is a `IValueSymbol` ([2254]

### Reference

- `IValueRawSymbol Interface` ([2244]
- `TwinCAT.TypeSystem Namespace` ([1622]

### 6.11.103.1.2 `IValueRawSymbol.ValueAccessor Property`

Gets the value accessor.

#### Syntax

**C#**

```csharp
IAccessorRawValue ValueAccessor { get; }
```
Property Value

Type: I_accessorRawValue

The value accessor.

Reference

I_valueRawSymbol Interface

TwinCAT.TypeSystem Namespace

6.11.103.2 I_valueRawSymbol Methods

The I_valueRawSymbol type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadRawValue()</td>
<td>Reads the raw value of the I_valueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the raw value of the I_valueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>ReadRawValueAsyc</td>
<td>Reads the raw value of the I_valueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>SetParent</td>
<td>Sets the parent of the Symbol (Inherited from I_HierarchicalSymbol)</td>
</tr>
<tr>
<td>WriteRawValue(ByteArray)</td>
<td>Writes the raw value of the I_valueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteRawValue(ByteArray, Int32)</td>
<td>Writes the raw value of the I_valueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteRawValueAsyc</td>
<td>Writes the raw value of the I_valueSymbol (Ads Read / Write)</td>
</tr>
</tbody>
</table>

Reference

I_valueRawSymbol Interface

TwinCAT.TypeSystem Namespace

6.11.103.2.1 I_valueRawSymbol.ReadRawValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadRawValue()</td>
<td>Reads the raw value of the I_valueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the raw value of the I_valueSymbol (Ads Read / Write)</td>
</tr>
</tbody>
</table>
IValueRawSymbol.ReadRawValue Method

Reads the raw value of the IValueSymbol (Ads Read / Write)

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
byte[] ReadRawValue()

Field Value

Type: .Byte.
The raw value.

Return Value

Type: .Byte.
System.Byte[].

Reference

IValueRawSymbol Interface
ReadRawValue Overload
TwinCAT.TypeSystem Namespace

IValueRawSymbol.ReadRawValue Method (Int32)

Reads the raw value of the IValueSymbol (Ads Read / Write)

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
byte[] ReadRawValue(
    int timeout
)

Parameters

timeout Type: System.Int32
The timeout in ms.
Field Value
Type: .Byte
The raw value.

Return Value
Type: .Byte
System.Byte[].

Remarks
A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference
IValueRawSymbol Interface [2244]
ReadRawValue Overload [2249]
TwinCAT.TypeSystem Namespace [1622]

6.11.103.2.2 IValueRawSymbol.ReadRawValueAsync Method

Reads the raw value of the IValueSymbol [2254] (Ads Read / Write) asynchronously.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
Task<ResultReadRawAccess> ReadRawValueAsync(
    CancellationToken cancel
)

Parameters
cancel Type: System.Threading.CancellationToken
The cancellation token.

Field Value
Type: Task<ResultReadRawAccess> [2564].
The raw value.

Return Value
Type: Task<ResultReadRawAccess> [2564].
System.Byte[].

Reference
IValueRawSymbol Interface [2244]
TwinCAT.TypeSystem Namespace [1622]
6.11.103.2.3 IValueRawSymbol.WriteRawValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteRawValue(byte[])</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write)</td>
</tr>
<tr>
<td>WriteRawValue(byte, int32)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write)</td>
</tr>
</tbody>
</table>

Reference

IValueRawSymbol Interface [2244]

TwinCAT.TypeSystem Namespace [1622]

IValueRawSymbol.WriteRawValue Method (.Byte.)

Writes the raw value of the IValueSymbol (Ads Read / Write)

Namespace: TwinCAT.TypeSystem [1622]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
void WriteRawValue(
    byte[] rawValue
)
```

Parameters

rawValue Type: System.Byte

The value as byte array.

Reference

IValueRawSymbol Interface [2244]

WriteRawValue Overload [2252]

TwinCAT.TypeSystem Namespace [1622]

IValueRawSymbol.WriteRawValue Method (.Byte., Int32)

Writes the raw value of the IValueSymbol (Ads Read / Write)

Namespace: TwinCAT.TypeSystem [1622]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

```csharp
void WriteRawValue(
    byte[] rawValue,
    int timeout
)
```

Parameters

- `rawValue` Type: `System.Byte`
The value as byte array.

- `timeout` Type: `System.Int32`
The timeout.

Field Value

Type: `The value`

Remarks

A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

- `IValueRawSymbol Interface` [2244]
- `WriteRawValue Overload` [2252]
- `TwinCAT.TypeSystem Namespace` [1622]

### 6.11.103.2.4 IValueRawSymbol.WriteRawValueAsync Method

Writes the raw value of the `IValueSymbol` (Ads Read / Write)

**Namespace:** `TwinCAT.TypeSystem` [1622]

**Assembly:** `TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Task<ResultWriteAccess> WriteRawValueAsync(
    byte[] rawValue,
    CancellationToken cancel
)
```

Parameters

- `rawValue` Type: `System.Byte`
The value as byte array.

- `cancel` Type: `System.Threading.CancellationToken`
The cancellation token.
Return Value

Type: Task.ResultWriteAccess [2575].
A task that represents the asynchronous read operation. The ResultRead [1008] parameter contains the total number of bytes read into the buffer (ReadBytes [1010]) and the ErrorCode [992] after execution.

Reference

IValueRawSymbol Interface [2244]
TwinCAT.TypeSystem Namespace [1622]

6.11.103.3 IValueRawSymbol Events

The IValueRawSymbol [2244] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged [2254]</td>
<td>Occurs when the RawValue of the IValueSymbol [2254] has changed.</td>
</tr>
</tbody>
</table>

Reference

IValueRawSymbol Interface [2244]
TwinCAT.TypeSystem Namespace [1622]

6.11.103.3.1 IValueRawSymbol.RawValueChanged Event

Occurs when the RawValue of the IValueSymbol [2254] has changed.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
event EventHandler<RawValueChangedEventArgs> RawValueChanged
```

Value

Type: System.EventHandler<RawValueChangedEventArgs [2289]>

Reference

IValueRawSymbol Interface [2244]
TwinCAT.TypeSystem Namespace [1622]

6.11.104 IValueSymbol Interface

Interface for a ISymbol [2176] that supports values.
Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public interface IValueSymbol : IValueRawSymbol, IHierarchicalSymbol, ISymbol, IAttributedInstance, IInstance, IBitSize

The IValueSymbol type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessRights</td>
<td>Gets the access rights.</td>
</tr>
<tr>
<td>Attributes</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance.)</td>
</tr>
<tr>
<td>BitSize</td>
<td>Gets the size of the IDataType in bits. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>ByteSize</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes. (Inherited from IBitSize.)</td>
</tr>
<tr>
<td>Category</td>
<td>Gets the Symbol/Datatype Category. (Inherited from ISymbol.)</td>
</tr>
<tr>
<td>Comment</td>
<td>Gets the comment of the Instance. (Inherited from IInstance.)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets the connection that produces values for this IValueSymbol.</td>
</tr>
<tr>
<td>DataType</td>
<td>Gets the IDataType of the Instance. (Inherited from IInstance.)</td>
</tr>
<tr>
<td>HasValue</td>
<td>Gets a value indicating whether this IValueSymbol has a value. (Inherited from IValueRawSymbol.)</td>
</tr>
<tr>
<td>InstanceName</td>
<td>Gets the name of the instance without periods. (Inherited from IInstance.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the relative / absolute access path to the instance. (with periods.)</td>
</tr>
<tr>
<td>IsBitType</td>
<td>Gets a value indicating whether this instance is not basing on a full Data</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadRawValue</td>
<td>Reads the raw value of the IValueSymbol (Ads Read / Write) (Inherited from IValueRawSymbol [2244].)</td>
</tr>
<tr>
<td>ReadRawValue(Int32)</td>
<td>Reads the raw value of the IValueSymbol (Ads Read / Write) (Inherited from IValueRawSymbol [2244].)</td>
</tr>
<tr>
<td>ReadRawValueAsync</td>
<td>Reads the raw value of the IValueSymbol (Ads Read / Write) asynchronously. (Inherited from IValueRawSymbol [2244].)</td>
</tr>
<tr>
<td>ReadValue</td>
<td>Reads the Value of the IValueSymbol</td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the Value of the IValueSymbol</td>
</tr>
<tr>
<td>ReadValueAsync</td>
<td>Reads the Value of the IValueSymbol asynchronously.</td>
</tr>
<tr>
<td>SetParent</td>
<td>Sets the parent of the Symbol (Inherited from IHierarchicalSymbol [2048].)</td>
</tr>
<tr>
<td>TryReadValue</td>
<td>Reads the Value of the IValueSymbol</td>
</tr>
<tr>
<td>TryWriteValue</td>
<td>Writes the Value of the IValueSymbol</td>
</tr>
<tr>
<td>WriteRawValue(Byte)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from IValueRawSymbol [2244].)</td>
</tr>
<tr>
<td>WriteRawValue(Byte, Int32)</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from IValueRawSymbol [2244].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteRawValueAsync[2253]</td>
<td>Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from IValueRawSymbol [2244].)</td>
</tr>
<tr>
<td>WriteValue(Object)[2266]</td>
<td>Writes the Value of the IValueSymbol</td>
</tr>
<tr>
<td>WriteValue(Object, Int32)[2267]</td>
<td>Writes the Value of the IValueSymbol</td>
</tr>
<tr>
<td>WriteValueAsync[2268]</td>
<td>Writes the Value of the IValueSymbol</td>
</tr>
</tbody>
</table>

#### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged[2254]</td>
<td>Occurs when the RawValue of the IValueSymbol has changed. (Inherited from IValueRawSymbol [2244].)</td>
</tr>
<tr>
<td>ValueChanged[2269]</td>
<td>Occurs when the (Primitive) value of the IValueSymbol has changed.</td>
</tr>
</tbody>
</table>

#### Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollValuesAnnotate d((I Observ able Unit.))[1111]</td>
<td>Overloaded. Polls the values as ValueChangedEventArgs[2439] sequence annotated value on trigger sequence (Defined by ValueSymbolExtensions[1106].)</td>
</tr>
<tr>
<td>PollValuesAnnotate d(TimeSpan)[1112]</td>
<td>Overloaded. Polls the values as ValueChangedEventArgs[2439] sequence with a specified period time. (Defined by ValueSymbolExtensions[1106].)</td>
</tr>
<tr>
<td>WhenValueChanged[1113]</td>
<td>Gets an observable sequence when the value of the IValueSymbol has changed. (Defined by ValueSymbolExtensions[1106].)</td>
</tr>
<tr>
<td>WriteValues((I Observable.Object.))[1117]</td>
<td>Overloaded. Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol. (Defined by ValueSymbolExtensions[1106].)</td>
</tr>
<tr>
<td>WriteValues((I Observable.Object., Action.Exception.))[1118]</td>
<td>Overloaded. Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol. (Defined by ValueSymbolExtensions[1106].)</td>
</tr>
<tr>
<td>WriteValues((I Observable.Object., CancellationToken))[1119]</td>
<td>Overloaded. Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol. (Defined by ValueSymbolExtensions[1106].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteValues(IQueryable&lt;Action.Exception, CancellationToken&gt;, IObservableObject)</td>
<td>Overloaded. Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol. (Defined by ValueSymbolExtensions)</td>
</tr>
</tbody>
</table>

### Reference
- TwinCAT.TypeSystem Namespace [1622]
- TwinCAT.TypeSystem.IValueRawSymbol [2244]

### 6.11.104.1 IValueSymbol Properties

The IValueSymbol [2254] type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessRights [2253]</td>
<td>Gets the access rights.</td>
</tr>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980])</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982])</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982])</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176])</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the Instance [2052] (Inherited from IInstance [2052])</td>
</tr>
<tr>
<td>Connection [2260]</td>
<td>Gets the connection that produces values for this IValueSymbol [2254]</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the Instance [2052] (Inherited from IInstance [2052])</td>
</tr>
<tr>
<td>HasValue [2248]</td>
<td>Gets a value indicating whether this IValueSymbol [2254] has a value (Inherited from IValueRawSymbol [2244])</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods .) (Inherited from IInstance [2052])</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods .) (Inherited from IInstance [2052])</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982])</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982])</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol [2176])</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from ISymbol [2176])</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the Instance [2052] represents a Pointer type (Pointer TO) (Inherited from Symbol [2176].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from Symbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from Symbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from Symbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the Instance [2052] represents a Reference type (REFERENCE TO) (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this Instance [2052] is static. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>NotificationSettings [2260]</td>
<td>Gets or sets the notification settings.</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from Symbol [2176].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from BitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the Symbol [2176] (Inherited from Symbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this Instance [2052]. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>ValueAccessor [2248]</td>
<td>Gets the value accessor. (Inherited from ValueRawSymbol [2244].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from AttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

**Reference**

IValueSymbol Interface [2254]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.104.1.1 IValueSymbol.AccessRights Property

Gets the access rights.

**Namespace**: TwinCAT.TypeSystem [1622]
**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
SymbolAccessRights AccessRights { get; }
```

**Property Value**

Type: SymbolAccessRights [2396]  
The access rights.
Reference

IValueSymbol Interface [2254]
TwinCAT.TypeSystem Namespace [1622]

6.11.104.1.2 IValueSymbol.Connection Property

Gets the connection that produces values for this IValueSymbol [2254]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
IConnection Connection { get; }

Property Value

Type: IConnection [74]
The connection object.

Reference

IValueSymbol Interface [2254]
TwinCAT.TypeSystem Namespace [1622]

6.11.104.1.3 IValueSymbol.NotificationSettings Property

Gets or sets the notification settings.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
INotificationSettings NotificationSettings { get; set; }

Property Value

Type: INotificationSettings [972]
The notification settings.

Remarks

The NotificationSettings will be inherited from Parent [2182] if the setting is not overwritten.

Reference

IValueSymbol Interface [2254]
TwinCAT.TypeSystem Namespace [1622]
### 6.11.104.2 IValueSymbol Methods

The **IValueSymbol** type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ReadRawValue</strong></td>
<td>Reads the raw value of the <strong>IValueSymbol</strong> (Ads Read / Write) (Inherited from <strong>IValueRawSymbol</strong>.)</td>
</tr>
<tr>
<td><strong>ReadRawValue(Int32)</strong></td>
<td>Reads the raw value of the <strong>IValueSymbol</strong> (Ads Read / Write) (Inherited from <strong>IValueRawSymbol</strong>.)</td>
</tr>
<tr>
<td><strong>ReadRawValueAsync</strong></td>
<td>Reads the raw value of the <strong>IValueSymbol</strong> asynchronously. (Inherited from <strong>IValueRawSymbol</strong>.)</td>
</tr>
<tr>
<td><strong>ReadValue</strong></td>
<td>Reads the Value of the <strong>IValueSymbol</strong>.</td>
</tr>
<tr>
<td><strong>ReadValue(Int32)</strong></td>
<td>Reads the Value of the <strong>IValueSymbol</strong>.</td>
</tr>
<tr>
<td><strong>ReadValueAsync</strong></td>
<td>Reads the Value of the <strong>IValueSymbol</strong> asynchronously.</td>
</tr>
<tr>
<td><strong>SetParent</strong></td>
<td>Sets the parent of the Symbol (Inherited from <strong>IHierarchicalSymbol</strong>.)</td>
</tr>
<tr>
<td><strong>TryReadValue</strong></td>
<td>Reads the Value of the <strong>IValueSymbol</strong>.</td>
</tr>
<tr>
<td><strong>TryWriteValue</strong></td>
<td>Writes the Value of the <strong>IValueSymbol</strong>.</td>
</tr>
<tr>
<td><strong>WriteRawValue(Byte, Int32)</strong></td>
<td>Writes the raw value of the <strong>IValueSymbol</strong> (Ads Read / Write) (Inherited from <strong>IValueRawSymbol</strong>.)</td>
</tr>
<tr>
<td><strong>WriteRawValueAsync(Byte, Int32)</strong></td>
<td>Writes the raw value of the <strong>IValueSymbol</strong> asynchronously. (Inherited from <strong>IValueRawSymbol</strong>.)</td>
</tr>
<tr>
<td><strong>WriteValue(Object)</strong></td>
<td>Writes the Value of the <strong>IValueSymbol</strong>.</td>
</tr>
<tr>
<td><strong>WriteValue(Object, Int32)</strong></td>
<td>Writes the Value of the <strong>IValueSymbol</strong>.</td>
</tr>
<tr>
<td><strong>WriteValueAsync</strong></td>
<td>Writes the Value of the <strong>IValueSymbol</strong>.</td>
</tr>
</tbody>
</table>

#### Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PollValuesAnnotated</strong></td>
<td>Overloaded. Polls the values as <strong>ValueChangedEventArgs</strong> sequence annotated value on trigger sequence (Defined by <strong>ValueSymbolExtensions</strong>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>PollValuesAnnotated(TimeSpan)</td>
<td>Overloaded. Polls the values as <code>ValueChangedEventArgs</code> sequence with a specified period time. (Defined by <code>ValueSymbolExtensions</code>.)</td>
</tr>
<tr>
<td>WhenValueChanged</td>
<td>Gets an observable sequence when the value of the <code>IValueSymbol</code> has changed. (Defined by <code>ValueSymbolExtensions</code>.)</td>
</tr>
<tr>
<td>WriteValues(IObservable.Object.)</td>
<td>Overloaded. Subscribes the <code>IValueSymbol</code> to an observable sequence of values and writes them to the <code>IValueSymbol</code>. (Defined by <code>ValueSymbolExtensions</code>.)</td>
</tr>
<tr>
<td>WriteValues(IObservable.Object., Action.Exception.)</td>
<td>Overloaded. Subscribes the <code>IValueSymbol</code> to an observable sequence of values and writes them to the <code>IValueSymbol</code>. (Defined by <code>ValueSymbolExtensions</code>.)</td>
</tr>
<tr>
<td>WriteValues(IObservable.Object., CancellationToken)</td>
<td>Overloaded. Subscribes the <code>IValueSymbol</code> to an observable sequence of values and writes them to the <code>IValueSymbol</code>. (Defined by <code>ValueSymbolExtensions</code>.)</td>
</tr>
<tr>
<td>WriteValues(IObservable.Object., Action.Exception., CancellationToken)</td>
<td>Overloaded. Subscribes the <code>IValueSymbol</code> to an observable sequence of values and writes them to the <code>IValueSymbol</code>. (Defined by <code>ValueSymbolExtensions</code>.)</td>
</tr>
</tbody>
</table>

Reference

`IValueSymbol Interface` [2254]

`TwinCAT.TypeSystem Namespace` [1622]

### 6.11.104.2.1 `IValueSymbol.ReadValue Method`

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadValue()</td>
<td>Reads the Value of the <code>IValueSymbol</code></td>
</tr>
<tr>
<td>ReadValue(Int32)</td>
<td>Reads the Value of the <code>IValueSymbol</code></td>
</tr>
</tbody>
</table>

Reference

`IValueSymbol Interface` [2254]

`TwinCAT.TypeSystem Namespace` [1622]
IValueSymbol.ReadValue Method

Reads the Value of the IValueSymbol [2254]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
Object ReadValue()

Field Value

Type: Object
The value.

Return Value

Type: Object
System.Object.

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly.

Reference

IValueSymbol Interface [2254]
ReadValue Overload [2262]
TwinCAT.TypeSystem Namespace [1622]

IValueSymbol.ReadValue Method (Int32)

Reads the Value of the IValueSymbol [2254]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
Object ReadValue(
    int timeout
)

Parameters

timeout Type: System.Int32
The timeout in ms.
Field Value

Type: Object
The value.

Return Value

Type: Object
System.Object.

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

IValueSymbol Interface
ReadValue Overload
TwinCAT.TypeSystem Namespace

6.11.104.2.2 IValueSymbol.ReadValueAsync Method

Reads the Value of the IValueSymbol as asynchronously.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions.dll (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

Task<ResultReadValueAccess> ReadValueAsync(
    CancellationToken cancel
)

Parameters

cancel Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultReadValueAccess>, A tasks that represents the asynchronous ‘ReadValue’ operation. The read result is stored in the ResultReadValueAccess return value and contains the Value and the ErrorCode.

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly.
6.11.104.2.3 IValueSymbol.TryReadValue Method

Reads the Value of the IValueSymbol [› 2254]

Namespace: TwinCAT.TypeSystem [› 1622]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
int TryReadValue(
    int timeout,
    out Object value
)
```

Parameters

timeout            Type: System.Int32
                   The timeout in ms.

value              Type: System.Object
                   The symbol value.

Return Value

Type: Int32
The error code.

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [› 2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

IValueSymbol Interface [› 2254]

TwinCAT.TypeSystem Namespace [› 1622]

6.11.104.2.4 IValueSymbol.TryWriteValue Method

Writes the Value of the IValueSymbol [› 2254]

Namespace: TwinCAT.TypeSystem [› 1622]

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
int TryWriteValue(
    Object value,
    int timeout
)
```

Parameters

- **value**
  Type: `System.Object`
  The value.

- **timeout**
  Type: `System.Int32`
  The timeout in ms.

Return Value

Type: `Int32`
The error code.

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the `ISymbolLoader` settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written in dynamic mode: A Value that represents the value will be accepted also. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

- `IValueSymbol Interface [2254]`
- `TwinCAT.TypeSystem Namespace [1622]`

### 6.11.104.2.5 IValueSymbol.WriteValue Method

**Overload List**

<table>
<thead>
<tr>
<th>Overload</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>WriteValue(Object)</code></td>
<td>Writes the Value of the <code>IValueSymbol [2254]</code></td>
</tr>
<tr>
<td><code>WriteValue(Object, Int32)</code></td>
<td>Writes the Value of the <code>IValueSymbol [2254]</code></td>
</tr>
</tbody>
</table>

Reference

- `IValueSymbol Interface [2254]`
- `TwinCAT.TypeSystem Namespace [1622]`

**IValueSymbol.WriteValue Method (Object)**

Writes the Value of the `IValueSymbol [2254]`

**Namespace:** `TwinCAT.TypeSystem [1622]`
**Assembly:** `TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

**C#**

```csharp
void WriteValue(
    Object value
)
```

**Parameters**

- `value`  
  Type: `System.Object`  
  The value.

**Remarks**

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the `ISymbolLoader` settings what will happen. In non dynamic modes: Only `byte` Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also.

**Reference**

- `IValueSymbol Interface` [2254]
- `WriteValue Overload` [2266]
- `TwinCAT.TypeSystem Namespace` [1622]

**IValueSymbol.WriteValue Method (Object, Int32)**

Writes the Value of the `IValueSymbol`  

**Namespace:** `TwinCAT.TypeSystem` [1622]  
**Assembly:** `TwinCAT.Ads.Abstractions` (in `TwinCAT.Ads.Abstractions.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
void WriteValue(
    Object value,
    int timeout
)
```

**Parameters**

- `value`  
  Type: `System.Object`  
  The value.
- `timeout`  
  Type: `System.Int32`  
  The timeout in ms.

**Remarks**

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the `ISymbolLoader` settings what will happen. In non dynamic modes: Only `byte` Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.
6.11.104.2.6 IValueSymbol.WriteValueAsync Method

Writes the Value of the IValueSymbol [2254] Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

Task<ResultWriteAccess> WriteValueAsync(
    Object value,
    CancellationToken cancel)

Parameters

value
Type: System.Object
The value.
cancel
Type: System.Threading.CancellationToken
The cancellation token.

Return Value

Type: Task<ResultWriteAccess> [2575].
A tasks that represents the asynchronous 'ReadValue' operation. The read result is stored in the ResultWriteAccess [2575] return value and contains the ErrorCode [2559].

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader [2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also.

Reference

IValueSymbol Interface [2254]
TwinCAT.TypeSystem Namespace [1622]

6.11.104.3 IValueSymbol Events

The IValueSymbol [2254] type exposes the following members.
### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawValueChanged</td>
<td>Occurs when the RawValue of the IValueSymbol has changed. (Inherited from IValueRawSymbol.)</td>
</tr>
<tr>
<td>ValueChanged</td>
<td>Occurs when the (Primitive) value of the IValueSymbol has changed.</td>
</tr>
</tbody>
</table>

#### Reference

IValueSymbol Interface

TwinCAT.TypeSystem Namespace

---

### 6.11.104.3.1 IValueSymbol.ValueChanged Event

Occurs when the (Primitive) value of the IValueSymbol has changed.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
event EventHandler<ValueChangedEventArgs> ValueChanged
```

**Value**

Type: System.EventHandlerValueChangedEventArgs

**Reference**

IValueSymbol Interface

TwinCAT.TypeSystem Namespace

---

### 6.11.105 IVirtualStructInstance Interface

Virtual Struct instance interface.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public interface IVirtualStructInstance : IStructInstance, ISymbol, IAttributedInstance, IInstance, IBitSize
```

The IVirtualStructInstance type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes [1982]</td>
<td>Gets the Type Attributes. (Inherited from IAttributedInstance [1980].)</td>
</tr>
<tr>
<td>BitSize [1984]</td>
<td>Gets the size of the IDataType [1986] in bits. (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>ByteSize [1984]</td>
<td>Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>Category [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Comment [2053]</td>
<td>Gets the comment of the IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>DataType [2054]</td>
<td>Gets the IDataType [1986] of the IInstance [2052]. (Inherited from Instance [2052].)</td>
</tr>
<tr>
<td>HasRpcMethods [2161]</td>
<td>Gets a value indicating whether this instance has RPC methods (Inherited from IStructInstance [2158].)</td>
</tr>
<tr>
<td>InstanceName [2054]</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>InstancePath [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsBitType [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsByteAligned [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Indicates that this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this IInstance [2052] is static. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>MemberInstances [2162]</td>
<td>Gets the member instances of the Struct Instance [2158]. (Inherited from IStructInstance [2158].)</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TypeName</strong> [1986]</td>
<td>Gets the name of the <strong>DataType</strong> [1986] that is used for this <strong>Instance</strong> [2052]. (Inherited from <strong>Instance</strong> [2052].)</td>
</tr>
<tr>
<td><strong>ValueEncoding</strong> [1982]</td>
<td>Gets the value encoding. (Inherited from <strong>AttributedInstance</strong> [1980].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddMember [2273]</td>
<td>Adds the member.</td>
</tr>
</tbody>
</table>

### Remarks

Virtual struct instance are used to create a TreeView from the flat list of symbols.

### Reference

TwinCAT.TypeSystem Namespace [1622]

TwinCAT.TypeSystem.IStructInstance [2158]

### 6.11.105.1 IVirtualStructInstance Properties

The **IVirtualStructInstance** [2269] type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attributes</strong> [1982]</td>
<td>Gets the Type Attributes. (Inherited from <strong>AttributedInstance</strong> [1980].)</td>
</tr>
<tr>
<td><strong>BitSize</strong> [1982]</td>
<td>Gets the size of the <strong>DataType</strong> [1986] in bits. (Inherited from <strong>BitSize</strong> [1982].)</td>
</tr>
<tr>
<td><strong>ByteSize</strong> [1982]</td>
<td>Gets the (aligned) size of the <strong>Type/Instance</strong> in Bytes (Inherited from <strong>BitSize</strong> [1982].)</td>
</tr>
<tr>
<td><strong>Category</strong> [2179]</td>
<td>Gets the Symbol/Datatype Category (Inherited from <strong>Symbol</strong> [2176].)</td>
</tr>
<tr>
<td><strong>Comment</strong> [2053]</td>
<td>Gets the comment of the <strong>Instance</strong> [2052] (Inherited from <strong>Instance</strong> [2052].)</td>
</tr>
<tr>
<td><strong>DataType</strong> [2054]</td>
<td>Gets the <strong>DataType</strong> [1986] of the <strong>Instance</strong> [2052]. (Inherited from <strong>Instance</strong> [2052].)</td>
</tr>
<tr>
<td><strong>HasRpcMethods</strong> [2161]</td>
<td>Gets a value indicating whether this instance has RPC methods (Inherited from <strong>StructInstance</strong> [2158].)</td>
</tr>
<tr>
<td><strong>InstanceName</strong> [2054]</td>
<td>Gets the name of the instance (without periods (.)) (Inherited from <strong>Instance</strong> [2052].)</td>
</tr>
<tr>
<td><strong>InstancePath</strong> [2055]</td>
<td>Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from <strong>Instance</strong> [2052].)</td>
</tr>
<tr>
<td><strong>IsBitType</strong> [1984]</td>
<td>Gets a value indicating whether this instance is not basing on a full <strong>DataType</strong> but instead of some sort of bit mapping (Inherited from <strong>BitSize</strong> [1982].)</td>
</tr>
<tr>
<td><strong>IsByteAligned</strong> [1985]</td>
<td>Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from <strong>BitSize</strong> [1982].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsContainerType [2179]</td>
<td>Gets a value indicating whether this Symbol is a container type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPersistent [2180]</td>
<td>Gets a value indicating whether this ISymbol [2176] is persistent. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsPointer [2055]</td>
<td>Indicates that the IInstance [2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsPrimitiveType [2180]</td>
<td>Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReadOnly [2181]</td>
<td>Gets a value indicating whether this instance is read only. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsRecursive [2181]</td>
<td>Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>IsReference [2056]</td>
<td>Indicates that the IInstance [2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>IsStatic [2056]</td>
<td>Gets a value indicating whether this IInstance [2052] is static. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>MemberInstances [2162]</td>
<td>Gets the member instances of the Struct Instance [2158]. (Inherited from IStructInstance [2158].)</td>
</tr>
<tr>
<td>Parent [2182]</td>
<td>Gets the parent Symbol (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>Size [1985]</td>
<td>Gets the size of the object in bytes or Bits dependant on IsBitType [1984] (Inherited from IBitSize [1982].)</td>
</tr>
<tr>
<td>SubSymbols [2182]</td>
<td>Gets the SubSymbols of the ISymbol [2176] (Inherited from ISymbol [2176].)</td>
</tr>
<tr>
<td>TypeName [2056]</td>
<td>Gets the name of the DataType [1986] that is used for this IInstance [2052]. (Inherited from IInstance [2052].)</td>
</tr>
<tr>
<td>ValueEncoding [1982]</td>
<td>Gets the value encoding. (Inherited from IAttributedInstance [1980].)</td>
</tr>
</tbody>
</table>

Reference

IVirtualStructInstance Interface [2269]

TwinCAT.TypeSystem Namespace [1622]

6.11.105.2 IVirtualStructInstance Methods

The IVirtualStructInstance [2269] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddMember [2279]</td>
<td>Adds the member.</td>
</tr>
</tbody>
</table>

Reference

IVirtualStructInstance Interface [2269]

TwinCAT.TypeSystem Namespace [1622]
6.11.105.2.1 IVirtualStructInstance.AddMember Method

Adds the member.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
bool AddMember(
    ISymbol memberInstance,
    IVirtualStructInstance parent
)
```

**Parameters**

- **memberInstance**
  
  Type: TwinCAT.TypeSystem.ISymbol
  
  The member instance.

- **parent**
  
  Type: TwinCAT.TypeSystem.IVirtualStructInstance
  
  The parent struct instance. Usually the this pointer.

**Return Value**

Type: Boolean

**Reference**

- IVirtualStructInstance Interface
- TwinCAT.TypeSystem Namespace

6.11.106 MarshalException Class

Common Marshalling Exception

**Inheritance Hierarchy**

- System.Object
  - System.Exception
    - TwinCAT.AdsException
      - TwinCAT.TypeSystem.MarshalException

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
[SerializableAttribute]
public sealed class MarshalException : AdsException
```

The MarshalException type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MarshalException</td>
<td>Initializes a new instance of the MarshalException class.</td>
</tr>
<tr>
<td>MarshalException(String)</td>
<td>Initializes a new instance of the MarshalException class.</td>
</tr>
<tr>
<td>MarshalException(DataType)</td>
<td>Initializes a new instance of the MarshalException class.</td>
</tr>
<tr>
<td>MarshalException(String, Exception)</td>
<td>Initializes a new instance of the AdsException class.</td>
</tr>
<tr>
<td>MarshalException(DataType, Object)</td>
<td>Initializes a new instance of the MarshalException class.</td>
</tr>
<tr>
<td>MarshalException(DataType, Type)</td>
<td>Initializes a new instance of the MarshalException class.</td>
</tr>
<tr>
<td>MarshalException(DataType, Type, String)</td>
<td>Initializes a new instance of the MarshalException class.</td>
</tr>
<tr>
<td>MarshalException(Iinstance, Type, MemberInfo)</td>
<td>Initializes a new instance of the MarshalException class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception.</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]

6.11.106.1 MarshalException Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MarshalException([2276])</td>
<td>Initializes a new instance of the MarshalException [2273] class.</td>
</tr>
<tr>
<td>MarshalException(String) [2276]</td>
<td>Initializes a new instance of the MarshalException [2273] class.</td>
</tr>
<tr>
<td>MarshalException(IDataType) [2277]</td>
<td>Initializes a new instance of the MarshalException [2273] class.</td>
</tr>
<tr>
<td>MarshalException(String, Exception) [2277]</td>
<td>Initializes a new Instance of the AdsException class.</td>
</tr>
<tr>
<td>MarshalException(IDataType, Object) [2278]</td>
<td>Initializes a new instance of the MarshalException [2273] class.</td>
</tr>
<tr>
<td>MarshalException(IDataType, Type) [2278]</td>
<td>Initializes a new instance of the MarshalException [2273] class.</td>
</tr>
<tr>
<td>MarshalException(IDataType, Type, String) [2279]</td>
<td>Initializes a new instance of the MarshalException [2273] class.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MarshalException(IIstance, Type, MemberInfo])</td>
<td>Initializes a new instance of the MarshalException class.</td>
</tr>
</tbody>
</table>

**Reference**

MarshalException Class [2273]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.106.1.1 MarshalException Constructor

Initializes a new instance of the MarshalException class.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public MarshalException()
```

**Reference**

MarshalException Class [2273]
MarshalException Overload [2275]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.106.1.2 MarshalException Constructor (String)

Initializes a new instance of the MarshalException class.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public MarshalException(
    string message)
```

**Parameters**

- **message**
  - Type: System.String
  - The message.

**Reference**

MarshalException Class [2273]
6.11.106.1.3 MarshalException Constructor (IDataType)

Initializes a new instance of the MarshalException class.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public MarshalException(
    IDataType source
)
```

**Parameters**

- **source**  
  Type: TwinCAT.TypeSystem.IDataType  
  The source dataType.

**Reference**

- MarshalException Class
- MarshalException Overload
- TwinCAT.TypeSystem Namespace

6.11.106.1.4 MarshalException Constructor (String, Exception)

Initializes a new instance of the AdsException class.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public MarshalException(
    string message,
    Exception innerException
)
```

**Parameters**

- **message**  
  Type: System.String  
  The message.

- **innerException**  
  Type: System.Exception  
  The inner exception.

**Reference**

- MarshalException Class
6.11.106.1.5 MarshalException Constructor (IDataType, Object)

Initializes a new instance of the MarshalException class.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public MarshalException(
    IDataType target,
    Object value
)
```

**Parameters**

- **target**
  - Type: TwinCAT.TypeSystem.IDataType
  - The datatype information.

- **value**
  - Type: System.Object
  - The value.

**Reference**

- MarshalException Class
- MarshalException Overload
- TwinCAT.TypeSystem Namespace

6.11.106.1.6 MarshalException Constructor (IDataType, Type)

Initializes a new instance of the MarshalException class.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public MarshalException(
    IDataType source,
    Type target
)
```

**Parameters**

- **source**
  - Type: TwinCAT.TypeSystem.IDataType
  - The source.

- **target**
  - Type: System.Type
  - The target.
6.11.106.1.7 MarshalException Constructor (IDataType, Type, String)

Initializes a new instance of the MarshalException class.

**Namespace**: TwinCAT.TypeSystem

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb91b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public MarshalException(
    IDataType source,
    Type target,
    string message
)
```

**Parameters**

- `source` Type: TwinCAT.TypeSystem.IDataType
  The source.
- `target` Type: System.Type
  The target.
- `message` Type: System.String
  The message.

**Reference**

MarshalException Class [2273]

MarshalException Overload [2275]

TwinCAT.TypeSystem Namespace [1622]

6.11.106.1.8 MarshalException Constructor (IInstance, Type, MemberInfo)

Initializes a new instance of the MarshalException class.

**Namespace**: TwinCAT.TypeSystem

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb91b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public MarshalException(
    IInstance instance,
    Type tp,
    MemberInfo member
)
```
Parameters

instance Type: TwinCAT.TypeSystem.IInstance
The instance.

tp Type: System.Type
The type.

member Type: System.Reflection.MemberInfo
The member.

Reference

MarshalException Class [2273]

MarshalException Overload [2275]

TwinCAT.TypeSystem Namespace [1622]

6.11.106.2 MarshalException Properties

The MarshalException [2273] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

MarshalException Class [2273]

TwinCAT.TypeSystem Namespace [1622]

6.11.106.3 MarshalException Methods

The MarshalException [2273] type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetBaseException</code></td>
<td>When overridden in a derived class, returns the <code>Exception</code> that is the root</td>
</tr>
<tr>
<td></td>
<td>cause of one or more subsequent exceptions. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetObjectData</code></td>
<td>When overridden in a derived class, sets the <code>SerializationInfo</code> with</td>
</tr>
<tr>
<td></td>
<td>information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the runtime type of the current instance. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Creates and returns a string representation of the current exception.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

Reference

MarshaledException Class [› 2273]

TwinCAT.TypeSystem Namespace [› 1622]

6.11.107 MemberCollection Class

Collection of `IMember` [› 2065] objects.

Inheritance Hierarchy

```
System.Object
    TwinCAT.TypeSystem.Generic.InstanceCollection [› 2460] `IMember` [› 2065].
    TwinCAT.TypeSystem.MemberCollection
```

Namespace:  TwinCAT.TypeSystem [› 1622]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

```
public class MemberCollection : InstanceCollection<IMember>,
    IMemberCollection, IInstanceCollection<IMember>, IIList<IMember>,
    IList<IMember>, IEnumerable<IMember>, IEnumerable
```

The MemberCollection type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MemberCollection</code></td>
<td>Initializes a new instance of the MemberCollection class.</td>
</tr>
<tr>
<td>[› 2283]</td>
<td></td>
</tr>
<tr>
<td><code>MemberCollection(IEnumerable.IMember)</code></td>
<td>Initializes a new instance of the MemberCollection class (copy constructor)</td>
</tr>
<tr>
<td>[› 2284]</td>
<td></td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the collection count. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>InnerList</td>
<td>Gets the List of instances. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>InnerPathDict</td>
<td>The Path dictionary (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether this instance is read only. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the Instance [2052] at the specified index. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the Instance [2052] with the specified instance path. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Mode</td>
<td>The mode this InstanceCollection.T. [2460] is working in. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
</tbody>
</table>

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds the specified item. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>AddRange</td>
<td>Adds the specified items to this collection. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>Returns a read only copy of this collection (shallow copy)</td>
</tr>
<tr>
<td>Clear</td>
<td>Clears this instance. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Clone</td>
<td>Clones this MemberCollection.</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether this collection contains an Instance [2052] with the specified InstanceName / InstancePath (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>ContainsName</td>
<td>Determines whether the specified instance name contains name. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies this InstanceCollection.T. [2460] to the specified array. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Empty</td>
<td>Returns an Empty Member Collection.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Gets the enumerator. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetInstance</td>
<td>Gets the Instance [2052] by instance path. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>GetInstanceByName</td>
<td>Gets the name of the instance by. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of the specified IInstance [2052]. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts the specified IInstance [2052] at the specified index. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the specified item. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IInstance [2052] at the specified index. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetInstance</td>
<td>Tries to get the IInstance [2052]. of the specified path. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>TryGetInstanceByName</td>
<td>Tries to get Instances by name. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Tries to get the specified member</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]

6.11.107.1 MemberCollection Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemberCollection</td>
<td>Initializes a new instance of the MemberCollection [2281] class.</td>
</tr>
<tr>
<td>MemberCollection(IEnumerable&lt;IMember&gt;)</td>
<td>Initializes a new instance of the MemberCollection [2281] class (copy constructor)</td>
</tr>
</tbody>
</table>

Reference

MemberCollection Class [2281]

TwinCAT.TypeSystem Namespace [1622]

6.11.107.1.1 MemberCollection Constructor

Initializes a new instance of the MemberCollection [2281] class.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
6.11.107.1.2 MemberCollection Constructor (IEnumerable.IMember.)

Initializes a new instance of the MemberCollection class (copy constructor)

**Namespace:** TwinCAT>TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#  
public MemberCollection()

**Parameters**

coll Type: System.Collections.Generic.IEnumerable.IMember.

**Reference**

MemberCollection Class [2281]

MemberCollection Overload [2283]

TwinCAT.TypeSystem Namespace [1622]

### Properties

The MemberCollection type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [2464]</td>
<td>Gets the collection count. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>InnerList [2465]</td>
<td>Gets the List of instances. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>InnerPathDict [2465]</td>
<td>The Path dictionary (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>IsReadOnly [2465]</td>
<td>Gets a value indicating whether this instance is read only. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
</tbody>
</table>
**Name** | **Description**
--- | ---
Item.Int32. ![2466](image) | Gets or sets the `IInstance ![2052](image)` at the specified index. (Inherited from `InstanceCollection.T ![2460](image)`.)
Item.String. ![2467](image) | Gets the `IInstance ![2052](image)` with the specified instance path. (Inherited from `InstanceCollection.T ![2460](image)`.)
Mode ![2468](image) | The mode this `InstanceCollection.T ![2460](image)` is working in. (Inherited from `InstanceCollection.T ![2460](image)`.)

**Reference**

MemberCollection Class ![2281](image)

TwinCAT.TypeSystem Namespace ![1622](image)

### 6.11.107.3 MemberCollection Methods

The `MemberCollection ![2281](image)` type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add <img src="image" alt="2469" /></td>
<td>Adds the specified item. (Inherited from <code>InstanceCollection.T ![2460](image)</code>.)</td>
</tr>
<tr>
<td>AddRange <img src="image" alt="2470" /></td>
<td>Adds the specified items to this collection. (Inherited from <code>InstanceCollection.T ![2460](image)</code>.)</td>
</tr>
<tr>
<td>AsReadOnly <img src="image" alt="2286" /></td>
<td>Returns a read only copy of this collection (shallow copy)</td>
</tr>
<tr>
<td>Clear <img src="image" alt="2471" /></td>
<td>Clears this instance. (Inherited from <code>InstanceCollection.T ![2460](image)</code>.)</td>
</tr>
<tr>
<td>Clone <img src="image" alt="2287" /></td>
<td>Clones this <code>MemberCollection ![2281](image)</code></td>
</tr>
<tr>
<td>Contains(String) <img src="image" alt="2472" /></td>
<td>Determines whether this collection contains an <code>IInstance ![2052](image)</code> with the specified <code>InstanceName / InstancePath</code> (Inherited from <code>InstanceCollection.T ![2460](image)</code>.)</td>
</tr>
<tr>
<td>Contains(T) <img src="image" alt="2472" /></td>
<td>Determines whether this collection contains the specified <code>IInstance ![2052](image)</code> (Inherited from <code>InstanceCollection.T ![2460](image)</code>.)</td>
</tr>
<tr>
<td>ContainsName <img src="image" alt="2473" /></td>
<td>Determines whether the specified instance name contains name. (Inherited from <code>InstanceCollection.T ![2460](image)</code>.)</td>
</tr>
<tr>
<td>CopyTo <img src="image" alt="2474" /></td>
<td>Copies this <code>InstanceCollection.T ![2460](image)</code> to the specified array. (Inherited from <code>InstanceCollection.T ![2460](image)</code>.)</td>
</tr>
<tr>
<td>Empty <img src="image" alt="2287" /></td>
<td>Returns an Empty Member Collection.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object ![image]</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object ![image]</code>.)</td>
</tr>
<tr>
<td>GetEnumerator <img src="image" alt="2474" /></td>
<td>Gets the enumerator. (Inherited from <code>InstanceCollection.T ![2460](image)</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object ![image]</code>.)</td>
</tr>
<tr>
<td>GetInstance <img src="image" alt="2475" /></td>
<td>Gets the <code>IInstance ![2052](image)</code> by instance path. (Inherited from <code>InstanceCollection.T ![2460](image)</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetInstanceByName</td>
<td>Gets the name of the instance by. (Inherited from InstanceCollection.T.</td>
</tr>
<tr>
<td></td>
<td>[2476]</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf [2476]</td>
<td>Determines the index of the specified Instance [2052]. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Insert [2477]</td>
<td>Inserts the specified Instance [2052] at the specified index. (Inherited</td>
</tr>
<tr>
<td></td>
<td>from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove [2478]</td>
<td>Removes the specified item. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>RemoveAt [2479]</td>
<td>Removes the Instance [2052] at the specified index. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object)</td>
</tr>
<tr>
<td>TryGetInstance [2479]</td>
<td>Tries to get the Instance [2052] of the specified path. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>TryGetInstanceByName</td>
<td>Tries to get Instances by name. (Inherited from InstanceCollection.T.</td>
</tr>
<tr>
<td></td>
<td>[2460].)</td>
</tr>
<tr>
<td>TryGetMember [2287]</td>
<td>Tries to get the specified member</td>
</tr>
</tbody>
</table>

**Reference**

MemberCollection Class [2281]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.107.3.1 MemberCollection.AsReadOnly Method

Returns a read only copy of this collection (shallow copy)

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:

5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public ReadOnlyMemberCollection AsReadOnly()
```

**Return Value**

Type: ReadOnlyMemberCollection [2322]

The readonly copy.

**Reference**

MemberCollection Class [2281]

TwinCAT.TypeSystem Namespace [1622]
6.11.107.3.2 MemberCollection.Clone Method

Clones this MemberCollection.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public MemberCollection Clone()
```

**Return Value**

Type: MemberCollection

A cloned MemberCollection.

**Reference**

- MemberCollection Class
- TwinCAT.TypeSystem Namespace

6.11.107.3.3 MemberCollection.Empty Method

Returns an Empty Member Collection.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public static MemberCollection Empty()
```

**Return Value**

Type: MemberCollection

MemberCollection.

**Reference**

- MemberCollection Class
- TwinCAT.TypeSystem Namespace

6.11.107.3.4 MemberCollection.TryGetMember Method

Tries to get the specified member

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

```csharp
public bool TryGetMember(
    string fieldName,
    out IMember symbol
)
```

Parameters

- `fieldName`  
  Type: `System.String`  
  Name of the member.

- `symbol`  
  Type: `TwinCAT.TypeSystem.IMember`  
  The symbol.

Return Value

Type: `Boolean`  
true if found, false otherwise.

Implements

`IMemberCollection.TryGetMember(String, IMember.)`  

Reference

MemberCollection Class  
[TwinCAT.TypeSystem.IMember](#)

TwinCAT.TypeSystem Namespace  
[TwinCAT.TypeSystem](#)

6.11.108   MethodParamFlags Enumeration

Flag set specifying the MethodParameter context

Namespace: `TwinCAT.TypeSystem`  
Assembly: `TwinCAT.Ads.Abstractions`  
Version: `5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`

Syntax

C#

```csharp
[FlagsAttribute]
public enum MethodParamFlags
```

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>In</td>
<td>1</td>
<td>Input Parameter (ADSMETHODPARAFLAG_IN)</td>
</tr>
<tr>
<td>Out</td>
<td>2</td>
<td>Output Parameter (ADSMETHODPARAFLAG_OUT)</td>
</tr>
<tr>
<td>ByReference</td>
<td>4</td>
<td>By reference Parameter (ADSMETHODPARAFLAG_BYREFERENCE)</td>
</tr>
</tbody>
</table>
## 6.11.109 PrimitiveTypeFlags Enumeration

Enum PrimitiveTypeFlags

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

C#  

```csharp
[FlagsAttribute]
public enum PrimitiveTypeFlags
```

### Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>None / Uninitialized</td>
</tr>
<tr>
<td>System</td>
<td>1</td>
<td>System Type like Byte / Word / DWORD</td>
</tr>
<tr>
<td>Unsigned</td>
<td>2</td>
<td>Primitive Type is Unsigned</td>
</tr>
<tr>
<td>Bool</td>
<td>4</td>
<td>Boolean Value (maps to true and false)</td>
</tr>
<tr>
<td>Float</td>
<td>8</td>
<td>Floating Point</td>
</tr>
<tr>
<td>Date</td>
<td>16</td>
<td>Type represents a Date</td>
</tr>
<tr>
<td>Time</td>
<td>32</td>
<td>Type represents a Time</td>
</tr>
<tr>
<td>Numeric</td>
<td>64</td>
<td>Numeric value</td>
</tr>
<tr>
<td>Bitset</td>
<td>128</td>
<td>Bitset</td>
</tr>
<tr>
<td>MaskNumericUnsigned</td>
<td>66</td>
<td>Numeric / Unsigned Mask</td>
</tr>
<tr>
<td>MaskDateTime</td>
<td>48</td>
<td>Date / Time Mask</td>
</tr>
<tr>
<td>MaskSpecialType</td>
<td>67</td>
<td>Special type Mask</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]

## 6.11.110 RawValueChangedEventArgs Class

Event args for the RawValueChanged [2254] event.

### Inheritance Hierarchy

```
System.Object
    System.EventArgs
        TwinCAT.TypeSystem.ValueChangedBaseEventArgs [2435]
        TwinCAT.TypeSystem.RawValueChangedEventArgs
```

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
public class RawValueChangedEventArgs : ValueChangedBaseEventArgs
```

The RawValueChangedEventArgs type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DateTime</code></td>
<td>Notification timestamp (Inherited from <code>ValueChangedBaseEventArgs</code>).</td>
</tr>
<tr>
<td><code>Symbol</code></td>
<td>Gets the symbol. (Inherited from <code>ValueChangedBaseEventArgs</code>.)</td>
</tr>
<tr>
<td><code>Value</code></td>
<td>New Value (byte[])</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]

6.11.110.1 RawValueChangedEventArgs Properties

The RawValueChangedEventArgs [2289] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DateTime</code></td>
<td>Notification timestamp (Inherited from <code>ValueChangedBaseEventArgs</code>.)</td>
</tr>
<tr>
<td><code>Symbol</code></td>
<td>Gets the symbol. (Inherited from <code>ValueChangedBaseEventArgs</code>.)</td>
</tr>
<tr>
<td><code>Value</code></td>
<td>New Value (byte[])</td>
</tr>
</tbody>
</table>

Reference

RawValueChangedEventArgs Class [2289]

TwinCAT.TypeSystem Namespace [1622]
6.11.110.1 RawValueChangedEventArgs.Value Property

New Value (byte[])

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public ReadOnlyMemory Value { get; }

Property Value

Type: ReadOnlyMemory

Reference

RawValueChangedEventArgs Class [2289]
TwinCAT.TypeSystem Namespace [1622]

6.11.110.2 RawValueChangedEventArgs Methods

The RawValueChangedEventArgs [2289] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

RawValueChangedEventArgs Class [2289]
TwinCAT.TypeSystem Namespace [1622]

6.11.111 ReadOnlyDataTypeCollection Class

ReadOnly Collection of IDataType [1986] objects.
Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection\<IDataType\>[\>1986].
  TwinCAT.TypeSystem.Generic.ReadOnlyDataTypeCollection[\>2499],\<IDataType\>[\>1986].
  TwinCAT.TypeSystemReadOnlyDataTypeCollection

Namespace: TwinCAT.TypeSystem[\>1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class ReadOnlyDataTypeCollection : ReadOnlyDataTypeCollection<\<IDataType\>>,
  IDataTypeCollection, IDataTypeCollection<\<IDataType\>>, ICollection<\<IDataType\>>,
  IEnumerable<\<IDataType\>>, IEnumerable

The ReadOnlyDataTypeCollection type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ReadOnlyDataTypeCollection</strong>[&gt;2293]</td>
<td>Initializes a new instance of the ReadOnlyDataTypeCollection class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection&lt;&lt;IDataType&gt;[&gt;1986].)</td>
</tr>
<tr>
<td>Item.Int32.</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection&lt;&lt;IDataType&gt;[&gt;1986].)</td>
</tr>
<tr>
<td>Item.String.[&gt;2502]</td>
<td>Gets the element with the specified type name. (Inherited from ReadOnlyDataTypeCollection&lt;T&gt;[&gt;2499].)</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the IList&lt;T&gt; that the ReadOnlyCollection&lt;T&gt; wraps. (Inherited from ReadOnlyCollection&lt;&lt;IDataType&gt;[&gt;1986].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>Determines whether an element is in the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;&lt;IDataType&gt;[&gt;1986].)</td>
</tr>
<tr>
<td>ContainsType</td>
<td>Determines whether the specified name contains type. (Inherited from ReadOnlyDataTypeCollection&lt;T&gt;[&gt;2499].)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire ReadOnlyCollection&lt;T&gt; to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection&lt;&lt;IDataType&gt;[&gt;1986].)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;&lt;IDataType&gt;[&gt;1986].)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;IDataType&gt; [1986].)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetType</td>
<td>Tries to get the Type with the specified name out of the collection. (Inherited from ReadOnlyDataTypeCollection&lt;T&gt; [2499].)</td>
</tr>
</tbody>
</table>

### TwinCAT.TypeSystem Namespace [1622]

## 6.11.111.1 ReadOnlyDataTypeCollection Constructor

Initializes a new instance of the ReadOnlyDataTypeCollection [2291] class.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0eaa15da1c14

### Syntax

**C#**

```csharp
public ReadOnlyDataTypeCollection(
    DataTypeCollection<IDataType> coll
)
```

**Parameters**

- **coll**
  
  Type: TwinCAT.TypeSystem.Generic.DataTypeCollection[IDataType [1986].
  Collection of types.

### Reference

ReadOnlyDataTypeCollection Class [2291]
TwinCAT.TypeSystem Namespace [1622]

## 6.11.111.2 ReadOnlyDataTypeCollection Properties

The ReadOnlyDataTypeCollection [2291] type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ReadOnlyCollection&lt;T&gt; instance. (Inherited from ReadOnlyCollection&lt;IDataType&gt; [1986].)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection.IDataType [1986].)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the element with the specified type name. (Inherited from ReadOnlyDataTypeCollection.T. [2499].)</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the IList.T that the ReadOnlyCollection.T wraps. (Inherited from ReadOnlyCollection.IDataType [1986].)</td>
</tr>
</tbody>
</table>

### Reference

ReadOnlyDataTypeCollection Class [2291]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.111.3 ReadOnlyDataTypeCollection Methods

The ReadOnlyDataTypeCollection [2291] type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>Determines whether an element is in the ReadOnlyCollection.T. (Inherited from ReadOnlyCollection.IDataType [1986].)</td>
</tr>
<tr>
<td>ContainsType</td>
<td>Determines whether the specified name contains type. (Inherited from ReadOnlyDataTypeCollection.T. [2499].)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IDataType [1986].)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection.T. (Inherited from ReadOnlyCollection.IDataType [1986].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T. (Inherited from ReadOnlyCollection.IDataType [1986].)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetType</td>
<td>Tries to get the Type with the specified name out of the collection. (Inherited from ReadOnlyDataTypeCollection.T. [2499].)</td>
</tr>
</tbody>
</table>

### Reference

ReadOnlyDataTypeCollection Class [2291]

TwinCAT.TypeSystem Namespace [1622]
6.11.12 **ReadOnlyDimensionCollection Class**

ReadOnly version of the DimensionCollection

**Inheritance Hierarchy**

- System.Object
  - System.Collections.ObjectModel.ReadOnlyCollection<br>  - IDimension
- TwinCAT.TypeSystem.ReadOnlyDimensionCollection

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public class ReadOnlyDimensionCollection : ReadOnlyCollection<IDimension>,
IDimensionCollection, IList<IDimension>, ICollection<IDimension>,
IEnumerable<IDimension>, IEnumerable
```

The `ReadOnlyDimensionCollection` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>ReadOnlyCollection</code>. Instance. (Inherited from <code>ReadOnlyCollection</code>. <code>IDimension</code>)</td>
</tr>
<tr>
<td>ElementCount</td>
<td>Gets the Number of elements in all Dimensions</td>
</tr>
<tr>
<td>Item</td>
<td>Gets the element at the specified index. (Inherited from <code>ReadOnlyCollection</code>. <code>IDimension</code>)</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the <code>IList</code> that the <code>ReadOnlyCollection</code> wraps. (Inherited from <code>ReadOnlyCollection</code>. <code>IDimension</code>)</td>
</tr>
<tr>
<td>LowerBounds</td>
<td>Gets the lower bounds.</td>
</tr>
<tr>
<td>UpperBounds</td>
<td>Gets the upper bounds.</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>Determines whether an element is in the <code>ReadOnlyCollection</code>. (Inherited from <code>ReadOnlyCollection</code>. <code>IDimension</code>)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire <code>ReadOnlyCollection</code> to a compatible one-dimensional <code>Array</code>, starting at the specified index of the target array. (Inherited from <code>ReadOnlyCollection</code>. <code>IDimension</code>)</td>
</tr>
<tr>
<td>Empty</td>
<td>Returns an empty <code>ReadOnlyDimensionCollection</code></td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDimensionLengths</td>
<td>Gets an array that specifies the lengths of each array dimension.</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;IDimension&gt;[1998]..)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection[IDimension][1998]..)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.112.1 ReadOnlyDimensionCollection Properties

The ReadOnlyDimensionCollection [2295] type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ReadOnlyCollection&lt;T&gt; instance. (Inherited from ReadOnlyCollection,IDimension[1998]..)</td>
</tr>
<tr>
<td>ElementCount</td>
<td>Gets the Number of elements in all Dimensions</td>
</tr>
<tr>
<td>Item</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection,IDimension[1998]..)</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the IList&lt;T&gt; that the ReadOnlyCollection&lt;T&gt; wraps. (Inherited from ReadOnlyCollection,IDimension[1998]..)</td>
</tr>
<tr>
<td>LowerBounds</td>
<td>Gets the lower bounds.</td>
</tr>
<tr>
<td>UpperBounds</td>
<td>Gets the upper bounds.</td>
</tr>
</tbody>
</table>

### Reference

ReadOnlyDimensionCollection Class [2295]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.112.1.1 ReadOnlyDimensionCollection.ElementCount Property

Gets the Number of elements in all Dimensions

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdcdca3e72bc0ea15da1c14
Syntax

C#
public int ElementCount { get; }

Property Value

Type: Int32

Implements

IDimensionCollection.ElementCount [› 2002]

Reference

ReadOnlyDimensionCollection Class [› 2295]
TwinCAT.TypeSystem Namespace [› 1622]

6.11.112.1.2 ReadOnlyDimensionCollection.LowerBounds Property

Gets the lower bounds.

Namespace: TwinCAT.TypeSystem [› 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public int[] LowerBounds { get; }

Property Value

Type: Int32.
The lower bounds.

Implements

IDimensionCollection.LowerBounds [› 2002]

Reference

ReadOnlyDimensionCollection Class [› 2295]
TwinCAT.TypeSystem Namespace [› 1622]

6.11.112.1.3 ReadOnlyDimensionCollection.UpperBounds Property

Gets the upper bounds.

Namespace: TwinCAT.TypeSystem [› 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#
public int[] UpperBounds { get; }

Property Value
Type: .Int32.
The upper bounds.

Implements
IDimensionCollection.UpperBounds [2002]

Reference
ReadOnlyDimensionCollection Class [2295]
TwinCAT.TypeSystem Namespace [1622]

6.11.112.2 ReadOnlyDimensionCollection Methods

The ReadOnlyDimensionCollection [2295] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>Determines whether an element is in the ReadOnlyCollection&lt;T.. (Inherited from ReadOnlyCollection.IDimension[1998].)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire ReadOnlyCollection&lt;T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IDimension[1998].)</td>
</tr>
<tr>
<td>Empty</td>
<td>Returns an empty ReadOnlyDimensionCollection [2295]</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetDimensionLengths</td>
<td>Gets an array the specifies the Lengths of each Array Dimension [2299]</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection&lt;T.. (Inherited from ReadOnlyCollection.IDimension[1998].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection&lt;T.. (Inherited from ReadOnlyCollection.IDimension[1998].)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### ToString

**Description**
Returns a string that represents the current object. (Inherited from `Object`.)

### 6.11.112.2.1 `ReadOnlyDimensionCollection.Empty` Method

**Description**
Returns an empty `ReadOnlyDimensionCollection`.

**Namespace**: `TwinCAT.TypeSystem`  
**Assembly**: `TwinCAT.Ads.Abstractions` (in `TwinCAT.Ads.Abstractions.dll`)  
**Version**: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**C#**
```
public static ReadOnlyDimensionCollection Empty()
```

**Return Value**
Type: `ReadOnlyDimensionCollection`  
`ReadOnlyDimensionCollection`.

**Reference**
- `ReadOnlyDimensionCollection Class`  
- `TwinCAT.TypeSystem Namespace`  

### 6.11.112.2.2 `ReadOnlyDimensionCollection.GetDimensionLengths` Method

**Description**
Gets an array the specifies the Lengths of each Array Dimension

**Namespace**: `TwinCAT.TypeSystem`  
**Assembly**: `TwinCAT.Ads.Abstractions` (in `TwinCAT.Ads.Abstractions.dll`)  
**Version**: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**C#**
```
public int[] GetDimensionLengths()
```

**Return Value**
Type: `Int32`.  
System.Int32[].

**Implements**
- `IDimensionCollection.GetDimensionLengths`
### TwinCAT.Ads Namespaces

**Reference**
- `ReadOnlyDimensionCollection Class` [*2295*]
- `TwinCAT.TypeSystem Namespace` [*1622*]

### 6.11.13 `ReadOnlyEnumValueCollection Class`

Read only version of the `EnumValueCollection.T` [*1926*]

**Inheritance Hierarchy**

```
System.Object
   System.Collections.ObjectModel.ReadOnlyCollection<IEnumValue>
      TwinCAT.TypeSystem.ReadOnlyEnumValueCollection
```

**Namespace:** `TwinCAT.TypeSystem` [*1622*]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public class ReadOnlyEnumValueCollection : ReadOnlyCollection<IEnumValue>, IEnumValueCollection, IEnumValueCollection<IEnumValue, IConvertible>, ICollection<IEnumValue>, IEnumerable<IEnumValue>, IEnumerable
```

The `ReadOnlyEnumValueCollection` type exposes the following members.

**Constructors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReadOnlyEnumValueCollection</code> [<em>2301</em>]</td>
<td>Initializes a new instance of the <code>ReadOnlyEnumValueCollection.T</code> [<em>2309</em>] class.</td>
</tr>
</tbody>
</table>

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Count</code></td>
<td>Gets the number of elements contained in the <code>ReadOnlyCollection.T</code> instance. (Inherited from <code>ReadOnlyCollection.IEnumValue</code> [<em>2028</em>].)</td>
</tr>
<tr>
<td><code>Item.Int32</code></td>
<td>Gets the element at the specified index. (Inherited from <code>ReadOnlyCollection.IEnumValue</code> [<em>2028</em>].)</td>
</tr>
<tr>
<td><code>Item.String</code> [<em>2302</em>]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td><code>Items</code></td>
<td>Returns the <code>IList.T</code> that the <code>ReadOnlyCollection.T</code> wraps. (Inherited from <code>ReadOnlyCollection.IEnumValue</code> [<em>2028</em>].)</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Contains(String)</code> [<em>2305</em>]</td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
<tr>
<td><code>Contains(T)</code></td>
<td>Determines whether an element is in the <code>ReadOnlyCollection.T</code>. (Inherited from <code>ReadOnlyCollection.IEnumValue</code> [<em>2028</em>].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire <code>ReadOnlyCollection&lt;T&gt;</code> to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from <code>ReadOnlyCollection&lt;TEnumValue&gt;</code>)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the <code>ReadOnlyCollection&lt;T&gt;</code>. (Inherited from <code>ReadOnlyCollection&lt;TEnumValue&gt;</code>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>GetNames</td>
<td>Gets the Value Names.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>GetValues</td>
<td>Gets the values.</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire <code>ReadOnlyCollection&lt;T&gt;</code>. (Inherited from <code>ReadOnlyCollection&lt;TEnumValue&gt;</code>)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>Parse</td>
<td>Parses the specified name.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>TryParse</td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
<tr>
<td>TryParse</td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

### 6.11.113.1 `ReadOnlyEnumValueCollection` Constructor

Initializes a new instance of the `ReadOnlyEnumValueCollection<T>` class.

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-S.0.9bb9a1b43b6095934ffdc3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public ReadOnlyEnumValueCollection(
    EnumValueCollection coll
)```

TC1000  
Version: 1.1  
2301
Parameters

coll

Type: TwinCAT.TypeSystem.EnumValueCollection

The coll.

Reference

ReadOnlyEnumValueCollection Class [2300]

TwinCAT.TypeSystem Namespace [1622]

6.11.113.2  ReadOnlyEnumValueCollection Properties

The ReadOnlyEnumValueCollection [2300] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ReadOnlyCollection&lt;T&gt;. instance. (Inherited from ReadOnlyCollection.IEnumValue [2028]..)</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection.IEnumValue [2028]..)</td>
</tr>
<tr>
<td>Item.String [2302]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the IList&lt;T&gt; that the ReadOnlyCollection&lt;T&gt; wraps. (Inherited from ReadOnlyCollection.IEnumValue [2028]..)</td>
</tr>
</tbody>
</table>

Reference

ReadOnlyEnumValueCollection Class [2300]

TwinCAT.TypeSystem Namespace [1622]

6.11.113.2.1  ReadOnlyEnumValueCollection.Item Property

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection.IEnumValue [2028]..)</td>
</tr>
<tr>
<td>Item.String [2302]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
</tbody>
</table>

Reference

ReadOnlyEnumValueCollection Class [2300]

TwinCAT.TypeSystem Namespace [1622]

ReadOnlyEnumValueCollection.Item Property (String)

Gets or sets the element at the specified index.
Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public IConvertible this[string name] { get; }
```

Parameters

- name: Type: System.String
  The name of the value

Return Value

Type: IConvertible
EnumerableValue<T>.

Implements

IEnumerableValueCollection.TEnumerableValue, TValue..Item.String

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

ReadOnlyEnumerableValueCollection Class

Item Overload

TwinCAT.TypeSystem Namespace

6.11.113.3 ReadOnlyEnumerableValueCollection Methods

The ReadOnlyEnumerableValueCollection type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String)</td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;TEnumValue, TValue, Item.String)</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyTo</td>
<td>Copies the entire ReadOnlyCollection&lt;T&gt; to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection&lt;IEnumValue&gt;[..])</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection.IEnumValue[..])</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetNames</td>
<td>Gets the Value Names.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetValues</td>
<td>Gets the values.</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection.IEnumValue[..])</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Parse</td>
<td>Parses the specified name.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryParse(String, IConvertible)</td>
<td>Tries to pars the string value of the Enum.</td>
</tr>
<tr>
<td>TryParse(String, IEnumValue)</td>
<td>Tries to pars the string value of the Enum.</td>
</tr>
</tbody>
</table>

**Reference**

ReadOnlyEnumValueCollection Class [2300]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.113.1 ReadOnlyEnumValueCollection.Contains Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String) [2305]</td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection.IEnumValue[..])</td>
</tr>
</tbody>
</table>

**Reference**

ReadOnlyEnumValueCollection Class [2300]
TwinCAT.TypeSystem Namespace [1622]
ReadOnlyEnumValueCollection.Contains Method (String)

Determines whether [contains] [the specified name].

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool Contains(
    string value
)
```

**Parameters**

- **value**
  - Type: System.String
  - Value

**Return Value**

- Type: Boolean
  - true if [contains] [the specified name]; otherwise, false.

**Implements**

IEnumValueCollection.TEnumValue, TValue..Contains(String)

**Reference**

ReadOnlyEnumValueCollection Class
Contains Overload
TwinCAT.TypeSystem Namespace

6.11.113.3.2 ReadOnlyEnumValueCollection.GetNames Method

Gets the Value Names.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public string[] GetNames()
```

**Return Value**

- Type: System.String[]

- System.String[].
### Implements

IEnumValueCollection.TEnumValue, TValue..GetNames. [口 2037]

### Reference

ReadOnlyEnumValueCollection Class [口 2300]

TwinCAT.TypeSystem Namespace [口 1622]

### 6.11.113.3.3 ReadOnlyEnumValueCollection.GetValues Method

Gets the values.

**Namespace:** TwinCAT.TypeSystem [口 1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public IConvertible[] GetValues()
```

**Return Value**

Type: IConvertible

T[].

### Implements

IEnumValueCollection.TEnumValue, TValue..GetValues. [口 2037]

### Reference

ReadOnlyEnumValueCollection Class [口 2300]

TwinCAT.TypeSystem Namespace [口 1622]

### 6.11.113.3.4 ReadOnlyEnumValueCollection.Parse Method

 Parses the specified name.

**Namespace:** TwinCAT.TypeSystem [口 1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public IConvertible Parse(
    string name
)
```
Parameters

name 
  Type: System.String
  The name.

Return Value

Type: IConvertible
T.

Implements

IEnumeratorValueCollection.TEnumValueCollection, TValue.Parse(String) [2038]

Reference

ReadOnlyEnumValueCollection Class [2300]
TwinCAT.TypeSystem Namespace [1622]

6.11.113.3.5 ReadOnlyEnumValueCollection.TryParse Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryParse(String, IConvertible.) [2307]</td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
<tr>
<td>TryParse(String, IEnumValueCollection) [2308]</td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
</tbody>
</table>

Reference

ReadOnlyEnumValueCollection Class [2300]
TwinCAT.TypeSystem Namespace [1622]

ReadOnlyEnumValueCollection.TryParse Method (String, IConvertible.)

Tries to parse the string value of the Enum.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool TryParse(
    string strValue,
    out IConvertible value
)
```
Parameters
strValue Type: System.String
The Value in string representation.

value Type: System.IConvertible.
The value.

Return Value
Type: Boolean
ture if XXXX, false otherwise.

Implements
IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue.) [2039]

Reference
ReadOnlyEnumValueCollection Class [2300]

TryParse Overload [2307]

TwinCAT.TypeSystem Namespace [1622]

ReadOnlyEnumValueCollection.TryParse Method (String, IEnumValue.)
Tries to parse the string value of the Enum.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
public bool TryParse(
    string strValue,
    out TwinCAT.TypeSystem.IEnumValue value
)

Parameters
strValue Type: System.String
The Value in string representation.

value Type: TwinCAT.TypeSystem.IEnumValue [2028].
The value.

Return Value
Type: Boolean
ture if XXXX, false otherwise.

Implements
IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue.) [2039]
Reference
ReadOnlyEnumValueCollection Class [2300]
TryParse Overload [2307]
TwinCAT.TypeSystem Namespace [1622]

6.11.14  ReadOnlyEnumValueCollection<T> Class
Read only version of the EnumValueCollection<T> [1926]

Inheritance Hierarchy
System.Object
        ReadOnlyEnumValueCollection<T>

Namespace:  TwinCAT.TypeSystem [1622]
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#

    public class ReadOnlyEnumValueCollection<T> : ReadOnlyCollection<EnumValue<T>, T>, IEnumValueCollection<EnumValue<T>, T>, ICollection<EnumValue<T>>, IEnumerable<EnumValue<T>>, IEnumerable
where T : IConvertible

Type Parameters
T
The ReadOnlyEnumValueCollection<T> type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ]</td>
<td>Initializes a new instance of the ReadOnlyEnumValueCollection&lt;T&gt; class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ]</td>
<td>Gets the number of elements contained in the ReadOnlyCollection&lt;T&gt; instance. (Inherited from ReadOnlyCollection&lt;EnumValue&lt;T&gt;, T&gt;...</td>
</tr>
<tr>
<td>![ ]</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection&lt;EnumValue&lt;T&gt;, T&gt;...</td>
</tr>
<tr>
<td>![ ]</td>
<td>Gets the enumeration value T from its string representation.</td>
</tr>
<tr>
<td>![ ]</td>
<td>Returns the IList&lt;T&gt; that the ReadOnlyCollection&lt;T&gt; wraps. (Inherited from ReadOnlyCollection&lt;EnumValue&lt;T&gt;, T&gt;...</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String)</td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the ReadOnlyCollection.T. (Inherited from ReadOnlyCollection,EnumValue)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire ReadOnlyCollection.T, to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection,EnumValue)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection.T. (Inherited from ReadOnlyCollection,EnumValue)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object)</td>
</tr>
<tr>
<td>GetNames</td>
<td>Gets the Value Names.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object)</td>
</tr>
<tr>
<td>GetValues</td>
<td>Gets the values.</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T. (Inherited from ReadOnlyCollection,EnumValue)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object)</td>
</tr>
<tr>
<td>Parse</td>
<td>Parses the specified name.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object)</td>
</tr>
<tr>
<td>TryParse(String, EnumValue,T..)</td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
<tr>
<td>TryParse(String, T.)</td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
</tbody>
</table>

## Reference

TwinCAT.TypeSystem Namespace [1622]

### 6.11.114.1 ReadOnlyEnumValueCollection.T. Constructor

Initializes a new instance of the ReadOnlyEnumValueCollection.T class.

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public ReadOnlyEnumValueCollection(
    EnumValueCollection<T> coll)
```

Parameters

coll Type: 

```csharp
TwinCAT.TypeSystem.EnumValueCollection [1926] T [2309].
```

The coll.

Reference

ReadOnlyEnumValueCollection.T. Class [2309]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.114.2 ReadOnlyEnumValueCollection.T. Properties

The `ReadOnlyEnumValueCollection.T` generic type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>ReadOnlyCollection&lt;T&gt;</code> instance. (Inherited from <code>ReadOnlyCollection.EnumValueCollection</code>...)</td>
</tr>
<tr>
<td>Item.Int32.</td>
<td>Gets the element at the specified index. (Inherited from <code>ReadOnlyCollection.EnumValueCollection</code>...)</td>
</tr>
<tr>
<td>Item.String. [2312]</td>
<td>Gets the enumeration value <code>T</code> from its string representation.</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the <code>IList&lt;T&gt;</code> that the <code>ReadOnlyCollection&lt;T&gt;</code> wraps. (Inherited from <code>ReadOnlyCollection.EnumValueCollection</code>...)</td>
</tr>
</tbody>
</table>

Reference

ReadOnlyEnumValueCollection.T. Class [2309]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.114.2.1 ReadOnlyEnumValueCollection.T..Item Property

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32.</td>
<td>Gets the element at the specified index. (Inherited from <code>ReadOnlyCollection.EnumValueCollection</code>...)</td>
</tr>
<tr>
<td>Item.String. [2312]</td>
<td>Gets the enumeration value <code>T</code> from its string representation.</td>
</tr>
</tbody>
</table>

Reference

ReadOnlyEnumValueCollection.T. Class [2309]

TwinCAT.TypeSystem Namespace [1622]
ReadOnlyEnumValueCollection.T..Item Property (String)

Gets the enumeration value T from its string representation.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public T this[string name] { get; }
```

**Parameters**

- **name**
  - Type: System.String
  - The name of the enum value.

**Return Value**

- Type: T[2309]
  - T.

**Implements**

- IEnumValueCollection.TEnumValue, TValue..Item.String. [2035]

**Reference**

- ReadOnlyEnumValueCollection.T. Class [2309]
- Item Overload [2311]
- TwinCAT.TypeSystem Namespace [1622]

### 6.11.114.3 ReadOnlyEnumValueCollection.T. Methods

The **ReadOnlyEnumValueCollection.T. [2309]** generic type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String) [2313]</td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.EnumValue [1901], T[2309]... )</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.EnumValue [1901], T[2309]... )</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object. )</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object. )</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the <code>ReadOnlyCollection&lt;T&gt;</code> . (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code> . <code>EnumValue { 1901, T } { 2309 }...</code>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>GetNames</td>
<td>Gets the Value Names.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>GetValues</td>
<td>Gets the values.</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire <code>ReadOnlyCollection&lt;T&gt;</code> . (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code> . <code>EnumValue { 1901, T } { 2309 }...</code>)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>Parse</td>
<td>Parses the specified name.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>TryParse(String, EnumValue&lt;T&gt;)</td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
<tr>
<td>TryParse(String, T)</td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
</tbody>
</table>

**Reference**

`ReadOnlyEnumValueCollection<T> Class { 2309 } |

TwinCAT.TypeSystem Namespace { 1622 }

**6.11.143.1 `ReadOnlyEnumValueCollection<T>..Contains Method**

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String) { 2313 }</td>
<td>Determines whether [contains] [the specified name].</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the <code>ReadOnlyCollection&lt;T&gt;</code> . (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code> . <code>EnumValue { 1901, T } { 2309 }...</code>)</td>
</tr>
</tbody>
</table>

**Reference**

`ReadOnlyEnumValueCollection<T> Class { 2309 } |

TwinCAT.TypeSystem Namespace { 1622 }

**ReadOnlyEnumValueCollection<T>..Contains Method (String)**

Determines whether [contains] [the specified name].

**Namespace:** TwinCAT.TypeSystem { 1622 }

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public bool Contains(
    string value
)
```

**Parameters**

- `value`  
  Type: `System.String`  
  Value

**Return Value**

Type: `Boolean`  
true if [contains] [the specified name]; otherwise, false.

**Implements**

`IEnumValueCollection.TEnumValue, TValue..Contains(String)` [2313]

**Reference**

- `ReadOnlyEnumValueCollection.T. Class [2309]`
- `Contains Overload [2313]`
- `TwinCAT.TypeSystem Namespace [1622]`

### 6.11.143.2 `ReadOnlyEnumValueCollection.T..GetNames Method`

Gets the Value Names.

**Namespace:** `TwinCAT.TypeSystem [1622]`
**Assembly:** `TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`

**Syntax**

**C#**

```csharp
public string[] GetNames()
```

**Return Value**

Type: `System.String[]`.  
Implements  
`IEnumValueCollection.TEnumValue, TValue..GetNames. [2037]`

**Reference**

- `ReadOnlyEnumValueCollection.T. Class [2309]`
- `TwinCAT.TypeSystem Namespace [1622]`
6.11.114.3.3  **ReadonlyEnumValueCollection.T..GetValues Method**

Gets the values.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdc3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public T[] GetValues()
```

**Return Value**

Type: T[].

**Implements**

IEnumValueCollection.TEnumValue, TValue..GetValues.

**Reference**

ReadonlyEnumValueCollection.T. Class

TwinCAT.TypeSystem Namespace

6.11.114.3.4  **ReadonlyEnumValueCollection.T..Parse Method**

Parses the specified name.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffdc3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public T Parse(
    string name
)
```

**Parameters**

- **name**
  - Type: System.String
  - The name.

**Return Value**

Type: T[].

**Implements**

IEnumValueCollection.TEnumValue, TValue.Parse(String)
6.11.114.3.5  ReadOnlyEnumValueCollection.T..TryParse Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryParse(String, EnumValue&lt;T..)</td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
<tr>
<td>TryParse(String, T.)</td>
<td>Tries to parse the string value of the Enum.</td>
</tr>
</tbody>
</table>

**Reference**

ReadOnlyEnumValueCollection.T..Class [2309]

TwinCAT.TypeSystem Namespace [1622]

**ReadOnlyEnumValueCollection.T..TryParse Method (String, EnumValue<T..)**

Tries to parse the string value of the Enum.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool TryParse(
    string strValue,
    out EnumValue<T> value
)
```

**Parameters**

- `strValue`  
  Type: `System.String`  
  The Value in string representation.

- `value`  
  Type: `TwinCAT.TypeSystem.EnumValue<T..>`  
  The value.

**Return Value**

Type: `Boolean`  
true if XXXX, false otherwise.

**Implements**

`IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue..)` [2039]
ReadonlyEnumValueCollection.T..TryParse Method (String, T.)

Tries to parse the string value of the Enum.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  
```csharp
public bool TryParse(
    string strValue,
    out T value
)
```

Parameters

- **strValue**  
  Type: System.String
  The Value in string representation.

- **value**  
  Type: T [2309],
  The value.

Return Value

Type: Boolean  
true if XXXX, false otherwise.

Implements

- IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue.) [2039]

Reference

ReadonlyEnumValueCollection.T . Class [2309]  
TryParse Overload [2316]  
TwinCAT.TypeSystem Namespace [1622]

6.11.115 ReadonlyFieldCollection Class

Read only collection of IField [2040] objects

Inheritance Hierarchy

System.Object  
  TwinCAT.TypeSystem.Generic.ReadOnlyInstanceCollection [2505].IField [2040].
TwinCAT.Ads Namespaces

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class ReadOnlyFieldCollection : ReadOnlyInstanceCollection<IField>,
    IFIELDCollection, IInstanceCollection<IField>, IList<IField>,
    ICollection<IField>, IEnumerable<IField>, IEnumerable
```

The ReadOnlyFieldCollection type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| GetEnumerator         | Returns an enumerator that iterates through the ReadOnlyCollection<T>. (Inherited from ReadOnlyCollection<
|                       | Field>..)                                                                                                                                                                                                  |
| GetHashCode           | Serves as the default hash function. (Inherited from Object.)                                                                                                                                              |
| GetInstance           | Gets the IInstance by instance path. (Inherited from ReadOnlyInstanceCollection<T>.)                                                                                                                    |
| GetInstanceByName     | Gets the IInstance by instance name. (Inherited from ReadOnlyInstanceCollection<T>.)                                                                                                                    |
| GetType               | Gets the Type of the current instance. (Inherited from Object.)                                                                                                                                             |
| IndexOf               | Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection<T>. (Inherited from 
|                       | IReadOnlyField>..)                                                                                                                                                                                          |
| MemberwiseClone       | Creates a shallow copy of the current Object. (Inherited from Object.)                                                                                                                                      |
| ToString              | Returns a string that represents the current object. (Inherited from Object.)                                                                                                                             |
| TryGetInstance         | Tries to get the instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection<T>.)                                                                                                 |
| TryGetInstanceByName   | Tries to get the instance by name. (Inherited from ReadOnlyInstanceCollection<T>.)                                                                                                                      |
| TryGetMember          | Tries to get the specified member                                                                                                                                                                          |

Reference

TwinCAT.TypeSystem Namespace [1622]

6.11.115.1 ReadOnlyFieldCollection Constructor

Initializes a new instance of the ReadOnlyMemberCollection class.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd1c3e72bc0ea15da1c14

Syntax

C#

```csharp
public ReadOnlyFieldCollection(
    FieldCollection members
)
```

Parameters

members Type: TwinCAT.TypeSystem.FieldCollection [1945] The members.

Reference

ReadOnlyFieldCollection Class [2317]

TwinCAT.TypeSystem Namespace [1622]
### 6.11.115.2 ReadOnlyFieldCollection Properties

The `ReadOnlyFieldCollection` type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>ReadOnlyCollection</code> instance. (Inherited from <code>ReadOnlyCollection</code>.)</td>
</tr>
<tr>
<td>Item(Int32)</td>
<td>Gets the element at the specified index. (Inherited from <code>ReadOnlyCollection</code>.)</td>
</tr>
<tr>
<td>Item(String)</td>
<td>Gets the element with the specified instance path. (Inherited from <code>ReadOnlyInstanceCollection</code>.)</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the <code>IList&lt;T&gt;</code> that the <code>ReadOnlyCollection&lt;T&gt;</code> wraps. (Inherited from <code>ReadOnlyInstanceCollection</code>.)</td>
</tr>
<tr>
<td>Mode</td>
<td>Gets the <code>InstanceCollectionMode</code>. (Inherited from <code>ReadOnlyInstanceCollection</code>.)</td>
</tr>
</tbody>
</table>

#### Reference

- `ReadOnlyFieldCollection Class`  
- `TwinCAT.TypeSystem Namespace`  

### 6.11.115.3 ReadOnlyFieldCollection Methods

The `ReadOnlyFieldCollection` type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String)</td>
<td>Determines whether the <code>ReadOnlyInstanceCollection</code> contains an instance with the specified instance path. (Inherited from <code>ReadOnlyInstanceCollection</code>.)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the <code>ReadOnlyCollection</code>. (Inherited from <code>ReadOnlyCollection</code>.)</td>
</tr>
<tr>
<td>ContainsName</td>
<td>Determines whether the specified instance is contained. (Inherited from <code>ReadOnlyInstanceCollection</code>.)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire <code>ReadOnlyCollection</code>, to a compatible one-dimensional <code>Array</code>, starting at the specified index of the target array. (Inherited from <code>ReadOnlyCollection</code>.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the <code>ReadOnlyCollection</code>. (Inherited from <code>ReadOnlyCollection</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetInstance</td>
<td>Gets the <code>IInstance</code> by instance path. (Inherited from <code>ReadOnlyInstanceCollection</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetInstanceByName [2513]</td>
<td>Gets the IInstance [2052] by instance name. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IField [2040].)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetInstance [2513]</td>
<td>Tries to get the instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
<tr>
<td>TryGetInstanceByName [2514]</td>
<td>Tries to get the instance by name. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
<tr>
<td>TryGetMember [2321]</td>
<td>Tries to get the specified member</td>
</tr>
</tbody>
</table>

**Reference**

ReadOnlyFieldCollection Class [2317]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.115.3.1 ReadOnlyFieldCollection.TryGetMember Method

Tries to get the specified member

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool TryGetMember(
    string fieldName,
    out TwinCAT.TypeSystem.IField symbol
)
```

**Parameters**

- **fieldName**
  - Type: `System.String`
  - Name of the member.

- **symbol**
  - Type: `TwinCAT.TypeSystem.IField`
  - The symbol.

**Return Value**

- Type: `Boolean`
  - true if found, false otherwise.
TwinCAT.Ads Namespaces

Implements

IFieldCollection.TryGetMember(String, IField) [▶ 2045]

Reference

ReadOnlyFieldCollection Class [▶ 2317]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.116   ReadonlyMemberCollection Class

Read only collection of IMember [▶ 2065] objects

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection(IMember) [▶ 2065],
  TwinCAT.TypeSystem.Generic.ReadOnlyInstanceCollection [▶ 2505], IMember [▶ 2065],
  TwinCAT.TypeSystem.ReadOnlyMemberCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class ReadonlyMemberCollection : ReadOnlyInstanceCollection<IMember>, IMemberCollection, IInstanceCollection<IMember>, IList<IMember>, ICollection<IMember>, IEnumerable<IMember>, IEnumerable

The ReadonlyMemberCollection type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TC1000
## TwinCAT.Ads Namespaces

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String)</td>
<td>Determines whether the <code>ReadOnlyInstanceCollection.T. [2505]</code> contains an instance with the specified instance path. (Inherited from <code>ReadOnlyInstanceCollection.T. [2505]</code>.)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the <code>ReadOnlyCollection.T.</code>. (Inherited from <code>ReadOnlyCollection.IMember [2065]</code>. )</td>
</tr>
<tr>
<td>ContainsName</td>
<td>Determines whether the specified instance is contained. (Inherited from <code>ReadOnlyInstanceCollection.T. [2505]</code>.)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire <code>ReadOnlyCollection.T.</code> to a compatible one-dimensional <code>Array</code>, starting at the specified index of the target array. (Inherited from <code>ReadOnlyCollection.IMember [2065]</code>.)</td>
</tr>
<tr>
<td>Empty</td>
<td>Returns an Empty <code>ReadOnlyMemberCollection</code></td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the <code>ReadOnlyCollection.T.</code>. (Inherited from <code>ReadOnlyCollection.IMember [2065]</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetInstance</td>
<td>Gets the <code>IInstance [2052]</code> by instance path. (Inherited from <code>ReadOnlyInstanceCollection.T. [2505]</code>.)</td>
</tr>
<tr>
<td>GetInstanceByName</td>
<td>Gets the <code>IInstance [2052]</code> by instance name. (Inherited from <code>ReadOnlyInstanceCollection.T. [2505]</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire <code>ReadOnlyCollection.T.</code>. (Inherited from <code>ReadOnlyCollection.IMember [2065]</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>TryGetInstance</td>
<td>Tries to get the instance with the specified instance path. (Inherited from <code>ReadOnlyInstanceCollection.T. [2505]</code>.)</td>
</tr>
<tr>
<td>TryGetInstanceByName</td>
<td>Tries to get the instance by name. (Inherited from <code>ReadOnlyInstanceCollection.T. [2505]</code>.)</td>
</tr>
<tr>
<td>TryGetMember</td>
<td>Tries to get the specified member</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]

### 6.11.116.1 `ReadOnlyMemberCollection` Constructor

Initializes a new instance of the `ReadOnlyMemberCollection [2322]` class.
Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public ReadOnlyMemberCollection(
    MemberCollection members
)
```

Parameters

members Type: TwinCAT.TypeSystem.MemberCollection [2281]
The members.

Reference

ReadOnlyMemberCollection Class [2322]
TwinCAT.TypeSystem Namespace [1622]

6.11.116.2 ReadOnlyMemberCollection Properties

The ReadOnlyMemberCollection [2322] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ReadOnlyCollection{T}.instance. (Inherited from ReadOnlyCollection.IMember [2065].)</td>
</tr>
<tr>
<td>Item.Int32.</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection.IMember [2065].)</td>
</tr>
<tr>
<td>Item.String.</td>
<td>Gets the element with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the IList{T} that the ReadOnlyCollection{T}.wraps. (Inherited from ReadOnlyCollection.IMember [2065].)</td>
</tr>
<tr>
<td>Mode</td>
<td>Gets the InstanceCollectionMode [2075]. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
</tbody>
</table>

Reference

ReadOnlyMemberCollection Class [2322]
TwinCAT.TypeSystem Namespace [1622]

6.11.116.3 ReadOnlyMemberCollection Methods

The ReadOnlyMemberCollection [2322] type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Contains(String)</code> ![2511]</td>
<td>Determines whether the <code>ReadOnlyInstanceCollection.T. ![2505]</code> contains an instance with the specified instance path. (Inherited from <code>ReadOnlyInstanceCollection.T. ![2505]</code>.)</td>
</tr>
<tr>
<td><code>Contains(T)</code></td>
<td>Determines whether an element is in the <code>ReadOnlyCollection.T.</code>. (Inherited from <code>ReadOnlyCollection.IMember ![2065]</code>.)</td>
</tr>
<tr>
<td><code>ContainsName ![2511]</code></td>
<td>Determines whether the specified instance is contained. (Inherited from <code>ReadOnlyInstanceCollection.T. ![2505]</code>.)</td>
</tr>
<tr>
<td><code>CopyTo</code></td>
<td>Copies the entire <code>ReadOnlyCollection.T.</code> to a compatible one-dimensional <code>Array</code>, starting at the specified index of the target array. (Inherited from <code>ReadOnlyCollection.IMember ![2065]</code>.)</td>
</tr>
<tr>
<td><code>Empty ![2326]</code></td>
<td>Returns an Empty <code>ReadOnlyMemberCollection ![2322]</code></td>
</tr>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object.</code>)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object.</code>)</td>
</tr>
<tr>
<td><code>GetEnumerator</code></td>
<td>Returns an enumerator that iterates through the <code>ReadOnlyCollection.T.</code>. (Inherited from <code>ReadOnlyCollection.IMember ![2065]</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object.</code>)</td>
</tr>
<tr>
<td><code>GetInstance ![2512]</code></td>
<td>Gets the <code>IInstance ![2052]</code> by instance path. (Inherited from <code>ReadOnlyInstanceCollection.T. ![2505]</code>.)</td>
</tr>
<tr>
<td><code>GetInstanceByName ![2513]</code></td>
<td>Gets the <code>IInstance ![2052]</code> by instance name. (Inherited from <code>ReadOnlyInstanceCollection.T. ![2505]</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object.</code>)</td>
</tr>
<tr>
<td><code>IndexOf</code></td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire <code>ReadOnlyCollection.T.</code>. (Inherited from <code>ReadOnlyCollection.IMember ![2065]</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object.</code>)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object.</code>)</td>
</tr>
<tr>
<td><code>TryGetInstance ![2513]</code></td>
<td>Tries to get the instance with the specified instance path. (Inherited from <code>ReadOnlyInstanceCollection.T. ![2505]</code>.)</td>
</tr>
<tr>
<td><code>TryGetInstanceByName ![2514]</code></td>
<td>Tries to get the instance by name. (Inherited from <code>ReadOnlyInstanceCollection.T. ![2505]</code>.)</td>
</tr>
<tr>
<td><code>TryGetMember ![2326]</code></td>
<td>Tries to get the specified member</td>
</tr>
</tbody>
</table>

### Reference

- `ReadOnlyMemberCollection Class ![2322]`
- `TwinCAT.TypeSystem Namespace ![1622]`
6.11.116.3.1 ReadOnlyMemberCollection.Empty Method

Returns an Empty ReadOnlyMemberCollection.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static ReadOnlyMemberCollection Empty()
```

Return Value

Type: ReadOnlyMemberCollection
ReadOnlyMemberCollection.

Reference

ReadOnlyMemberCollection Class
TwinCAT.TypeSystem Namespace

6.11.116.3.2 ReadOnlyMemberCollection.TryGetMember Method

Tries to get the specified member

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool TryGetMember(
    string memberName,
    out IMember symbol)
```

Parameters

- memberName
  Type: System.String
  Name of the member.

- symbol
  Type: TwinCAT.TypeSystem.IMember
  The symbol.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

IMemberCollection.TryGetMember(String, IMember)
Reference

ReadOnlyMemberCollection Class [2322]
TwinCAT.TypeSystem Namespace [1622]

6.11.117  ReadOnlyMethodParameterCollection Class

Read only RpcMethodParameterCollection [2384].

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.ReadOnlyMethodParameterCollection

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public class ReadOnlyMethodParameterCollection : ReadOnlyCollection<IRpcMethodParameter>,
  RpcMethodParameterCollection, IList<IRpcMethodParameter>, ICollection<IRpcMethodParameter>,
  IEnumerable<IRpcMethodParameter>, IEnumerable

The ReadOnlyMethodParameterCollection type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ReadOnlyCollection&lt;T&gt; instance. (Inherited from ReadOnlyCollection&lt;IRpcMethodParameter&gt;[2133]..)</td>
</tr>
</tbody>
</table>
| Item    | Gets the element at the specified index. (Inherited from
|         | ReadOnlyCollection<IRpcMethodParameter>[2133]..)                             |
| Items   | Returns the IList<T> that the ReadOnlyCollection<T> wraps. (Inherited from
|         | ReadOnlyCollection<IRpcMethodParameter>[2133]..)                             |

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>Determines whether an element is in the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;IRpcMethodParameter&gt;[2133]..)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire ReadOnlyCollection&lt;T&gt; to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection&lt;IRpcMethodParameter&gt;[2133]..)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;IRpcMethodParameter&gt;[2133]..)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### TwinCAT.TypeSystem Namespace [► 1622]

#### 6.11.117.1 `ReadOnlyMethodParameterCollection` Properties

The `ReadOnlyMethodParameterCollection` [► 2327] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>ReadOnlyCollection&lt;T&gt;</code> instance. (Inherited from <code>ReadOnlyCollection&lt;IRpcMethodParameter&gt;[► 2133]</code>.)</td>
</tr>
<tr>
<td>Item</td>
<td>Gets the element at the specified index. (Inherited from <code>ReadOnlyCollection&lt;IRpcMethodParameter&gt;[► 2133]</code>.)</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the <code>IList&lt;T&gt;</code> that the <code>ReadOnlyCollection&lt;T&gt;</code> wraps. (Inherited from <code>ReadOnlyCollection&lt;IRpcMethodParameter&gt;[► 2133]</code>.)</td>
</tr>
</tbody>
</table>

**Reference**

`ReadOnlyMethodParameterCollection` Class [► 2327]

TwinCAT.TypeSystem Namespace [► 1622]

#### 6.11.117.2 `ReadOnlyMethodParameterCollection` Methods

The `ReadOnlyMethodParameterCollection` [► 2327] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>Determines whether an element is in the <code>ReadOnlyCollection&lt;T&gt;</code>. (Inherited from <code>ReadOnlyCollection&lt;IRpcMethodParameter&gt;[► 2133]</code>.)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire <code>ReadOnlyCollection&lt;T&gt;</code> to a compatible one-dimensional <code>Array</code>, starting at the specified index of the target array. (Inherited from <code>ReadOnlyCollection&lt;IRpcMethodParameter&gt;[► 2133]</code>.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;IRpcMethodParameter&gt;).</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetLengthIsParameter</td>
<td>Gets the corresponding LengthIs parameter.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;IRpcMethodParameter&gt;.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Reference**

ReadOnlyMethodParameterCollection Class [2327]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.117.2.1 ReadOnlyMethodParameterCollection.GetLengthIsParameter Method

Gets the corresponding LengthIs parameter.

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdcca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public IRpcMethodParameter GetLengthIsParameter(
    IRpcMethodParameter parameter
)
```

**Parameters**

`parameter`  
Type: TwinCAT.TypeSystem.IRpcMethodParameter [2133]  
The value parameter

**Return Value**

Type: IRpcMethodParameter [2133]  
The LengthIs Parameter

**Implements**

IRpcMethodParameterCollection.GetLengthIsParameter(IRpcMethodParameter) [2139]
6.11.118  **ReadOnlyRpcMethodCollection Class**

Read only RpcMethodCollection [2364]

**Inheritance Hierarchy**

System.Object  
  TwinCAT.TypeSystem.ReadOnlyRpcMethodCollection

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public class ReadOnlyRpcMethodCollection : ReadOnlyCollection<IRpcMethod>, IRpcMethodCollection, IList<IRpcMethod>, ICollection<IRpcMethod>, IEnumerable<IRpcMethod>, IEnumerable
```

The ReadOnlyRpcMethodCollection type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ReadOnlyCollection&lt;T&gt; instance. (Inherited from ReadOnlyCollection&lt;IRpcMethod&gt;)</td>
</tr>
<tr>
<td>Item[Int32]</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection&lt;IRpcMethod&gt;)</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the IList&lt;T&gt; that the ReadOnlyCollection&lt;T&gt; wraps. (Inherited from ReadOnlyCollection&lt;IRpcMethod&gt;)</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String) [2334]</td>
<td>Determines whether this collection contains the specified method name.</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;IRpcMethod&gt;)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire ReadOnlyCollection&lt;T&gt; to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection&lt;IRpcMethod&gt;)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection. (Inherited from ReadOnlyCollection.IRpcMethod.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection. (Inherited from ReadOnlyCollection.IRpcMethod.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetMethod(Int32, IRpcMethod.)</td>
<td>Tries to get the specified method.</td>
</tr>
<tr>
<td>TryGetMethod(String, IRpcMethod.)</td>
<td>Tries to get the specified method.</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.118.1 ReadOnlyRpcMethodCollection Properties

The ReadOnlyRpcMethodCollection [2330] type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ReadOnlyCollection. (Inherited from ReadOnlyCollection.IRpcMethod.)</td>
</tr>
<tr>
<td>Item.Int32.</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection.IRpcMethod.)</td>
</tr>
<tr>
<td>Item.String.</td>
<td>Gets the IRpcMethod with the specified method name.</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the IList that the ReadOnlyCollection wraps. (Inherited from ReadOnlyCollection.IRpcMethod.)</td>
</tr>
</tbody>
</table>

### Reference

ReadOnlyRpcMethodCollection Class [2330]

TwinCAT.TypeSystem Namespace [1622]
6.11.118.1.1 **ReadOnlyRpcMethodCollection.Item Property**

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32.</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection.IRpcMethod[2123].)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the IRpcMethod[2123] with the specified method name.</td>
</tr>
</tbody>
</table>

**Reference**

**ReadOnlyRpcMethodCollection Class [2330]**

**TwinCAT.TypeSystem Namespace [1622]**

**ReadOnlyRpcMethodCollection.Item Property (String)**

Gets the IRpcMethod[2123] with the specified method name.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public IRpcMethod this[string methodName] { get; }
```

**Parameters**

- **methodName**
  - Type: `System.String`
  - Name of the method.

**Return Value**

Type: IRpcMethod[2123]

RpcMethod.

**Implements**

- IRpcMethodCollection.Item.String[2129]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyNotFoundException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

**ReadOnlyRpcMethodCollection Class [2330]**

**Item Overload [2332]**

**TwinCAT.TypeSystem Namespace [1622]**
6.11.118.2 ReadOnlyRpcMethodCollection Methods

The ReadOnlyRpcMethodCollection [2330] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String)</td>
<td>Determines whether this collection contains the specified method name.</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the ReadOnlyCollection.T. (Inherited from ReadOnlyCollection.IRpcMethod [2123].)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IRpcMethod [2123].)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection.T. (Inherited from ReadOnlyCollection.IRpcMethod [2123].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T. (Inherited from ReadOnlyCollection.IRpcMethod [2123].)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetMethod(Int32, IRpcMethod)</td>
<td>Tries to get the specified method.</td>
</tr>
<tr>
<td>TryGetMethod(String, IRpcMethod)</td>
<td>Tries to get the specified method.</td>
</tr>
</tbody>
</table>

Reference

ReadOnlyRpcMethodCollection Class [2330]

TwinCAT.TypeSystem Namespace [1622]

6.11.118.2.1 ReadOnlyRpcMethodCollection.Contains Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String)</td>
<td>Determines whether this collection contains the specified method name.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the <code>ReadOnlyCollection&lt;T&gt;</code> (Inherited from <code>ReadOnlyCollection&lt;IRpcMethod&gt;</code>.)</td>
</tr>
</tbody>
</table>

### Reference

- `ReadOnlyRpcMethodCollection Class [2330]`
- `TwinCAT.TypeSystem Namespace [1622]`

#### `ReadOnlyRpcMethodCollection.Contains Method (String)`

Determines whether this collection contains the specified method name.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool Contains(
    string methodName
)
```

**Parameters**

- `methodName`
  - Type: `System.String`
  - Name of the method.

**Return Value**

- Type: `Boolean`
  - true if contained.; otherwise, false.

**Implements**

- `IRpcMethodCollection.Contains(String) [2131]`

### Reference

- `ReadOnlyRpcMethodCollection Class [2330]`
- `Contains Overload [2333]`
- `TwinCAT.TypeSystem Namespace [1622]`

#### 6.11.118.2.2 `ReadOnlyRpcMethodCollection.TryGetMethod Method`

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>TryGetMethod(Int32, IRpcMethod.)</code></td>
<td>Tries to get the specified method.</td>
</tr>
</tbody>
</table>

---

2334 Version: 1.1 TC1000
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryGetMethod(String, IRpcMethod.)</td>
<td>Tries to get the specified method.</td>
</tr>
</tbody>
</table>

**Reference**

ReadonlyRpcMethodCollection Class [2330]

TwinCAT.TypeSystem Namespace [1622]

**ReadonlyRpcMethodCollection.TryGetMethod Method (Int32, IRpcMethod.)**

Tries to get the specified method.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```c#
public bool TryGetMethod(
    int vTableIndex,
    out IRpcMethod method
)
```

**Parameters**

- **vTableIndex**  
  Type: System.Int32
  vTableIndex.

- **method**  
  Type: TwinCAT.TypeSystem.IRpcMethod [2123].
  The method if found, NULL otherwise.

**Return Value**

Type: Boolean  
true if found, false otherwise.

**Implements**

IRpcMethodCollection.TryGetMethod(Int32, IRpcMethod.) [2132]

**Reference**

ReadonlyRpcMethodCollection Class [2330]

TryGetMethod Overload [2334]

TwinCAT.TypeSystem Namespace [1622]

**ReadonlyRpcMethodCollection.TryGetMethod Method (String, IRpcMethod.)**

Tries to get the specified method.
**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public bool TryGetMethod(
    string methodName,
    out IRpcMethod method
)
```

### Parameters

- **methodName**
  - Type: `System.String`
  - Name of the method.

- **method**
  - Type: `TwinCAT.TypeSystem.IRpcMethod`
  - The method if found, NULL otherwise.

### Return Value

- Type: `Boolean`
  - true if found, false otherwise.

### Implements

- `IRpcMethodCollection.TryGetMethod(String, IRpcMethod)` [2133]

### Reference

- `ReadOnlyRpcMethodCollection Class` [2330]
- `TryGetMethod Overload` [2334]
- `TwinCAT.TypeSystem Namespace` [1622]

### 6.11.119 ReadOnlySymbolCollection Class

ReadOnly collection containing `ISymbol` objects.

### Inheritance Hierarchy

- `System.Object`
    - `TwinCAT.TypeSystem.Generic.ReadOnlyInstanceCollection<ISymbol>` [2505]
      - `TwinCAT.TypeSystem.Generic.ReadOnlySymbolCollection<ISymbol>` [2522]
        - `TwinCAT.TypeSystem.ReadOnlySymbolCollection` [1622]
The ReadOnlySymbolCollection type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="#">ReadOnlySymbolCollection</a></td>
<td>Initializes a new instance of the ReadOnlySymbolCollection class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ReadOnlyCollection&lt;T&gt; instance. (Inherited from ReadOnlyCollection&lt;T&gt;.ISymbol...)</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection&lt;T&gt;.ISymbol...)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the element with the specified instance path. (Inherited from ReadOnlyInstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the IList&lt;T&gt; that the ReadOnlyCollection&lt;T&gt; wraps. (Inherited from ReadOnlyCollection&lt;T&gt;.ISymbol...)</td>
</tr>
<tr>
<td>Mode</td>
<td>Gets the InstanceCollectionMode. (Inherited from ReadOnlyInstanceCollection&lt;T&gt;.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String)</td>
<td>Determines whether the ReadOnlyInstanceCollection&lt;T&gt;.Contains&lt;T&gt; contains an instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;T&gt;.ISymbol...)</td>
</tr>
<tr>
<td>ContainsName</td>
<td>Determines whether the specified instance is contained. (Inherited from ReadOnlyInstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire ReadOnlyCollection&lt;T&gt; to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection&lt;T&gt;.ISymbol...)</td>
</tr>
<tr>
<td>Empty</td>
<td>Returns an Empty collection.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;T&gt;.ISymbol...)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetInstance [2512]</td>
<td>Gets the IInstance [2052] by instance path. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
<tr>
<td>GetInstanceByName [2513]</td>
<td>Gets the IInstance [2052] by instance name. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.ISymbol [2176].)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetInstance [2513]</td>
<td>Tries to get the instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
<tr>
<td>TryGetInstanceByName [2514]</td>
<td>Tries to get the instance by name. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]

### 6.11.119.1 ReadOnlySymbolCollection Constructor

Initializes a new instance of the ReadOnlySymbolCollection [2336] class.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public ReadOnlySymbolCollection(
    IInstanceCollection<ISymbol> symbols
)
```

**Parameters**

- symbols Type: TwinCAT.TypeSystem.InstanceCollection [2057]. ISymbol [2176]. The symbols.

### Reference

ReadOnlySymbolCollection Class [2336]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.119.2 ReadOnlySymbolCollection Properties

The ReadOnlySymbolCollection [2336] type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>ReadOnlyCollection&lt;T&gt;</code> instance. (Inherited from <code>ReadOnlyCollection&lt;ISymbol&gt;</code>.)</td>
</tr>
<tr>
<td>Item[Int32]</td>
<td>Gets the element at the specified index. (Inherited from <code>ReadOnlyCollection&lt;ISymbol&gt;</code>.)</td>
</tr>
<tr>
<td>Item[String,[2508]]</td>
<td>Gets the element with the specified instance path. (Inherited from <code>ReadOnlyInstanceCollection&lt;T&gt;</code>.)</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the <code>IList&lt;T&gt;</code> that the <code>ReadOnlyCollection&lt;T&gt;</code> wraps. (Inherited from <code>ReadOnlyCollection&lt;ISymbol&gt;</code>.)</td>
</tr>
<tr>
<td>Mode,[2509]</td>
<td>Gets the <code>InstanceCollectionMode</code> (Inherited from <code>ReadOnlyInstanceCollection&lt;T&gt;</code>.)</td>
</tr>
</tbody>
</table>

### Reference

`ReadOnlySymbolCollection Class`[2336]

`TwinCAT.TypeSystem Namespace`[1622]

### 6.11.119.3 `ReadOnlySymbolCollection Methods`

The `ReadOnlySymbolCollection`[2336] type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String)</td>
<td>Determines whether the <code>ReadOnlyInstanceCollection&lt;T&gt;</code> contains an instance with the specified instance path. (Inherited from <code>ReadOnlyInstanceCollection&lt;T&gt;</code>.)</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the <code>ReadOnlyCollection&lt;T&gt;</code>. (Inherited from <code>ReadOnlyCollection&lt;ISymbol&gt;</code>.)</td>
</tr>
<tr>
<td>ContainsName</td>
<td>Determines whether the specified instance is contained. (Inherited from <code>ReadOnlyInstanceCollection&lt;T&gt;</code>.)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire <code>ReadOnlyCollection&lt;T&gt;</code> to a compatible one-dimensional <code>Array</code>, starting at the specified index of the target array. (Inherited from <code>ReadOnlyCollection&lt;ISymbol&gt;</code>.)</td>
</tr>
<tr>
<td>Empty,[2340]</td>
<td>Returns an Empty collection.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the <code>ReadOnlyCollection&lt;T&gt;</code>. (Inherited from <code>ReadOnlyCollection&lt;ISymbol&gt;</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.</td>
</tr>
<tr>
<td>GetInstance,[2512]</td>
<td>Gets the <code>IInstance</code> by instance path. (Inherited from <code>ReadOnlyInstanceCollection&lt;T&gt;</code>.)</td>
</tr>
<tr>
<td>GetInstanceByName,[2513]</td>
<td>Gets the <code>IInstance</code> by instance name. (Inherited from <code>ReadOnlyInstanceCollection&lt;T&gt;</code>.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection. (Inherited from ReadOnlyCollection.ISymbol.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetInstance</td>
<td>Tries to get the instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.)</td>
</tr>
<tr>
<td>TryGetInstanceByName</td>
<td>Tries to get the instance by name. (Inherited from ReadOnlyInstanceCollection.)</td>
</tr>
</tbody>
</table>

### Reference

ReadOnlySymbolCollection Class [2336]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.119.3.1 ReadOnlySymbolCollection.Empty Method

Returns an Empty collection.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```
public static ReadOnlySymbolCollection Empty()
```

**Return Value**

Type: ReadOnlySymbolCollection [2336]

ReadOnlySymbolCollection.

**Reference**

ReadOnlySymbolCollection Class [2336]

TwinCAT.TypeSystem Namespace [1622]

#### 6.11.120 ReadOnlyTypeAttributeCollection Class

Read only version of the TypeAttributeCollection [2418]

**Inheritance Hierarchy**

System.Object

 System.Collections.ObjectModel.ReadOnlyCollection.ITypeAttribute [2209],
 TwinCAT.TypeSystem.ReadonlyTypeAttributeCollection
Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class ReadOnlyTypeAttributeCollection : ReadOnlyCollection<ITypeAttribute>,
    ITypeAttributeCollection, IList<ITypeAttribute>, ICollection<ITypeAttribute>,
    IEnumerable<ITypeAttribute>, IEnumerable
```

The `ReadOnlyTypeAttributeCollection` type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>ReadOnlyCollection&lt;T&gt;</code> instance. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>.)</td>
</tr>
<tr>
<td>Item[Int32]</td>
<td>Gets the element at the specified index. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>Item[String]</td>
<td>Gets the <code>String</code> with the specified name.</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the <code>IList&lt;T&gt;</code> that the <code>ReadOnlyCollection&lt;T&gt;</code> wraps. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>.)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String)</td>
<td>Determines whether this <code>ReadOnlyTypeAttributeCollection</code> contains the specified attribute.</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the <code>ReadOnlyCollection&lt;T&gt;</code>. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire <code>ReadOnlyCollection&lt;T&gt;</code> to a compatible one-dimensional <code>Array</code>, starting at the specified index of the target array. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>Empty</td>
<td>Returns an empty <code>ReadOnlyTypeAttributeCollection</code></td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the <code>ReadOnlyCollection&lt;T&gt;</code>. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire <code>ReadOnlyCollection&lt;T&gt;</code>. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryGetAttribute</td>
<td>Tries to get the specified ITypeAttribute</td>
</tr>
<tr>
<td>TryGetValue</td>
<td>Tries to get the specified Attribute value.</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

### 6.11.120.1 ReadOnlyTypeAttributeCollection Properties

The ReadOnlyTypeAttributeCollection [2340] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ReadOnlyCollection.T instance. (Inherited from ReadOnlyCollection.ITypeAttribute..)</td>
</tr>
<tr>
<td>Item.Int32.</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection.ITypeAttribute..)</td>
</tr>
<tr>
<td>Item.String.</td>
<td>Gets the String with the specified name.</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the IList.T that the ReadOnlyCollection.T wraps. (Inherited from ReadOnlyCollection.ITypeAttribute..)</td>
</tr>
</tbody>
</table>

**Reference**

ReadOnlyTypeAttributeCollection Class [2340]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.120.1.1 ReadOnlyTypeAttributeCollection.Item Property

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32.</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection.ITypeAttribute..)</td>
</tr>
<tr>
<td>Item.String.</td>
<td>Gets the String with the specified name.</td>
</tr>
</tbody>
</table>

**Reference**

ReadOnlyTypeAttributeCollection Class [2340]

TwinCAT.TypeSystem Namespace [1622]

### ReadOnlyTypeAttributeCollection.Item Property (String)

Gets the String with the specified name.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

C#

```csharp
public string this[string name] { get; }
```

Parameters

name Type: System.String
The name.

Return Value

Type: String
System.String.

Implements

ITypeAttributeCollection.Item.String. [2213]

Reference

ReadOnlyTypeAttributeCollection Class [2340]

Item Overload [2342]

TwinCAT.TypeSystem Namespace [1622]

6.11.120.2 ReadOnlyTypeAttributeCollection Methods

The ReadOnlyTypeAttributeCollection [2340] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String) [2344]</td>
<td>Determines whether this ReadOnlyTypeAttributeCollection [2340] contains the specified attribute.</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection ITypeAttribute [2209].)</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire ReadOnlyCollection&lt;T&gt; to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection ITypeAttribute [2209].)</td>
</tr>
<tr>
<td>Empty [2345]</td>
<td>Returns an empty ReadOnlyTypeAttributeCollection [2340]</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection ITypeAttribute [2209].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

TC1000 Version: 1.1 2343
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the</td>
</tr>
<tr>
<td></td>
<td>first occurrence within the entire ReadOnlyCollection.T. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>ReadOnlyCollection.ITypeAttribute...)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetAttribute</td>
<td>Tries to get the specified ITypeAttribute.</td>
</tr>
<tr>
<td></td>
<td>![2209]</td>
</tr>
<tr>
<td>TryGetValue</td>
<td>Tries to get the specified Attribute value.</td>
</tr>
<tr>
<td></td>
<td>![2346]</td>
</tr>
</tbody>
</table>

Reference

ReadOnlyTypeAttributeCollection Class ![2340]
TwinCAT.TypeSystem Namespace ![1622]

6.11.120.2.1 ReadOnlyTypeAttributeCollection.Contains Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String)</td>
<td>Determines whether this ReadOnlyTypeAttributeCollection ![2340] contains</td>
</tr>
<tr>
<td>![2344]</td>
<td>the specified attribute.</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the ReadOnlyCollection.T. (Inherited</td>
</tr>
<tr>
<td>![2209]</td>
<td>from ReadOnlyCollection.ITypeAttribute...)</td>
</tr>
</tbody>
</table>

Reference

ReadOnlyTypeAttributeCollection Class ![2340]
TwinCAT.TypeSystem Namespace ![1622]

ReadOnlyTypeAttributeCollection.Contains Method (String)

Determines whether this ReadOnlyTypeAttributeCollection ![2340] contains the specified attribute.

Namespace: TwinCAT.TypeSystem ![1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool Contains(
    string name
)
```

Parameters

<table>
<thead>
<tr>
<th>name</th>
<th>Type: System.String</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The name.</td>
</tr>
</tbody>
</table>
Return Value

Type: Boolean
true if [contains] [the specified name]; otherwise, false.

Implements

ITypeAttributeCollection.Contains(String) [2214]

Reference

ReadOnlyTypeAttributeCollection Class [2340]
Contains Overload [2344]
TwinCAT.TypeSystem Namespace [1622]

6.11.120.2.2  ReadOnlyTypeAttributeCollection.Empty Method

Returns an empty ReadOnlyTypeAttributeCollection [2340]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

public static ReadOnlyTypeAttributeCollection Empty()

Return Value

Type: ReadOnlyTypeAttributeCollection [2340]
ReadOnlyTypeAttributeCollection.

Reference

ReadOnlyTypeAttributeCollection Class [2340]
TwinCAT.TypeSystem Namespace [1622]

6.11.120.2.3  ReadOnlyTypeAttributeCollection.TryGetAttribute Method

Tries to get the specified ITypeAttribute [2209]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

public bool TryGetAttribute(
    string name,
    out ITypeAttribute attribute
)
Parameters

name  
Type: System.String
The name of the ITypeAttribute [2209].

attribute  
Type: TwinCAT.TypeSystem.ITypeAttribute [2209].
The attribute.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

ITypeAttributeCollection.TryGetAttribute(String, ITypeAttribute) [2215]

Reference

ReadOnlyTypeAttributeCollection Class [2340]

TwinCAT.TypeSystem Namespace [1622]

6.11.120.2.4 ReadOnlyTypeAttributeCollection.TryGetValue Method

Tries to get the specified Attribute value.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public bool TryGetValue(
    string name,
    out string value
)

Parameters

name  
Type: System.String
The name.

value  
Type: System.String.
The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

ITypeAttributeCollection.TryGetValue(String, String) [2215]
Reference

ReadOnlyTypeAttributeCollection Class [2340]

TwinCAT.TypeSystem Namespace [1622]

6.11.121 ResultDataTypes Class

Class representing the asynchronous result of reading a IDataTypeCollection [1993] via ADS. Implements the ResultValue.TValue [1029]

Inheritance Hierarchy

System Object
  TwinCAT.Ads.ResultAds [989]
    TwinCAT.Ads.ResultValue [1029], IDataTypeCollection [1995], IDataType [1986].
  TwinCAT.TypeSystem.ResultDataTypes

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch/releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public class ResultDataTypes : ResultValue<IDataTypeCollection<IDataType>>

The ResultDataTypes type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultDataTypes</td>
<td>Initializes a new instance of the ResultDataTypes class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypes</td>
<td>Gets the data types.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Value</td>
<td>The value object. (Inherited from ResultValue.TValue [1029].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>SetError</strong></td>
<td>Sets the error state of this ResultAds [997] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Remarks

The Result value of this operation can be accessed by the DataTypes [2349] property.

### Reference

TwinCAT.TypeSystem Namespace [1622]

TwinCAT.Ads.ResultValue.TValue [1029]

#### 6.11.121.1 ResultDataTypes Constructor

Initializes a new instance of the ResultDataTypes [2347] class.

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ResultDataTypes(
    AdsErrorCode errorCode,
    IDataTypeCollection<IDataType> dataTypes
)
```

**Parameters**

- **errorCode**  
  Type: TwinCAT.Ads.AdsErrorCode [575]  
  The error code.

- **dataTypes**  
  Type: TwinCAT.TypeSystem.IDataTypeCollection [1995], IDataType [1986]  
  The datatype collection.

### Reference

ResultDataTypes Class [2347]  
TwinCAT.TypeSystem Namespace [1622]

#### 6.11.121.2 ResultDataTypes Properties

The ResultDataTypes [2347] type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DataTypes</strong></td>
<td>Gets the data types.</td>
</tr>
<tr>
<td><strong>ErrorCode</strong></td>
<td>Gets the ADS Error code bound to this Result object. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td><strong>Failed</strong></td>
<td>Gets a value indicating whether the ResultAds state is failed. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td><strong>Succeeded</strong></td>
<td>Gets a value indicating whether the ResultAds state is succeeded. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td><strong>Value</strong></td>
<td>The value object. (Inherited from ResultValue.TValue.)</td>
</tr>
</tbody>
</table>

### Reference

- ResultDataTypes Class [2347]  
- TwinCAT.TypeSystem Namespace [1622]

#### 6.11.121.2.1 ResultDataTypes.DataTypes Property

Gets the data types.

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll)  
**Version:** 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fedca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public IDataTypeCollection<IDataType> DataTypes { get; }
```

**Property Value**

Type: IDataTypeCollection[IDatatype]  
The data types.

**Reference**

- ResultDataTypes Class [2347]  
- TwinCAT.TypeSystem Namespace [1622]

#### 6.11.121.3 ResultDataTypes Methods

The ResultDataTypes [2347] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError [997]</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

#### Reference

ResultDataTypes Class [2347]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.122 ResultDynamicSymbols Class

Class representing the asynchronous result of reading a dynamic symbol collection via ADS. Implements the ResultValue.TValue. [1029]

#### Inheritance Hierarchy

System.Object

TwinCAT.Ads.ResultAds [989]


TwinCAT.TypeSystem.ResultDynamicSymbols

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractsions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public class ResultDynamicSymbols : ResultSymbols<IDynamicSymbolsCollection>
```

The ResultDynamicSymbols type exposes the following members.

#### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultDynamicSymbols [2351]</td>
<td>Initializes a new instance of the ResultDynamicSymbols class.</td>
</tr>
</tbody>
</table>

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Inherited From</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed.</td>
<td>ResultAds [989]</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded.</td>
<td>ResultAds [989]</td>
</tr>
<tr>
<td>Symbols [2358]</td>
<td>Get the Symbols enumerable (T) as result of an asynchronous operation.</td>
<td>ResultSymbols.T [2356]</td>
</tr>
<tr>
<td>Value [1032]</td>
<td>The value object.</td>
<td>ResultValue.TValue [1029]</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Inherited From</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
<td>Object</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
<td>Object</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
<td>Object</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
<td>Object</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
<td>Object</td>
</tr>
<tr>
<td>SetError [997]</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989])</td>
<td>ResultAds [989]</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
<td>Object</td>
</tr>
</tbody>
</table>

### Remarks

This result class is used to return Symbol instances of dynamic nature in an IDynamicSymbolsCollection [2010]. The value/result of the completed operation can be returned by the Symbols [2358] property.

### Reference

TwinCAT.TypeSystem Namespace [1622]
TwinCAT.Ads.ResultValue.TValue [1029]

### 6.11.122.1 ResultDynamicSymbols Constructor

Initializes a new instance of the ResultDynamicSymbols [2350] class.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934ffddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ResultDynamicSymbols(    AdsErrorCode errorCode,    IDynamicSymbolsCollection symbols)
```
TwinCAT.Ads Namespaces

Parameters

errorCode  
Type: TwinCAT.Ads.AdsErrorCode [575]  
The error code.

symbols  
Type: TwinCAT.TypeSystem.IDynamicSymbolsCollection [2010]  
The result symbols.

Reference

ResultDynamicSymbols Class [2350]  
TwinCAT.TypeSystem Namespace [1622]

6.11.122.2  ResultDynamicSymbols Properties

The ResultDynamicSymbols [2350] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Symbols</td>
<td>Get the Symbols enumerable (T) as result of an asynchronous operation. (Inherited from ResultSymbols.T [2356].)</td>
</tr>
<tr>
<td>Value</td>
<td>The value object. (Inherited from ResultValue TValue [1029].)</td>
</tr>
</tbody>
</table>

Reference

ResultDynamicSymbols Class [2350]  
TwinCAT.TypeSystem Namespace [1622]

6.11.122.3  ResultDynamicSymbols Methods

The ResultDynamicSymbols [2350] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds (Inherited from ResultAds)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object)</td>
</tr>
</tbody>
</table>

**Reference**

- ResultDynamicSymbols Class [2350]
- TwinCAT.TypeSystem Namespace [1622]

### 6.11.123 ResultSymbols Class

Class representing the asynchronous result of reading an symbol collection of type `ISymbolCollection<T>` via ADS. Implements the `ResultValue.TValue`.

**Inheritance Hierarchy**

```
System.Object                                                                                               
TwinCAT.Ads.ResultAds [989]                                                                                  
TwinCAT.TypeSystem.ResultSymbols [2356] ISymbolCollection [2185] ISymbol [2176]..                           
TwinCAT.TypeSystem.ResultSymbols
```

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b436095934fadca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public class ResultSymbols : ResultSymbols<ISymbolCollection<ISymbol>>
```

The ResultSymbols type exposes the following members.

**Constructors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultSymbols</td>
<td>Initializes a new instance of the ResultSymbols class.</td>
</tr>
</tbody>
</table>

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result (Inherited from ResultAds)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds state is failed. (Inherited</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds state is succeeded. (Inherited</td>
</tr>
<tr>
<td>Symbols</td>
<td>Get the Symbols enumerable (T) as result of an asynchronous operation.</td>
</tr>
<tr>
<td>Value</td>
<td>The value object. (Inherited from ResultValue.TValue)</td>
</tr>
</tbody>
</table>
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds (Inherited from ResultAds)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Remarks
This result class is used to return generic Symbol instances in an enumerable class. The value/result of the completed operation can be returned by the Symbols property.

Reference
TwinCAT.TypeSystem Namespace
TwinCAT.Ads.ResultValue.TValue

6.11.123.1 ResultSymbols Constructor
Initializes a new instance of the ResultSymbols class.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax
C#

```csharp
public ResultSymbols(
    AdsErrorCode errorCode,
    ISymbolCollection<ISymbol> symbols
)
```

Parameters

eerrorCode Type: TwinCAT.Ads.AdsErrorCode [575]
The error code.

symbols Type: TwinCAT.TypeSystem.ISymbolCollection [2185].ISymbol [2176].
The result symbols.

Reference

ResultSymbols Class
TwinCAT.TypeSystem Namespace
6.11.123.2 ResultSymbols Properties

The ResultSymbols [2353] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Symbols</td>
<td>Get the Symbols enumerable (T) as result of an asynchronous operation. (Inherited from ResultSymbols.T [2356].)</td>
</tr>
<tr>
<td>Value</td>
<td>The value object. (Inherited from ResultValue.TValue [1029].)</td>
</tr>
</tbody>
</table>

Reference

ResultSymbols Class [2353]

TwinCAT.TypeSystem Namespace [1622]

6.11.123.3 ResultSymbols Methods

The ResultSymbols [2353] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds [989] (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

ResultSymbols Class [2353]

TwinCAT.TypeSystem Namespace [1622]
6.11.124 ResultSymbols.T. Class

Class representing the asynchronous result of reading a symbol enumeration of type IEnumerable.T via ADS. Implements the ResultValue.TValue.\[1029]\

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [989]
    TwinCAT.Ads.ResultValue [1029].T.
      TwinCAT.TypeSystem.ResultSymbols.T.
        TwinCAT.TypeSystem.ResultDynamicSymbols [2350]
          TwinCAT.TypeSystem.ResultSymbols [2353]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public class ResultSymbols<T> : ResultValue<T>
where T : class, Object, IEnumerable<ISymbol>

Type Parameters

T

The type parameter is an IEnumerable.T.

The ResultSymbols.T. type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Symbols</td>
<td>Get the Symbols enumerable (T) as result of an asynchronous operation.</td>
</tr>
<tr>
<td>Value</td>
<td>The value object. (Inherited from ResultValue.TValue.[1029].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>SetError</code></td>
<td>Sets the error state of this <code>ResultAds</code> (Inherited from <code>ResultAds</code>).</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

**Remarks**

This result class is used to return generic Symbol instances in an enumerable class. The value/result of the completed operation can be returned by the `Symbols` property.

**Reference**

- TwinCAT.TypeSystem Namespace [1622]
- TwinCAT.Ads.ResultValue.TValue [1029]

### 6.11.124.1 ResultSymbols.T. Constructor

Initializes a new instance of the `ResultSymbols.T` class.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public ResultSymbols(
    AdsErrorCode errorCode,
    T symbols
)
```

**Parameters**

- `errorCode` Type: TwinCAT.Ads.AdsErrorCode [575]
  - The error code.
- `symbols` Type: T [2356]
  - The symbols.

**Reference**

- ResultSymbols.T. Class [2356]
- TwinCAT.TypeSystem Namespace [1622]

### 6.11.124.2 ResultSymbols.T. Properties

The `ResultSymbols.T` generic type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode [992]</td>
<td>Gets the ADS Error code bound to this Result [989] object. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Failed [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is failed. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Succeeded [993]</td>
<td>Gets a value indicating whether the ResultAds [989] state is succeeded. (Inherited from ResultAds [989].)</td>
</tr>
<tr>
<td>Symbols [2358]</td>
<td>Get the Symbols enumerable (T) as result of an asynchronous operation.</td>
</tr>
<tr>
<td>Value [1032]</td>
<td>The value object. (Inherited from ResultValue.TValue [1029].)</td>
</tr>
</tbody>
</table>

Reference

ResultSymbols.T. Class [2356]
TwinCAT.TypeSystem Namespace [1622]

6.11.124.2.1 ResultSymbols.T..Symbols Property

Get the Symbols enumerable (T) as result of an asynchronous operation.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public T Symbols { get; }
```

Property Value

Type: T [2356]
The symbols.

Reference

ResultSymbols.T. Class [2356]
TwinCAT.TypeSystem Namespace [1622]

6.11.124.3 ResultSymbols.T. Methods

The ResultSymbols.T. [2356] generic type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error state of this ResultAds. (Inherited from ResultAds.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

ResultSymbols.T. Class [2356]
TwinCAT.TypeSystem Namespace [1622]

6.11.125 RpcInvokeException Class

Class RpcInvokeException. Implements the SymbolException [2401]

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [57]
      TwinCAT.TypeSystem.SymbolException [2401]
        TwinCAT.TypeSystem.RpcInvokeException

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

[SerializableAttribute]
public class RpcInvokeException : SymbolException

The RpcInvokeException type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RpcInvokeException (SerializationInfo, StreamingContext)</td>
<td>Initializes a new instance of the RpcInvokeException class.</td>
</tr>
<tr>
<td>RpcInvokeException (IRpcStructInstance, String, Int32)</td>
<td>Initializes a new instance of the RpcInvokeException class.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the instance path. (Inherited from <code>SymbolException</code>).</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol. (Inherited from <code>SymbolException</code>).</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the <code>Exception</code> that is the root cause of one or more subsequent exceptions. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the <code>SerializationInfo</code> with information about the exception. (Overrides <code>SymbolException.GetObjectData(SerializationInfo, StreamingContext)</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>
6.11.125.1  RpcInvokeException Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RpcInvokeException (SerializationInfo, StreamingContext) [2361]</td>
<td>Initializes a new instance of the RpcInvokeException [2359] class.</td>
</tr>
<tr>
<td>RpcInvokeException (IRpcStructInstance, String, Int32) [2362]</td>
<td>Initializes a new instance of the RpcInvokeException [2359] class.</td>
</tr>
</tbody>
</table>

Reference

RpcInvokeException Class [2359]

TwinCAT.TypeSystem Namespace [1622]

6.11.125.1.1  RpcInvokeException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the RpcInvokeException [2359] class.

Namespace: TwinCAT.TypeSystem [1622]  
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#  

```csharp
protected RpcInvokeException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>

Reference

RpcInvokeException Class [2359]  
RpcInvokeException Overload [2361]  
TwinCAT.TypeSystem Namespace [1622]
6.11.125.2 RpcInvokeException Constructor (IRpcStructInstance, String, Int32)

Initializes a new instance of the RpcInvokeException [2359] class.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public RpcInvokeException(
    IRpcStructInstance rpcInstance,
    string methodName,
    int errorCode
)
```

Parameters

- **rpcInstance**: Type: TwinCAT.TypeSystem.IRpcStructInstance [2140]
The RPC instance.
- **methodName**: Type: System.String
Name of the method.
- **errorCode**: Type: System.Int32
The error code.

Reference

RpcInvokeException Class [2359]
RpcInvokeException Overload [2361]
TwinCAT.TypeSystem Namespace [1622]

6.11.125.2 RpcInvokeException Properties

The RpcInvokeException [2359] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HRESULT</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the instance path. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Symbol [2411]</td>
<td>Gets the symbol. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

**Reference**

RpcInvokeException Class [2359]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.125.3 RpcInvokeException Methods

The RpcInvokeException [2359] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData [2363]</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides SymbolException.GetObjectData(SerializationInfo, StreamingContext) [2412].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

**Reference**

RpcInvokeException Class [2359]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.125.3.1 RpcInvokeException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context)
```

Parameters

- **info**
  - Type: `System.Runtime.Serialization.SerializationInfo`
  - The `SerializationInfo` that holds the serialized object data about the exception being thrown.

- **context**
  - Type: `System.Runtime.Serialization.StreamingContext`
  - The `StreamingContext` that contains contextual information about the source or destination.

Implements

- `ISerializable.GetObjectData(SerializationInfo, StreamingContext)`
- `Exception.GetObjectData(SerializationInfo, StreamingContext)`

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>info</td>
</tr>
</tbody>
</table>

Reference

- `RpcInvokeException Class [2359]`
- `TwinCAT.TypeSystem Namespace [1622]`

6.11.125.4 RpcInvokeException Events

The `RpcInvokeException [2359]` type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

Reference

- `RpcInvokeException Class [2359]`
- `TwinCAT.TypeSystem Namespace [1622]`

6.11.126 RpcMethodCollection Class

Collection of `RpcMethods [2123]`
Inheritance Hierarchy

System.Object
  TwinCAT>TypeSystem.RpcMethodCollection

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd6ca3e72bc0ea15da1c14

Syntax

C#
public class RpcMethodCollection : IRpcMethodCollection,
    IList<IRpcMethod>, ICollection<IRpcMethod>, IEnumerable<IRpcMethod>,
    IEnumerable

The RpcMethodCollection type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T.</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T is read-only.</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the IRpcMethod with the specified method name.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection.T.</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>Gets a read only collection of this RpcMethodCollection</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T.</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether this collection contains the specified method name.</td>
</tr>
<tr>
<td>ContainsItem</td>
<td>Determines whether the ICollection.T contains a specific value.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies to.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList.T.</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList.T at the specified index.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Remove [2376]</td>
<td>Removes the first occurrence of a specific object from the <code>ICollection&lt;T&gt;</code>.</td>
</tr>
<tr>
<td>RemoveAt [2376]</td>
<td>Removes the <code>IList&lt;T&gt;</code> item at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>TryGetMethod(Int32, IRpcMethod) [2377]</td>
<td>Tries to get the specified method.</td>
</tr>
<tr>
<td>TryGetMethod(String, IRpcMethod) [2378]</td>
<td>Tries to get the specified method.</td>
</tr>
</tbody>
</table>

### Reference

- **TwinCAT.TypeSystem Namespace [1622]**

### 6.11.126.1 RpcMethodCollection Properties

The `RpcMethodCollection [2364]` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [2366]</td>
<td>Gets the number of elements contained in the <code>ICollection&lt;T&gt;</code>.</td>
</tr>
<tr>
<td>IsReadOnly [2367]</td>
<td>Gets a value indicating whether the <code>ICollection&lt;T&gt;</code> is read-only.</td>
</tr>
<tr>
<td>Item[Int32] [2368]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item[String] [2368]</td>
<td>Gets the <code>IRpcMethod [2123]</code> with the specified method name.</td>
</tr>
</tbody>
</table>

### Reference

- **RpcMethodCollection Class [2364]**
- **TwinCAT.TypeSystem Namespace [1622]**

### 6.11.126.1.1 RpcMethodCollection.Count Property

Gets the number of elements contained in the `ICollection<T>`. 

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#
```
public int Count { get; }
```
**Property Value**

Type: Int32
The count.

**Implements**

ICollection.T.Count

**Reference**

RpcMethodCollection Class [2364]

TwinCAT.TypeSystem Namespace [1622]

**6.11.126.1.2 RpcMethodCollection.IsReadOnly Property**

Gets a value indicating whether the ICollection.T is read-only.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool IsReadOnly { get; }
```

**Property Value**

Type: Boolean
true if this instance is read only; otherwise, false.

**Implements**

ICollection.T.IsReadOnly

**Reference**

RpcMethodCollection Class [2364]

TwinCAT.TypeSystem Namespace [1622]

**6.11.126.1.3 RpcMethodCollection.Item Property**

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32.</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item.String.</td>
<td>Gets the IRpcMethod with the specified method name.</td>
</tr>
</tbody>
</table>


RpcMethodCollection.Item Property (Int32)

Gets or sets the element at the specified index.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public IRpcMethod this[int index] { get; set; }
```

**Parameters**

- **index**
  - Type: System.Int32
  - The index.

**Return Value**

Type: IRpcMethod

RpcMethod.

**Implements**

IList<T>.Item.Int32

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

RpcMethodCollection Class [2364]

TwinCAT.TypeSystem Namespace [1622]

RpcMethodCollection.Item Property (String)

Gets the IRpcMethod with the specified method name.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public IRpcMethod this[string methodName] { get; }
```

**Parameters**

- **methodName**
  - Type: System.String

**Reference**

RpcMethodCollection Class [2364]

Item Overload [2367]

TwinCAT.TypeSystem Namespace [1622]
Syntax

C#

```csharp
public IRpcMethod this[
    string methodName
] { get; }
```

Parameters

**methodName**
Type: `System.String`
Name of the method.

Return Value

Type: `IRpcMethod` [2123]
`RpcMethod`.

Implements

`IRpcMethodCollection.Item.String` [2129]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyNotFoundException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

`RpcMethodCollection Class` [2364]

`Item Overload` [2367]

`TwinCAT.TypeSystem Namespace` [1622]

6.11.126.2 RpcMethodCollection Methods

The `RpcMethodCollection` [2364] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Add                   | Adds an item to the `ICollection<T>`.
| AsReadOnly            | Gets a read only collection of this `RpcMethodCollection` [2364] |
| Clear                 | Removes all items from the `ICollection<T>`.
<p>| Contains(String)      | Determines whether this collection contains the specified method name. |
| Contains(IRpcMethod)  | Determines whether the <code>ICollection&lt;T&gt;</code> contains a specific value. |
| CopyTo                | Copies to. |
| Equals                | Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.) |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList.</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList at the specified index.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection.</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IList item at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetMethod</td>
<td>Tries to get the specified method.</td>
</tr>
<tr>
<td>TryGetMethod2</td>
<td>Tries to get the specified method.</td>
</tr>
</tbody>
</table>

**Reference**

RpcMethodCollection Class [2364]

TwinCAT.TypeSystem Namespace [1622]

**6.11.126.2.1 RpcMethodCollection.Add Method**

Adds an item to the ICollection.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void Add(
    IRpcMethod item
)
```

**Parameters**

item

Type: TwinCAT.TypeSystem.IRpcMethod [2123]

The object to add to the ICollection.
Implements

ICollection<T>.Add(T)

Reference

RpcMethodCollection Class [2364]

TwinCAT.TypeSystem Namespace [1622]

6.11.126.2.2 RpcMethodCollection.AsReadOnly Method

Gets a read only collection of this RpcMethodCollection [2364]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public ReadOnlyRpcMethodCollection AsReadOnly()

Field Value

Type: ReadOnlyRpcMethodCollection [2330]
Returns a read only version of this RpcMethodCollection [2364]

Return Value

Type: ReadOnlyRpcMethodCollection [2330]
ReadOnlyRpcMethodCollection.

Reference

RpcMethodCollection Class [2364]

TwinCAT.TypeSystem Namespace [1622]

6.11.126.2.3 RpcMethodCollection.Clear Method

Removes all items from the ICollection<T>.

Namespace: ICollection<T>.

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public void Clear()
6.11.126.2.4 RpcMethodCollection.Contains Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String)</td>
<td>Determines whether this collection contains the specified method name.</td>
</tr>
<tr>
<td>Contains(IRpcMethod)</td>
<td>Determines whether the ICollection.T contains a specific value.</td>
</tr>
</tbody>
</table>

**Reference**

RpcMethodCollection Class [2364]

TwinCAT.TypeSystem Namespace [1622]

**RpcMethodCollection.Contains Method (String)**

Determines whether this collection contains the specified method name.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool Contains(
    string methodName
)
```

**Parameters**

**methodName**

Type: System.String

Name of the method.

**Return Value**

Type: Boolean

ture if contained.; otherwise, false.

**Implements**

IRpcMethodCollection.Contains(String) [2131]

**Reference**

RpcMethodCollection Class [2364]
RpcMethodCollection.Contains Method (IRpcMethod)

Determines whether the ICollection.T contains a specific value.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool Contains(
    IRpcMethod item
)
```

**Parameters**

item

Type: TwinCAT.TypeSystem.IRpcMethod

The object to locate in the ICollection.T.

**Return Value**

Type: Boolean

true if item is found in the ICollection.T.; otherwise, false.

**Implements**

ICollection.T.Contains(T)

**Reference**

RpcMethodCollection Class [2364]

Contains Overload [2372]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.126.2.5 RpcMethodCollection.CopyTo Method

Copies to.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void CopyTo(
    IRpcMethod[] array,
    int arrayIndex
)
```
Parameters

array Type: TwinCAT.TypeSystem.IRpcMethod[].
The array.
arrayIndex Type: System.Int32
Index of the array.

Implements

ICollection<T>.CopyTo(T, Int32)

Reference

RpcMethodCollection Class [2364]
TwinCAT.TypeSystem Namespace [1622]

6.11.126.2.6 RpcMethodCollection.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public IEnumerator<IRpcMethod> GetEnumerator()

Return Value

Type: IEnumerator<IRpcMethod>[2123].
A IEnumerator<T> that can be used to iterate through the collection.

Implements

IEnumerable<T>.GetEnumerator.

Reference

RpcMethodCollection Class [2364]
TwinCAT.TypeSystem Namespace [1622]

6.11.126.2.7 RpcMethodCollection.IndexOf Method

Determines the index of a specific item in the IList<T>.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public int IndexOf(
    IRpcMethod item
)
```

Parameters

- **item**
  
  Type: `TwinCAT.TypeSystem.IRpcMethod`
  
  The object to locate in the `IList<T>`.

Return Value

Type: `Int32`

The index of item if found in the list; otherwise, -1.

Implements

`IList<T>.IndexOf(T)`

Reference

- `RpcMethodCollection Class [2364]`
- `TwinCAT.TypeSystem Namespace [1622]`

**6.11.126.2.8 RpcMethodCollection.Insert Method**

Inserts an item to the `IList<T>` at the specified index.

Namespace: `TwinCAT.TypeSystem [1622]`

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb91b43b6095934fddca3e72bc0e1a5da1c14

Syntax

C#

```csharp
public void Insert(
    int index,
    IRpcMethod item
)
```

Parameters

- **index**
  
  Type: `System.Int32`
  
  The zero-based index at which item should be inserted.

- **item**
  
  Type: `TwinCAT.TypeSystem.IRpcMethod`
  
  The object to insert into the `IList<T>`.

Implements

`IList<T>.Insert(Int32, T)`
6.11.126.9 RpcMethodCollection.Remove Method

Removes the first occurrence of a specific object from the ICollection<T>.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public bool Remove(
    IRpcMethod item
)
```

**Parameters**

- `item`  
  Type: TwinCAT.TypeSystem.IRpcMethod [2123]  
  The object to remove from the ICollection<T>.

**Return Value**

- Type: Boolean  
  true if item was successfully removed from the ICollection<T>; otherwise, false. This method also returns false if item is not found in the original ICollection<T>.

**Implements**

- ICollection<T>.Remove(T)

---

6.11.126.2.1 RpcMethodCollection.RemoveAt Method

Removes the IList<T> item at the specified index.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public void RemoveAt(
    int index
)
```
Parameters

index Type: System.Int32
The zero-based index of the item to remove.

Implements

IList<T>.RemoveAt(Int32)

Reference

RpcMethodCollection Class [2364]
TwinCAT.TypeSystem Namespace [1622]

6.11.126.2.1 RpcMethodCollection.TryGetMethod Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryGetMethod(Int32, IRpcMethod.)</td>
<td>Tries to get the specified method.</td>
</tr>
<tr>
<td>TryGetMethod(String, IRpcMethod.)</td>
<td>Tries to get the specified method.</td>
</tr>
</tbody>
</table>

Reference

RpcMethodCollection Class [2364]
TwinCAT.TypeSystem Namespace [1622]

RpcMethodCollection.TryGetMethod Method (Int32, IRpcMethod.)

Tries to get the specified method.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool TryGetMethod(
    int vTableIndex,
    out IRpcMethod method
)
```

Parameters

vTableIndex Type: System.Int32
VTable index.
method

Type: TwinCAT.TypeSystem.IRpcMethod [ CLAIM 2123]
The method if fund, NULL otherwise.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

IRpcMethodCollection.TryGetMethod(Int32, IRpcMethod) [ CLAIM 2132]

Reference

RpcMethodCollection Class [ CLAIM 2364]
TryGetMethod Overload [ CLAIM 2377]
TwinCAT.TypeSystem Namespace [ CLAIM 1622]

RpcMethodCollection.TryGetMethod Method (String, IRpcMethod.)

Tries to get the specified method.

Namespace: TwinCAT.TypeSystem [ CLAIM 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

public bool TryGetMethod(
    string methodName,
    out IRpcMethod method)

Parameters

methodName
Type: System.String
Name of the method.

method
Type: TwinCAT.TypeSystem.IRpcMethod [ CLAIM 2123]
The method if fund, NULL otherwise.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

IRpcMethodCollection.TryGetMethod(String, IRpcMethod) [ CLAIM 2133]

Reference

RpcMethodCollection Class [ CLAIM 2364]
TryGetMethod Overload [ CLAIM 2377]
6.11.127  RpcMethodNotSupportedException Class

Symbol Exception

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException
      TwinCAT.TypeSystem.SymbolException
        TwinCAT.TypeSystem.RpcMethodNotSupportedException

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

[SerializableAttribute]
public class RpcMethodNotSupportedException : SymbolException

The RpcMethodNotSupportedException type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RpcMethodNotSupportedException(Int32, ISymbol)</td>
<td>Initializes a new instance of the RpcMethodNotSupportedException class.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RpcMethodNotSupportedException(SerializationInfo, StreamingContext)</td>
<td>Initializes a new instance of the RpcMethodNotSupportedException class.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RpcMethodNotSupportedException(String, ISymbol)</td>
<td>Initializes a new instance of the RpcMethodNotSupportedException class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HRESULT</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstancePath</td>
<td>Gets the instance path. (Inherited from SymbolException.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol. (Inherited from SymbolException.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

#### Reference

TwinCAT.TypeSystem Namespace [1622]
### 6.11.127.1 RpcMethodNotSupportedException Constructor

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RpcMethodNotSupportedException(Int32, ISymbol)</td>
<td>Initializes a new instance of the RpcMethodNotSupportedException class.</td>
</tr>
<tr>
<td>RpcMethodNotSupportedException(StreamInfo, StreamingContext)</td>
<td>Initializes a new instance of the RpcMethodNotSupportedException class.</td>
</tr>
<tr>
<td>RpcMethodNotSupportedException(String, ISymbol)</td>
<td>Initializes a new instance of the RpcMethodNotSupportedException class.</td>
</tr>
</tbody>
</table>

**Reference**

RpcMethodNotSupportedException Class [2379]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.127.1.1 RpcMethodNotSupportedException Constructor (Int32, ISymbol)

Initializes a new instance of the RpcMethodNotSupportedException class.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public RpcMethodNotSupportedException(
    int vTableIndex,
    ISymbol symbol
)
```

**Parameters**

- **vTableIndex**
  - Type: System.Int32
  - Index of the v table.

- **symbol**
  - Type: TwinCAT.TypeSystem.ISymbol [2176]
  - The symbol.

**Reference**

RpcMethodNotSupportedException Class [2379]

RpcMethodNotSupportedException Overload [2381]

TwinCAT.TypeSystem Namespace [1622]
6.11.127.1.2 RpcMethodNotSupportedException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the RpcMethodNotSupportedException [2379] class.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#:

```csharp
protected RpcMethodNotSupportedException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

**Parameters**

SerializationInfo: Type: System.Runtime.Serialization.SerializationInfo
The serialization information.

The streaming context.

**Exceptions**

```
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>
```

**Reference**

RpcMethodNotSupportedException Class [2379]
RpcMethodNotSupportedException Overload [2381]
TwinCAT.TypeSystem Namespace [1622]

6.11.127.1.3 RpcMethodNotSupportedException Constructor (String, ISymbol)

Initializes a new instance of the RpcMethodNotSupportedException [2379] class.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstracts.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#:

```csharp
public RpcMethodNotSupportedException(
    string methodName,
    ISymbol symbol
)
```
Parameters

methodName Type: System.String
Name of the method.

symbol Type: TwinCAT.TypeSystem.ISymbol
The symbol.

Reference

RpcMethodNotSupportedException Class [2379]
RpcMethodNotSupportedException Overload [2381]
TwinCAT.TypeSystem Namespace [1622]

6.11.127.2 RpcMethodNotSupportedException Properties

The RpcMethodNotSupportedException [2379] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the instance path. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Symbol [2411]</td>
<td>Gets the symbol. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

RpcMethodNotSupportedException Class [2379]
TwinCAT.TypeSystem Namespace [1622]

6.11.127.3 RpcMethodNotSupportedException Methods

The RpcMethodNotSupportedException [2379] type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

RpcMethodNotSupportedException Class [ » 2379 ]
TwinCAT.TypeSystem Namespace [ » 1622 ]

6.11.127.4 RpcMethodNotSupportedException Events

The RpcMethodNotSupportedException [ » 2379 ] type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

RpcMethodNotSupportedException Class [ » 2379 ]
TwinCAT.TypeSystem Namespace [ » 1622 ]

6.11.128 RpcMethodParameterCollection Class

Collection of RPC method parameters

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.RpcMethodParameterCollection
Namespace: TwinCAT.TypeSystem [ » 1622 ]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
The RpcMethodParameterCollection type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection&lt;T&gt; is read-only.</td>
</tr>
<tr>
<td>Item</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>Returns a read only version of this RpcMethodParameterCollection.</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the ICollection&lt;T&gt; contains a specific value.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies to.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetLengthIsParameter</td>
<td>Gets the length is parameter.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList&lt;T&gt;.</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList&lt;T&gt; at the specified index.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection&lt;T&gt;</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IList&lt;T&gt; item at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace [1622]
6.11.128.1 RpcMethodParameterCollection Properties

The RpcMethodParameterCollection [2384] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [2386]</td>
<td>Gets the number of elements contained in the ICollection.T.</td>
</tr>
<tr>
<td>IsReadOnly [2386]</td>
<td>Gets a value indicating whether the ICollection.T. is read-only.</td>
</tr>
<tr>
<td>Item [2387]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
</tbody>
</table>

Reference

RpcMethodParameterCollection Class [2384]
TwinCAT.TypeSystem Namespace [1622]

6.11.128.1.1 RpcMethodParameterCollection.Count Property

Gets the number of elements contained in the ICollection.T.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int Count { get; }
```

Property Value

Type: Int32
The count.

Implements

ICollection.T.Count

Reference

RpcMethodParameterCollection Class [2384]
TwinCAT.TypeSystem Namespace [1622]

6.11.128.1.2 RpcMethodParameterCollection.IsReadOnly Property

Gets a value indicating whether the ICollection.T is read-only.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#
public bool IsReadOnly { get; }

Property Value

Type: Boolean
true if this instance is read only; otherwise, false.

Implements

ICollection<T>.IsReadOnly

Reference

RpcMethodParameterCollection Class [↗ 2384]
TwinCAT.TypeSystem Namespace [↗ 1622]

6.11.128.1.3 RpcMethodParameterCollection.Item Property

Gets or sets the element at the specified index.

Namespace: TwinCAT.TypeSystem [↗ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public IRpcMethodParameter this[int index] { get; set; }

Parameters

index Type: System.Int32
The index.

Return Value

Type: IRpcMethodParameter [↗ 2133]
RpcMethodParameter.

Implements

IList<T>.Item.Int32

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>
Reference
RpcMethodParameterCollection Class [2384]
TwinCAT.TypeSystem Namespace [1622]

6.11.128.2 RpcMethodParameterCollection Methods

The RpcMethodParameterCollection [2384] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add [2388]</td>
<td>Adds an item to the ICollection{T}.</td>
</tr>
<tr>
<td>AsReadOnly [2389]</td>
<td>Returns a read only version of this RpcMethodParameterCollection [2384].</td>
</tr>
<tr>
<td>Clear [2390]</td>
<td>Removes all items from the ICollection{T}.</td>
</tr>
<tr>
<td>Contains [2390]</td>
<td>Determines whether the ICollection{T} contains a specific value.</td>
</tr>
<tr>
<td>CopyTo [2391]</td>
<td>Copies to.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator [2391]</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetLengthIsParameter [2392]</td>
<td>Gets the length is parameter.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf [2392]</td>
<td>Determines the index of a specific item in the IList{T}.</td>
</tr>
<tr>
<td>Insert [2393]</td>
<td>Inserts an item to the IList{T} at the specified index.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove [2394]</td>
<td>Removes the first occurrence of a specific object from the ICollection{T}.</td>
</tr>
<tr>
<td>RemoveAt [2394]</td>
<td>Removes the IList{T} item at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference
RpcMethodParameterCollection Class [2384]
TwinCAT.TypeSystem Namespace [1622]

6.11.128.2.1 RpcMethodParameterCollection.Add Method

Adds an item to the ICollection{T}. 
Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void Add(
    IRpcMethodParameter item
)
```

Parameters

item
Type: TwinCAT.TypeSystem.IRpcMethodParameter
The object to add to the ICollection<T>.

Implements

ICollection<T>.Add(T)

Reference

RpcMethodParameterCollection Class
TwinCAT.TypeSystem Namespace

6.11.128.2.2 RpcMethodParameterCollection.AsReadOnly Method

Returns a read only version of this RpcMethodParameterCollection.

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public IReadOnlyMethodParameterCollection AsReadOnly()
```

Field Value

Type: IReadOnlyMethodParameterCollection
Collection as read only version.

Return Value

Type: IReadOnlyMethodParameterCollection
ReadOnlyMethodParameterCollection.

Reference

RpcMethodParameterCollection Class
TwinCAT.TypeSystem Namespace
6.11.128.2.3  RpcMethodParameterCollection.Clear Method

Removes all items from the ICollection.T.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#  
public void Clear();

Implements

ICollection.T.Clear.

Reference

RpcMethodParameterCollection Class [2384]
TwinCAT.TypeSystem Namespace [1622]

6.11.128.2.4  RpcMethodParameterCollection.Contains Method

Determines whether the ICollection.T contains a specific value.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#  
public bool Contains(
    IRpcMethodParameter item
)  

Parameters

item Type: TwinCAT.TypeSystem.IRpcMethodParameter [2133]
The object to locate in the ICollection.T.

Return Value

Type: Boolean
ture if item is found in the ICollection.T.; otherwise, false.

Implements

ICollection.T.Contains(T)

Reference

RpcMethodParameterCollection Class [2384]
6.11.128.2.5 RpcMethodParameterCollection.CopyTo Method

Copies to.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void CopyTo(
    IRpcMethodParameter[] array,
    int arrayIndex
)
```

**Parameters**

- **array**
  Type: TwinCAT.TypeSystem.IRpcMethodParameter [2133].
  The array.

- **arrayIndex**
  Type: System.Int32
  Index of the array.

**Implements**

ICollection<T.CopyTo(T, Int32)>

**Reference**

RpcMethodParameterCollection Class [2384]

TwinCAT.TypeSystem Namespace [1622]

6.11.128.2.6 RpcMethodParameterCollection.GetEnumerator Method

Returns an enumerator that iterates through the collection.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public IEnumerator<IRpcMethodParameter> GetEnumerator()
```

**Return Value**

Type: IEnumerator<IRpcMethodParameter> [2133].
A IEnumerator<T> that can be used to iterate through the collection.
TwinCAT.Ads Namespaces

**Implements**

IEnumerable<T>.GetEnumerator.

**Reference**

RpcMethodParameterCollection Class [2384]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.128.2.7 RpcMethodParameterCollection.GetLengthIsParameter Method

Gets the length is parameter.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:

5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public IRpcMethodParameter GetLengthIsParameter(
    IRpcMethodParameter parameter
)
```

**Parameters**

- **parameter**
  
  Type: TwinCAT.TypeSystem.IRpcMethodParameter [2133]
  
  The parameter.

**Return Value**

Type: IRpcMethodParameter [2133]

IRpcMethodParameter.

**Implements**

IRpcMethodParameterCollection.GetLengthIsParameter(IRpcMethodParameter) [2139]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>parameter</td>
</tr>
<tr>
<td>ArgumentException</td>
<td>Parameter is not contained in ParameterList - parameter</td>
</tr>
</tbody>
</table>

**Reference**

RpcMethodParameterCollection Class [2384]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.128.2.8 RpcMethodParameterCollection.IndexOf Method

Determines the index of a specific item in the `List<T>`. 
Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public int IndexOf(
    IRpcMethodParameter item
)
```

Parameters

item 
Type: TwinCAT.TypeSystem.IRpcMethodParameter
The object to locate in the `IList<T>`.

Return Value

Type: Int32
The index of `item` if found in the list; otherwise, -1.

Implements

`IList<T>.IndexOf(T)`

Reference

 RpcMethodParameterCollection Class
TwinCAT.TypeSystem Namespace

6.11.128.2.9 RpcMethodParameterCollection.Insert Method

Inserts an item to the `IList<T>` at the specified index.

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void Insert(
    int index,
    IRpcMethodParameter item
)
```

Parameters

index 
Type: System.Int32
The zero-based index at which item should be inserted.

item 
Type: TwinCAT.TypeSystem.IRpcMethodParameter
The object to insert into the `IList<T>`.
6.11.128.2.1 RpcMethodParameterCollection.Remove Method

Removes the first occurrence of a specific object from the ICollection<T>.

**Namespace:** TwinCAT.TypeSystem [» 1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public bool Remove(
    IRpcMethodParameter item
)
```

**Parameters**

- **item**
  - Type: TwinCAT.TypeSystem.IRpcMethodParameter [» 2133]
  - The object to remove from the ICollection<T>.

**Return Value**

- **Type:** Boolean
  - true if item was successfully removed from the ICollection<T>; otherwise, false. This method also returns false if item is not found in the original ICollection<T>.

**Implements**

- ICollection<T>.Remove(T)

---

6.11.128.2.1 RpcMethodParameterCollection.RemoveAt Method

Removes the IList<T> item at the specified index.

**Namespace:** TwinCAT.TypeSystem [» 1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public void RemoveAt(
    int index
)
```

**Parameters**

- **index**
  
  Type: `System.Int32`

  The zero-based index of the item to remove.

**Implements**

- `IList<T>.RemoveAt(Int32)`

**Reference**

- `RpcMethodParameterCollection Class [▶ 2384]`
- `TwinCAT.TypeSystem Namespace [▶ 1622]`

### 6.11.129 StringConvertMode Enumeration

**Enum** StringConvertMode

**Namespace:** `TwinCAT.TypeSystem [▶ 1622]`

**Assembly:** `TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`

**Syntax**

**C#**

```csharp
public enum StringConvertMode
```

**Members**

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FixedLength</td>
<td>0</td>
<td>Fixed Length String</td>
</tr>
<tr>
<td>FixedLengthZeroTerminated</td>
<td>1</td>
<td>Fixed Length String that can be terminated with '\0'</td>
</tr>
<tr>
<td>ZeroTerminated</td>
<td>2</td>
<td>'\0' terminated dynamic length string</td>
</tr>
<tr>
<td>LengthPrefix</td>
<td>3</td>
<td>Length Prefix (number of following bytes as uint 4-Byte)</td>
</tr>
</tbody>
</table>

**Remarks**

The `StringConvertMode` is used to specify, how Strings will be marshalled / demarshalled.

**Reference**

- `TwinCAT.TypeSystem Namespace [▶ 1622]`
- `StringMarshaler`
6.11.130 SymbolAccessRights Enumeration

Enum specifying Access Rights to symbols

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
[FlagsAttribute]
public enum SymbolAccessRights
```

**Members**

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>None / Uninitialized</td>
</tr>
<tr>
<td>Read</td>
<td>1</td>
<td>Read-Access</td>
</tr>
<tr>
<td>Write</td>
<td>2</td>
<td>Write-Access</td>
</tr>
<tr>
<td>MethodInvoke</td>
<td>4</td>
<td>Right to Invoke Methods / RPC Invoke</td>
</tr>
<tr>
<td>ReadWrite</td>
<td>3</td>
<td>Read / Write Access</td>
</tr>
<tr>
<td>All</td>
<td>7</td>
<td>Full Access</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

6.11.131 SymbolCollection Class

Interface represents a collection of ISymbol [2176] objects.

**Inheritance Hierarchy**

System.Object
  TwinCAT.TypeSystem.Generic.InstanceCollection [2460].ISymbol [2176].
  TwinCAT.TypeSystem.Generic.SymbolCollection [2526].ISymbol [2176].
  TwinCAT.TypeSystem.SymbolCollection

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public class SymbolCollection : SymbolCollection<ISymbol>,
    ISymbolCollection, ISymbolCollection<ISymbol>,
    IInstanceCollection<ISymbol>,
    IList<ISymbol>, ICollection<ISymbol>, IEnumerable<ISymbol>,
    IEnumerable
```

The SymbolCollection type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [2464]</td>
<td>Gets the collection count. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>InnerList [2465]</td>
<td>Gets the List of instances. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>InnerPathDict [2465]</td>
<td>The Path dictionary (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>IsReadOnly [2465]</td>
<td>Gets a value indicating whether this instance is read only. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>Item.Int32. [2466]</td>
<td>Gets or sets the IInstance [2052] at the specified index. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>Item.String. [2467]</td>
<td>Gets the IInstance [2052] with the specified instance path. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>Mode [2468]</td>
<td>The mode this InstanceCollection.T [2460] is working in. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
</tbody>
</table>

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add [2469]</td>
<td>Adds the specified item. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>AddRange [2470]</td>
<td>Adds the specified items to this collection. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>AsReadOnly [2400]</td>
<td>Returns a Read only version of this collection (shallow copy).</td>
</tr>
<tr>
<td>Clear [2471]</td>
<td>Clears this instance. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>Clone [2400]</td>
<td>Clones this instance.</td>
</tr>
<tr>
<td>Contains(String) [2472]</td>
<td>Determines whether this collection contains an IInstance [2052] with the specified InstanceName / InstancePath (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>Contains(T) [2472]</td>
<td>Determines whether this collection contains the specified IInstance [2052] (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>ContainsName [2473]</td>
<td>Determines whether the specified instance name contains name. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>CopyTo [2474]</td>
<td>Copies this InstanceCollection.T [2460] to the specified array. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>Empty [2401]</td>
<td>Returns an Empty Collection.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator [2474]</td>
<td>Gets the enumerator. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>Get Instance [2475]</td>
<td>Gets the IInstance [2052] by instance path. (Inherited from InstanceCollection.T [2460].)</td>
</tr>
</tbody>
</table>
### SymbolCollection Properties

The **SymbolCollection** type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [2464]</td>
<td>Gets the collection count. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>InnerList [2465]</td>
<td>Gets the List of instances. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>InnerPathDict [2465]</td>
<td>The Path dictionary (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>IsReadOnly [2465]</td>
<td>Gets a value indicating whether this instance is read only. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Item.Int32 [2466]</td>
<td>Gets or sets the IInstance [2052] at the specified index. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Item.String [2467]</td>
<td>Gets the IInstance [2052] with the specified instance path. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Mode [2468]</td>
<td>The mode this InstanceCollection.T. [2460] is working in. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>InstanceCollection.T. [2460].)</td>
</tr>
</tbody>
</table>

**Reference**

SymbolCollection Class [2396]
### SymbolCollection Methods

The `SymbolCollection` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add ![2469]</td>
<td>Adds the specified item. (Inherited from <code>InstanceCollection.T. ![2460]</code>)</td>
</tr>
<tr>
<td>AddRange ![2470]</td>
<td>Adds the specified items to this collection. (Inherited from <code>InstanceCollection.T. ![2460]</code>)</td>
</tr>
<tr>
<td>AsReadOnly ![2400]</td>
<td>Returns a Read only version of this collection (shallow copy).</td>
</tr>
<tr>
<td>Clear ![2471]</td>
<td>Clears this instance. (Inherited from <code>InstanceCollection.T. ![2460]</code>)</td>
</tr>
<tr>
<td>Clone ![2400]</td>
<td>Clones this instance.</td>
</tr>
<tr>
<td>Contains(String) ![2472]</td>
<td>Determines whether this collection contains an <code>IInstance ![2052]</code> with the specified InstanceName / InstancePath (Inherited from <code>InstanceCollection.T. ![2460]</code>).</td>
</tr>
<tr>
<td>Contains(T) ![2472]</td>
<td>Determines whether this collection contains the specified <code>IInstance ![2052]</code> (Inherited from <code>InstanceCollection.T. ![2460]</code>).</td>
</tr>
<tr>
<td>ContainsName ![2473]</td>
<td>Determines whether the specified instance name contains name. (Inherited from <code>InstanceCollection.T. ![2460]</code>).</td>
</tr>
<tr>
<td>CopyTo ![2474]</td>
<td>Copies this <code>InstanceCollection.T. ![2460]</code> to the specified array. (Inherited from <code>InstanceCollection.T. ![2460]</code>).</td>
</tr>
<tr>
<td>Empty ![2401]</td>
<td>Returns an Empty Collection.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object.</code>)</td>
</tr>
<tr>
<td>GetEnumerator ![2474]</td>
<td>Gets the enumerator. (Inherited from <code>InstanceCollection.T. ![2460]</code>).</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object.</code>)</td>
</tr>
<tr>
<td>GetInstanceByName ![2476]</td>
<td>Gets the name of the instance by. (Inherited from <code>InstanceCollection.T. ![2460]</code>).</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object.</code>)</td>
</tr>
<tr>
<td>IndexOf ![2476]</td>
<td>Determines the index of the specified <code>IInstance ![2052]</code>. (Inherited from <code>InstanceCollection.T. ![2460]</code>).</td>
</tr>
<tr>
<td>Insert ![2477]</td>
<td>Inserts the specified <code>IInstance ![2052]</code> at the specified index. (Inherited from <code>InstanceCollection.T. ![2460]</code>).</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object.</code>)</td>
</tr>
<tr>
<td>Remove ![2478]</td>
<td>Removes the specified item. (Inherited from <code>InstanceCollection.T. ![2460]</code>).</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RemoveAt</td>
<td>Removes the IInstance at the specified index. (Inherited from InstanceCollection.T.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetInstance</td>
<td>Tries to get the IInstance of the specified path. (Inherited from InstanceCollection.T.)</td>
</tr>
<tr>
<td>TryGetInstanceByName</td>
<td>Tries to get Instances by name. (Inherited from InstanceCollection.T.)</td>
</tr>
<tr>
<td>TryGetInstances</td>
<td>Try to get instances with predicate function (Inherited from SymbolCollection.T.)</td>
</tr>
</tbody>
</table>

### Reference

SymbolCollection Class

TwinCAT.TypeSystem Namespace

### 6.11.131.2.1 SymbolCollection.AsReadOnly Method

Returns a Read only version of this collection (shallow copy).

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public ReadOnlySymbolCollection AsReadOnly()
```

**Return Value**

Type: ReadOnlySymbolCollection

Read only collection.

**Reference**

SymbolCollection Class

TwinCAT.TypeSystem Namespace

### 6.11.131.2.2 SymbolCollection.Clone Method

Clones this instance.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public SymbolCollection Clone()
```
Return Value

Type: SymbolCollection

Cloned SymbolCollection.

Reference

SymbolCollection Class

TwinCAT.TypeSystem Namespace

6.11.131.2.3 SymbolCollection.Empty Method

Returns an Empty Collection.

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public static SymbolCollection Empty()

Return Value

Type: SymbolCollection

SymbolCollection.

Reference

SymbolCollection Class

TwinCAT.TypeSystem Namespace

6.11.132 SymbolException Class

Symbol bound exceptions

Inheritance Hierarchy

System Object

System.Exception

TwinCAT.AdsException

TwinCAT.TypeSystem.SymbolException

TwinCAT.TypeSystem.CannotAccessVirtualSymbolException

TwinCAT.TypeSystem.InsufficientAccessRightsException

TwinCAT.TypeSystem.RpcInvokeException

TwinCAT.TypeSystem.RpcMethodNotSupportedException

TwinCAT.ValueAccess.CannotAccessValueException

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
[SerializableAttribute]
public class SymbolException : AdsException
```

The SymbolException type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolException(ISymbol)</td>
<td>Initializes a new instance of the SymbolException class.</td>
</tr>
<tr>
<td>SymbolException(DeserializationInfo, StreamingContext)</td>
<td>Initializes a new instance of the SymbolException class.</td>
</tr>
<tr>
<td>SymbolException(String, ISymbol)</td>
<td>Initializes a new instance of the SymbolException class.</td>
</tr>
<tr>
<td>SymbolException(ISymbol, Exception)</td>
<td>Initializes a new instance of the SymbolException class.</td>
</tr>
<tr>
<td>SymbolException(ISymbol, Int32)</td>
<td>Initializes a new instance of the SymbolException class.</td>
</tr>
<tr>
<td>SymbolException(String, ISymbol, Exception)</td>
<td>Initializes a new instance of the SymbolException class.</td>
</tr>
<tr>
<td>SymbolException(String, ISymbol, Int32, Exception)</td>
<td>Initializes a new instance of the SymbolException class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

**Name** | **Description**
--- | ---
HelpLink | Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
HRESULT | Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
InnerException | Gets the Exception instance that caused the current exception. (Inherited from Exception.)
InstancePath | Gets the instance path.
Message | Gets a message that describes the current exception. (Inherited from Exception.)
Source | Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
StackTrace | Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
Symbol | Gets the symbol.
TargetSite | Gets the method that throws the current exception. (Inherited from Exception.)

### Methods

**Name** | **Description**
--- | ---
Equals | Determines whether the specified object is equal to the current object. (Inherited from Object.)
Finalize | Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
GetBaseException | When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
GetHashCode | Serves as the default hash function. (Inherited from Object.)
GetObjectData | When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overridess Exception.GetObjectData(SerializationInfo, StreamingContext).)
GetType | Gets the runtime type of the current instance. (Inherited from Exception.)
MemberwiseClone | Creates a shallow copy of the current Object. (Inherited from Object.)
ToString | Creates and returns a string representation of the current exception. (Inherited from Exception.)

### Events

**Name** | **Description**
--- | ---
SerializeObjectState | Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

### Reference

TwinCAT.TypeSystem Namespace [1622]
### 6.11.132.1 SymbolException Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolException(ISymbol)</td>
<td>Initializes a new instance of the SymbolException [2401] class.</td>
</tr>
<tr>
<td>SymbolException(SerializationInfo, StreamingContext)</td>
<td>Initializes a new instance of the SymbolException [2401] class.</td>
</tr>
<tr>
<td>SymbolException(String, ISymbol)</td>
<td>Initializes a new instance of the SymbolException [2401] class.</td>
</tr>
<tr>
<td>SymbolException(ISymbol, Exception)</td>
<td>Initializes a new instance of the SymbolException [2401] class.</td>
</tr>
<tr>
<td>SymbolException(ISymbol, Int32)</td>
<td>Initializes a new instance of the SymbolException [2401] class.</td>
</tr>
<tr>
<td>SymbolException(String, ISymbol, Exception)</td>
<td>Initializes a new instance of the SymbolException [2401] class.</td>
</tr>
<tr>
<td>SymbolException(String, ISymbol, Int32)</td>
<td>Initializes a new instance of the SymbolException [2401] class.</td>
</tr>
<tr>
<td>SymbolException(ISymbol, Int32, Exception)</td>
<td>Initializes a new instance of the SymbolException [2401] class.</td>
</tr>
<tr>
<td>SymbolException(String, ISymbol, Int32, Exception)</td>
<td>Initializes a new instance of the SymbolException [2401] class.</td>
</tr>
</tbody>
</table>

#### Reference

- SymbolException Class [2401]
- TwinCAT.TypeSystem Namespace [1622]

### 6.11.132.1.1 SymbolException Constructor (ISymbol)

Initializes a new instance of the SymbolException [2401] class.
**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public SymbolException(
    ISymbol symbol
)
```

**Parameters**

- `symbol`: Type: TwinCAT.TypeSystem.ISymbol [2176]
  The symbol.

### Reference

- SymbolException Class [2401]
- SymbolException Overload [2404]
- TwinCAT.TypeSystem Namespace [1622]

#### 6.11.132.1.2 SymbolException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the SymbolException class.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected SymbolException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

**Parameters**

- `serializationInfo`: Type: System.Runtime.Serialization.SerializationInfo
  The serialization information.
  The streaming context.

### Reference

- SymbolException Class [2401]
- SymbolException Overload [2404]
- TwinCAT.TypeSystem Namespace [1622]

#### 6.11.132.1.3 SymbolException Constructor (String, ISymbol)

Initializes a new instance of the SymbolException class.
**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public SymbolException(
    string message,
    ISymbol symbol
)
```

**Parameters**

**message**

Type: System.String

The message.

**symbol**

Type: TwinCAT.TypeSystem.ISymbol [2176]

The symbol.

**Reference**

SymbolException Class [2401]

SymbolException Overload [2404]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.132.1.4 SymbolException Constructor (ISymbol, Exception)

Initializes a new instance of the SymbolException [2401] class.

**Namespace:** TwinCAT.TypeSystem [1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public SymbolException(
    ISymbol symbol,
    Exception innerException
)
```

**Parameters**

**symbol**

Type: TwinCAT.TypeSystem.ISymbol [2176]

The symbol.

**innerException**

Type: System.Exception

The inner exception.

**Reference**

SymbolException Class [2401]

SymbolException Overload [2404]

TwinCAT.TypeSystem Namespace [1622]
6.11.132.1.5 SymbolException Constructor (ISymbol, Int32)

Initializes a new instance of the SymbolException class.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#  

```csharp
public SymbolException(
    ISymbol symbol,
    int errorCode
)
```

Parameters

- **symbol**  
  Type: TwinCAT.TypeSystem.ISymbol  
  The symbol.

- **errorCode**  
  Type: System.Int32  
  The error code.

Reference

SymbolException Class
SymbolException Overload
TwinCAT.TypeSystem Namespace

6.11.132.1.6 SymbolException Constructor (String, ISymbol, Exception)

Initializes a new instance of the SymbolException class.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#  

```csharp
public SymbolException(
    string message,  
    ISymbol symbol,  
    Exception innerException
)
```

Parameters

- **message**  
  Type: System.String  
  The message.

- **symbol**  
  Type: TwinCAT.TypeSystem.ISymbol  
  The symbol.

- **innerException**  
  Type: System.Exception  
  The inner exception.
6.11.132.1.7 SymbolException Constructor (String, ISymbol, Int32)

Initializes a new instance of the SymbolException class.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public SymbolException(
    string message,
    ISymbol symbol,
    int errorCode
)
```

**Parameters**

- **message**
  - Type: `System.String`
  - The message.

- **symbol**
  - Type: `TwinCAT.TypeSystem.ISymbol`
  - The symbol.

- **errorCode**
  - Type: `System.Int32`
  - The error code.

Reference

SymbolException Class [2401]

SymbolException Overload [2404]

TwinCAT.TypeSystem Namespace [1622]

6.11.132.1.8 SymbolException Constructor (ISymbol, Int32, Exception)

Initializes a new instance of the SymbolException class.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public SymbolException(
    ISymbol symbol,
    int errorCode,
    Exception innerException
)
```
Parameters

symbol Type: TwinCAT.TypeSystem.ISymbol
The symbol.

errorCode Type: System.Int32
The error code.

innerException Type: System.Exception
The inner exception.

Reference

SymbolException Class [2401]
SymbolException Overload [2404]
TwinCAT.TypeSystem Namespace [1622]

6.11.132.1.9 SymbolException Constructor (String, ISymbol, Int32, Exception)
Initializes a new instance of the SymbolException [2401] class.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public SymbolException(
    string message,
    ISymbol symbol,
    int errorCode,
    Exception innerException
)

Parameters

message Type: System.String
The message.

symbol Type: TwinCAT.TypeSystem.ISymbol
The symbol.

errorCode Type: System.Int32
The error code.

innerException Type: System.Exception
The inner exception.

Reference

SymbolException Class [2401]
SymbolException Overload [2404]
TwinCAT.TypeSystem Namespace [1622]

6.11.132.2 SymbolException Properties
The SymbolException [2401] type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>HResult</strong></td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>InstancePath</strong></td>
<td>Gets the instance path.</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>Symbol</strong></td>
<td>Gets the symbol.</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

### Reference

- `SymbolException Class` [2401]
- `TwinCAT.TypeSystem Namespace` [1622]

### 6.11.132.2.1 SymbolException.InstancePath Property

Gets the instance path.

**Namespace:**  `TwinCAT.TypeSystem` [1622]  
**Assembly:**  `TwinCAT.Ads.Abstractions` (in `TwinCAT.Ads.Abstractions.dll`)  
Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public string InstancePath { get; }
```

### Property Value

**Type:** `String`  
The instance path.

### Reference

- `SymbolException Class` [2401]
- `TwinCAT.TypeSystem Namespace` [1622]
6.11.132.2.2 SymbolException.Symbol Property

Gets the symbol.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ISymbol Symbol { get; }
```

**Property Value**

Type: ISymbol

The symbol.

**Reference**

SymbolException Class

TwinCAT.TypeSystem Namespace

6.11.132.3 SymbolException Methods

The SymbolException type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

**Reference**

SymbolException Class

TwinCAT.TypeSystem Namespace
6.11.132.3.1 SymbolException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)

Parameters

info Type: System.Runtime.Serialization.SerializationInfo
The SerializationInfo that holds the serialized object data about the exception being thrown.

context Type: System.Runtime.Serialization.StreamingContext
The StreamingContext that contains contextual information about the source or destination.

Implements

ISerializable.GetObjectData(SerializationInfo, StreamingContext)
Exception.GetObjectData(SerializationInfo, StreamingContext)

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>info</td>
</tr>
</tbody>
</table>

Reference

SymbolException Class
TwinCAT.TypeSystem Namespace

6.11.132.4 SymbolException Events

The SymbolException type exposes the following members.

Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerializeObjectState</td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

Reference

SymbolException Class
TwinCAT.TypeSystem Namespace [► 1622]

### 6.11.133 TypeAttribute Class

**ADS Attribute**

**Inheritance Hierarchy**

- System.Object
- TwinCAT.TypeSystem.TypeAttribute

**Namespace:** TwinCAT.TypeSystem [► 1622]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public class TypeAttribute : ITypeAttribute
```

The TypeAttribute type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name [► 2414]</td>
<td>Name of the Attribute</td>
</tr>
<tr>
<td>Value [► 2414]</td>
<td>Gets the value of the attribute</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals [► 2415]</td>
<td>Equals (Overrides Object.Equals(Object).)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Operators**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equality [► 2417]</td>
<td>Operator==</td>
</tr>
<tr>
<td>Inequality [► 2417]</td>
<td>Implements the != operator.</td>
</tr>
</tbody>
</table>
6.11.133.1 TypeAttribute Properties

The TypeAttribute type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the Attribute</td>
</tr>
<tr>
<td>Value</td>
<td>Gets the value of the attribute</td>
</tr>
</tbody>
</table>

Reference

TypeAttribute Class [2413]

TwinCAT.TypeSystem Namespace [1622]

6.11.133.1.1 TypeAttribute.Name Property

Name of the Attribute

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads.Abstracts (in TwinCAT.Ads.Abstracts.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public string Name { get; }
```

Property Value

Type: String

The name.

Implements

ITypeAttribute.Name [2210]

Reference

TypeAttribute Class [2413]

TwinCAT.TypeSystem Namespace [1622]

6.11.133.1.2 TypeAttribute.Value Property

Gets the value of the attribute
**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public string Value { get; }
```

**Property Value**

Type: `String`

The value.

**Implements**

`ITypeAttribute.Value` [2210]

**Reference**

TypeAttribute Class [2413]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.133.2 TypeAttribute Methods

The `TypeAttribute` [2413] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Equals (Overrides <code>Object.Equals(Object)</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the HashCode of the Address (Overrides <code>Object.GetHashCode</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

**Reference**

TypeAttribute Class [2413]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.133.2.1 TypeAttribute.Equals Method

Equals
**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public override bool Equals(
    Object obj
)
```

**Parameters**

**obj**

Type: System.Object

The object to compare with the current object.

**Return Value**

Type: **Boolean**

true if the specified Object is equal to this instance; otherwise, false.

### Reference

**TypeAttribute Class**

TwinCAT.TypeSystem Namespace

6.11.133.2.2 *TypeAttribute.GetHashCode Method*

Gets the HashCode of the Address

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public override int GetHashCode()
```

**Return Value**

Type: **Int32**

A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

### Reference

**TypeAttribute Class**

TwinCAT.TypeSystem Namespace

6.11.133.3 *TypeAttribute Operators*

The *TypeAttribute* type exposes the following members.
### Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equality [2417]</td>
<td>Operator==</td>
</tr>
<tr>
<td>Inequality [2417]</td>
<td>Implements the != operator.</td>
</tr>
</tbody>
</table>

### Reference

**TypeAttribute Class [2413]**

**TwinCAT.TypeSystem Namespace [1622]**

#### 6.11.133.3.1 TypeAttribute.Equality Operator

**Operator==**

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static bool operator ==(
    TypeAttribute o1,
    TypeAttribute o2
)
```

**Parameters**

- `o1` Type: TwinCAT.TypeSystem.TypeAttribute [2413]
  - The o1.

- `o2` Type: TwinCAT.TypeSystem.TypeAttribute [2413]
  - The o2.

**Return Value**

Type: Boolean
- The result of the operator.

**Reference**

TypeAttribute Class [2413]

**TwinCAT.TypeSystem Namespace [1622]**

#### 6.11.133.3.2 TypeAttribute.Inequality Operator

**Implements the != operator.**

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```csharp
public static bool operator !=(
    TypeAttribute o1,
    TypeAttribute o2
)
```

Parameters

- **o1**
  - Type: `TwinCAT.TypeSystem.TypeAttribute`
  - The `o1`.
- **o2**
  - Type: `TwinCAT.TypeSystem.TypeAttribute`
  - The `o2`.

Return Value

- Type: `Boolean`
  - The result of the operator.

Reference

- `TypeAttribute Class` [2413]
- `TwinCAT.TypeSystem Namespace` [1622]

### 6.11.134 TypeAttributeCollection Class

Collection of `AdsAttributes` [2209]

Inheritance Hierarchy

- `System.Object`
  - `TwinCAT.TypeSystem.TypeAttributeCollection`

Namespace: `TwinCAT.TypeSystem` [1622]

Assembly: `TwinCAT.Ads` (in `TwinCAT.Ads.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class TypeAttributeCollection : ITypeAttributeCollection,
    IList<ITypeAttribute>, ICollection<ITypeAttribute>, IEnumerable<ITypeAttribute>,
    IEnumerable
```

The `TypeAttributeCollection` type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="2420" alt="TypeAttributeCollection" /></td>
<td>Initializes a new instance of the <code>TypeAttributeCollection</code> class.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TypeAttributeCollection(IEnumerable&gt;TypeAttribute.)</td>
<td>Initializes a new instance of the TypeAttributeCollection class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T.</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T is read-only.</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the String with the specified name.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection.T.</td>
</tr>
<tr>
<td>AddRange</td>
<td>Adds the range.</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>Gets a read only version of this TypeAttributeCollection</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T.</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether this TypeAttributeCollection contains the ITypeAttribute with the specified name.</td>
</tr>
<tr>
<td>Contains(ITypeAttribute)</td>
<td>Determines whether the ICollection.T contains a specific value.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies to.</td>
</tr>
<tr>
<td>Empty</td>
<td>Returns an Empty TypeAttributeCollection.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the_IList.T.</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the_IList.T at the specified index.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove(String)</td>
<td>Removes the specified ITypeAttribute [2209] from the TypeAttributeCollection.</td>
</tr>
<tr>
<td>Remove(ITypeAttribute)</td>
<td>Removes the first occurrence of a specific object from the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IList&lt;T&gt; item at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetAttribute</td>
<td>Tries to get the specified ITypeAttribute [2209].</td>
</tr>
<tr>
<td>TryGetValue</td>
<td>Tries to get the specified Attribute value.</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem Namespace [1622]

### 6.11.134.1 TypeAttributeCollection Constructor

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TypeAttributeCollection</td>
<td>Initializes a new instance of the TypeAttributeCollection [2418] class.</td>
</tr>
<tr>
<td>TypeAttributeCollection(IEnumerable.ITypeAttribute)</td>
<td>Initializes a new instance of the TypeAttributeCollection [2418] class.</td>
</tr>
</tbody>
</table>

**Reference**

TypeAttributeCollection Class [2418]

TwinCAT.TypeSystem Namespace [1622]

### 6.11.134.1.1 TypeAttributeCollection Constructor

Initializes a new instance of the TypeAttributeCollection [2418] class.

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public TypeAttributeCollection() 
```

**Reference**

TypeAttributeCollection Class [2418]
6.11.134.1.2 TypeAttributeCollection Constructor (IEnumerable.ITypeAttribute.)

Initializes a new instance of the TypeAttributeCollection class.

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

**C#**

```csharp
public TypeAttributeCollection(
    IEnumerable<ITypeAttribute> coll
)
```

Parameters

coll  
Type: System.Collections.Generic.IEnumerable.ITypeAttribute

The coll.

Reference

TypeAttributeCollection Class
TypeAttributeCollection Overload
TwinCAT.TypeSystem Namespace

6.11.134.2 TypeAttributeCollection Properties

The TypeAttributeCollection type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [2421]</td>
<td>Gets the number of elements contained in the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>IsReadOnly [2422]</td>
<td>Gets a value indicating whether the ICollection&lt;T&gt; is read-only.</td>
</tr>
<tr>
<td>Item.Int32 [2423]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item.String [2423]</td>
<td>Gets the string with the specified name.</td>
</tr>
</tbody>
</table>

Reference

TypeAttributeCollection Class
TwinCAT.TypeSystem Namespace

6.11.134.2.1 TypeAttributeCollection.Count Property

Gets the number of elements contained in the ICollection<T>.
Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Count { get; }
```

Property Value

Type: Int32
The count.

Implements

ICollection<T>.Count

Reference

TypeAttributeCollection Class

TwinCAT.TypeSystem Namespace

6.11.134.2.2 TypeAttributeCollection.IsReadOnly Property

Gets a value indicating whether the ICollection<T> is read-only.

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReadOnly { get; }
```

Property Value

Type: Boolean
true if this instance is read only; otherwise, false.

Implements

ICollection<T>.IsReadOnly

Reference

TypeAttributeCollection Class

TwinCAT.TypeSystem Namespace
### 6.11.134.2.3 TypeAttributeCollection.Item Property

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32. [2423]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item.String. [2423]</td>
<td>Gets the String with the specified name.</td>
</tr>
</tbody>
</table>

#### Reference

- TypeAttributeCollection Class [2418]
- TwinCAT.TypeSystem Namespace [1622]

### TypeAttributeCollection.Item Property (Int32)

Gets or sets the element at the specified index.

**Namespace:** TwinCAT.TypeSystem [1622]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public ITypeAttribute this[int index] { get; set; }
```

#### Parameters

- `index`  
  - Type: System.Int32  
  - The index.

#### Return Value

Type: ITypeAttribute [2209]  
AdsAttribute.

#### Implements

IList<T>.Item.Int32.

#### Reference

- TypeAttributeCollection Class [2418]
- Item Overload [2423]
- TwinCAT.TypeSystem Namespace [1622]

### TypeAttributeCollection.Item Property (String)

Gets the String with the specified name.
**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294*Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

---

**Syntax**

### C#

```csharp
public string this[string name] { get; }
```

**Parameters**

- **name**: Type: `System.String`
  
  The name.

**Return Value**

Type: `String`

`System.String`.

**Implements**

`ITypeAttributeCollection.Item.String` [2213]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyNotFoundException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

- TypeAttributeCollection Class [2418]
- Item Overload [2423]
- TwinCAT.TypeSystem Namespace [1622]

### 6.11.134.3 TypeAttributeCollection Methods

The TypeAttributeCollection [2418] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>AddRange</td>
<td>Adds the range.</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>Gets a read only version of this TypeAttributeCollection [2418].</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether this TypeAttributeCollection [2418] contains the ITypeAttribute [2209] with the specified name.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Contains(ITypeAttribute)</td>
<td>Determines whether the ICollection.T. contains a specific value.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies to.</td>
</tr>
<tr>
<td>Empty</td>
<td>Returns an Empty TypeAttributeCollection.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList.T.</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList.T. at the specified index.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove(String)</td>
<td>Removes the specified ITypeAttribute from the TypeAttributeCollection.</td>
</tr>
<tr>
<td>Remove(ITypeAttribute)</td>
<td>Removes the first occurrence of a specific object from the ICollection.T.</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IList.T. item at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetAttribute</td>
<td>Tries to get the specified ITypeAttribute.</td>
</tr>
<tr>
<td>TryGetValue</td>
<td>Tries to get the specified Attribute value.</td>
</tr>
</tbody>
</table>

Reference

TypeAttributeCollection Class

TwinCAT.TypeSystem Namespace

6.11.134.3.1 TypeAttributeCollection.Add Method

Adds an item to the ICollection.T.

Namespace: TwinCAT.TypeSystem

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:

5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public void Add(ITypeAttribute item)
```

Parameters

- **item**: Type: `TwinCAT.TypeSystem.ITypeAttribute`<br>The object to add to the `ICollection<T>`.

Implements

- `ICollection<T>.Add(T)`

Reference

- `TypeAttributeCollection Class` [2418]
- `TwinCAT.TypeSystem Namespace` [1622]

### 6.11.134.3.2 TypeAttributeCollection.AddRange Method

Adds the range.

**Namespace**: `TwinCAT.TypeSystem` [1622]<br>**Assembly**: `TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`

Syntax

C#

```csharp
public void AddRange(IEnumerable<ITypeAttribute> items)
```

Parameters

- **items**: Type: `System.Collections.Generic.IEnumerable`<br>`ITypeAttribute`<br>The items.

Reference

- `TypeAttributeCollection Class` [2418]
- `TwinCAT.TypeSystem Namespace` [1622]

### 6.11.134.3.3 TypeAttributeCollection.AsReadOnly Method

Gets a read only version of this `TypeAttributeCollection`.

**Namespace**: `TwinCAT.TypeSystem` [1622]<br>**Assembly**: `TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`
Syntax

C#

```csharp
public ReadOnlyTypeAttributeCollection AsReadOnly()
```

Field Value

Type: `ReadOnlyTypeAttributeCollection` [2340]
As read only.

Return Value

Type: `ReadOnlyTypeAttributeCollection` [2340]
`ReadOnlyAttributeCollection`.

Reference

`TypeAttributeCollection Class` [2418]
`TwinCAT.TypeSystem Namespace` [1622]

6.11.134.3.4 TypeAttributeCollection.Clear Method

Removes all items from the `ICollection<T>`.

Namespace: `TwinCAT.TypeSystem` [1622]
Assembly: `TwinCAT.Ads` (in `TwinCAT.Ads.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void Clear()
```

Implements

`IICollection<T.Clear>`

Reference

`TypeAttributeCollection Class` [2418]
`TwinCAT.TypeSystem Namespace` [1622]

6.11.134.3.5 TypeAttributeCollection.Contains Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String) [2428]</td>
<td>Determines whether this <code>TypeAttributeCollection</code> [2418] contains the <code>ITypeAttribute</code> [2209] with the specified name.</td>
</tr>
<tr>
<td>Contains(ITypeAttribute) [2428]</td>
<td>Determines whether the <code>IICollection&lt;T&gt;</code> contains a specific value.</td>
</tr>
</tbody>
</table>
TypeAttributeCollection.Contains Method (String)

Determines whether this `TypeAttributeCollection` contains the `ITypeAttribute` with the specified name.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool Contains(
    string name
)
```

**Parameters**

- name
  - Type: System.String
  - The name.

**Return Value**

- Type: Boolean
  - true if [contains] [the specified name]; otherwise, false.

**Implements**

- `ITypeAttributeCollection.Contains(String)`

**Reference**

- `TypeAttributeCollection Class` [2418]
- `TwinCAT.TypeSystem Namespace` [1622]

- `Contains Overload` [2427]

- `TwinCAT.TypeSystem Namespace` [1622]

TypesAttributeCollection.Contains Method (ITypeAttribute)

Determines whether the `ICollection<T>` contains a specific value.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
**Syntax**

**C#**

```csharp
public bool Contains(ITypeAttribute item)
```

**Parameters**

- `item`: Type: `TwinCAT.TypeSystem.ITypeAttribute`<sup>2209</sup>
  - The object to locate in the `ICollection<T>`.

**Return Value**

Type: `Boolean`
- true if `item` is found in the `ICollection<T>`; otherwise, false.

**Implements**

- `ICollection<T>.Contains(T)`

**Reference**

- `TypeAttributeCollection Class [】2418]`
- `Contains Overload [】2427]`
- `TwinCAT.TypeSystem Namespace [】1622]`

### 6.11.134.3.6 TypeAttributeCollection.CopyTo Method

Copies to.

**Namespace:** `TwinCAT.TypeSystem [】1622]`

**Assembly:** `TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14`

**Syntax**

**C#**

```csharp
public void CopyTo(ITypeAttribute[] array, int arrayIndex)
```

**Parameters**

- `array`: Type: `TwinCAT.TypeSystem.ITypeAttribute`<sup>2209</sup>
  - The array.

- `arrayIndex`: Type: `System.Int32`
  - Index of the array.

**Implements**

- `ICollection<T>.CopyTo(T, Int32)`
6.11.134.3.7 TypeAttributeCollection.Empty Method

Returns an Empty TypeAttributeCollection.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public static TypeAttributeCollection Empty()
```

**Return Value**

Type: TypeAttributeCollection.

Reference

TypeAttributeCollection Class

TwinCAT.TypeSystem Namespace

6.11.134.3.8 TypeAttributeCollection.GetEnumerator Method

Returns an enumerator that iterates through the collection.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public IEnumerator<ITypeAttribute> GetEnumerator()
```

**Return Value**

Type: IEnumerator.ITypeAttribute.

A IEnumerator.T. that can be used to iterate through the collection.

**Implements**

IEnumerator.T.GetEnumerator.

Reference

TypeAttributeCollection Class
### 6.11.134.3.9 TypeAttributeCollection.IndexOf Method

Determines the index of a specific item in the `IList<T>`.

**Namespace:**  TwinCAT.TypeSystem

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public int IndexOf(ITypeAttribute item)
```

**Parameters**

- `item`  
  Type: `TwinCAT.TypeSystem.ITypeAttribute`  
  The object to locate in the `IList<T>`.

**Return Value**

- **Type:** `Int32`  
  The index of `item` if found in the list; otherwise, -1.

**Implements**

- `IList<T>.IndexOf(T)`

**Reference**

- `TypeAttributeCollection Class [p.2418]`
- `TwinCAT.TypeSystem Namespace [p.1622]

### 6.11.134.3.1 TypeAttributeCollection.Insert Method

Inserts an item to the `IList<T>` at the specified index.

**Namespace:**  TwinCAT.TypeSystem

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void Insert(int index, ITypeAttribute item)
```
Parameters

index Type: System.Int32
The zero-based index at which item should be inserted.

item Type: TwinCAT.TypeSystem.ITypeAttribute
The object to insert into the IList.

Implements

IList<T>.Insert(Int32, T)

Reference

TypeAttributeCollection Class [2418]
TwinCAT.TypeSystem Namespace [1622]

6.11.134.3.1 TypeAttributeCollection.Remove Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove(String) [2432]</td>
<td>Removes the specified ITypeAttribute [2209] from the TypeAttributeCollection [2418]</td>
</tr>
<tr>
<td>Remove(ITypeAttribute) [2433]</td>
<td>Removes the first occurrence of a specific object from the ICollection&lt;T&gt;.</td>
</tr>
</tbody>
</table>

Reference

TypeAttributeCollection Class [2418]
TwinCAT.TypeSystem Namespace [1622]

TypeAttributeCollection.Remove Method (String)

Removes the specified ITypeAttribute [2209] from the TypeAttributeCollection [2418]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public bool Remove(
    string name
)
```

Parameters

name Type: System.String
The name.
Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

TypeAttributeCollection Class [2418]
Remove Overload [2432]
TwinCAT.TypeSystem Namespace [1622]

**TypeAttributeCollection**.Remove Method (ITypeAttribute)

Removes the first occurrence of a specific object from the ICollection.T.

**Namespace:** TwinCAT.TypeSystem [1622]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd1ca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool Remove(ITypeAttribute item)
```

**Parameters**

- item
  
  Type: TwinCAT.TypeSystem.ITypeAttribute[2209]

  The object to remove from the ICollection.T.

**Return Value**

Type: Boolean
true if item was successfully removed from the ICollection.T.; otherwise, false. This method also returns false if item is not found in the original ICollection.T.

**Implements**

ICollection<T>.Remove(T)

**Reference**

TypeAttributeCollection Class [2418]
Remove Overload [2432]
TwinCAT.TypeSystem Namespace [1622]

6.11.134.3.1 TypeAttributeCollection.RemoveAt Method 2

Removes the IList<T> item at the specified index.
Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public void RemoveAt(
    int index
)

Parameters

index Type: System.Int32
The zero-based index of the item to remove.

Implements

IList.RemoveAt(Int32)

Reference

TypeAttributeCollection Class [2418]
TwinCAT.TypeSystem Namespace [1622]

6.11.134.3.1 TypeAttributeCollection.TryGetAttribute Method

Tries to get the specified ITypeAttribute [2209]

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public bool TryGetAttribute(
    string name,
    out ITypeAttribute att
)

Parameters

name Type: System.String
The name of the ITypeAttribute [2209].

att Type: TwinCAT.TypeSystem.ITypeAttribute [2209].
The att.

Return Value

Type: Boolean
true if found, false otherwise.
Implements

ITypeAttributeCollection.TryGetAttribute(String, ITypeAttribute) [2215]

Reference

TypeAttributeCollection Class [2418]
TwinCAT.TypeSystem Namespace [1622]

6.11.134.3.1 TypeAttributeCollection.TryGetValue Method

Tries to get the specified Attribute value.

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public bool TryGetValue(
    string name,
    out string value
)

Parameters

name Type: System.String
    The name.
value Type: System.String
    The value.

Return Value

Type: Boolean
    true if XXXX, false otherwise.

Implements

ITypeAttributeCollection.TryGetValue(String, String) [2215]

Reference

TypeAttributeCollection Class [2418]
TwinCAT.TypeSystem Namespace [1622]

6.11.135 ValueChangedEventArgs Class

Event args for the RawValueChanged [2254] event.
Inheritance Hierarchy

```
System.Object
 System.EventArgs
 TwinCAT.TypeSystem.ValueChangedEventArgs
 TwinCAT.TypeSystem.RawValueChangedEventArgs
 TwinCAT.TypeSystem.ValueChangedBaseEventArgs
```

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

```
public class ValueChangedBaseEventArgs : EventArgs
```

The ValueChangedBaseEventArgs type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueChangedBaseEventArgs</td>
<td>Initializes a new instance of the RawValueChangedEventArgs class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime</td>
<td>Notification timestamp</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem Namespace

### 6.11.135.1 ValueChangedBaseEventArgs Constructor

Initializes a new instance of the RawValueChangedEventArgs class.
Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
protected ValueChangedBaseEventArgs(
    ISymbol symbol,
    DateTimeOffset timeStam
)
```

Parameters

- **symbol**: Type: TwinCAT.TypeSystem.ISymbol [2176]
  The symbol.
- **timeStam**: Type: System.DateTimeOffset
  The TwinCAT Real time time stamp (UTC)

Reference

ValueChangedBaseEventArgs Class [2435]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.135.2 ValueChangedBaseEventArgs Properties

The `ValueChangedBaseEventArgs` [2435] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime</td>
<td>Notification timestamp</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol.</td>
</tr>
</tbody>
</table>

Reference

ValueChangedBaseEventArgs Class [2435]
TwinCAT.TypeSystem Namespace [1622]

### 6.11.135.2.1 ValueChangedBaseEventArgs.DateTime Property

Notification timestamp

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public DateTimeOffset DateTime { get; }
```
6.11.135.2.2 ValueChangedEventArgs.Symbol Property

Gets the symbol.

**Namespace:** TwinCAT.TypeSystem

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 6.11.135.3

**Syntax**

```csharp
public ISymbol Symbol { get; }
```

**Property Value**

Type: **ISymbol**

The symbol.

**Reference**

ValueChangedBaseEventArgs Class [› 2435]

TwinCAT.TypeSystem Namespace [› 1622]

### 6.11.135.3 ValueChangedEventArgs Methods

The **ValueChangedBaseEventArgs** [› 2435] type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <strong>Object</strong>)</td>
</tr>
</tbody>
</table>

**Reference**

ValueChangedBaseEventArgs Class [› 2435]
6.11.136 ValueChangedEventArgs Class

Event args for the ValueChanged event.

Inheritance Hierarchy

System.Object
  System.EventArgs
  TwinCAT.TypeSystem.ValueChangedBaseEventArgs
  TwinCAT.TypeSystem.ValueChangedEventArgs

Namespace: TwinCAT.TypeSystem
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public class ValueChangedEventArgs : ValueChangedBaseEventArgs

The ValueChangedEventArgs type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime</td>
<td>Notification timestamp (Inherited from ValueChangedBaseEventArgs.)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Gets the symbol. (Inherited from ValueChangedBaseEventArgs.)</td>
</tr>
<tr>
<td>Value</td>
<td>The new received Value</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem Namespace [1622]
6.11.136.1 ValueChangedEventArgs Properties

The `ValueChangedEventArgs` type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime [2437]</td>
<td>Notification timestamp (Inherited from <code>ValueChangedBaseEventArgs</code> [2435].)</td>
</tr>
<tr>
<td>Symbol [2438]</td>
<td>Gets the symbol. (Inherited from <code>ValueChangedBaseEventArgs</code> [2435].)</td>
</tr>
<tr>
<td>Value [2440]</td>
<td>The new received Value</td>
</tr>
</tbody>
</table>

Reference

ValueChangedEventArgs Class [2439]

TwinCAT.TypeSystem Namespace [1622]

6.11.136.1.1 ValueChangedEventArgs.Value Property

The new received Value

Namespace: TwinCAT.TypeSystem [1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public Object Value { get; }
```

Property Value

Type: `Object`

Reference

ValueChangedEventArgs Class [2439]

TwinCAT.TypeSystem Namespace [1622]

6.11.136.2 ValueChangedEventArgs Methods

The `ValueChangedEventArgs` type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td><code>Object</code>.)</td>
</tr>
</tbody>
</table>

TwinCAT.Ads Namespaces
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Reference**

ValueChangedEventArgs Class [2439]

TwinCAT.TypeSystem Namespace [1622]

### 6.12 TwinCAT.TypeSystem.Generic Namespace

Namespace for the dynamic part of the common type system.

**Classes**

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypeCollection&lt;T&gt; [2442]</td>
<td>Data type collection</td>
</tr>
<tr>
<td>InstanceCollection&lt;T&gt; [2460]</td>
<td>Base class for Instance [2052] object collections (abstract).</td>
</tr>
<tr>
<td>NamespaceCollection&lt;T&gt; [2484]</td>
<td>Generic class for Namespace collections</td>
</tr>
<tr>
<td>ReadOnlyDataTypeCollection&lt;T&gt; [2499]</td>
<td>ReadOnly Data Type collection</td>
</tr>
<tr>
<td>ReadOnlyInstanceCollection&lt;T&gt; [2505]</td>
<td>ReadOnly Instance collection</td>
</tr>
<tr>
<td>ReadOnlyNamespaceCollection&lt;T&gt; [2505]</td>
<td>Read Only namespace collection</td>
</tr>
<tr>
<td>ReadOnlySymbolCollection&lt;T&gt; [2515]</td>
<td>Read only symbol collection</td>
</tr>
<tr>
<td>SymbolIterator&lt;T&gt; [2532]</td>
<td>Iterator class for enumerations of Symbols [2176].</td>
</tr>
</tbody>
</table>
## Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INamespace.TType.</td>
<td>Namespace interface</td>
</tr>
<tr>
<td>ISymbolProvider.TNamespace, TDataType, TSymbol.</td>
<td>Symbol provider interface</td>
</tr>
</tbody>
</table>

## Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolIterationMask.</td>
<td>Mask Flagset to specify filters for SymbolIterator.T.</td>
</tr>
</tbody>
</table>

## 6.12.1 DataTypeCollection.T. Class

Data type collection

### Inheritance Hierarchy

```
System.Object
  TwinCAT.TypeSystem.Generic.DataTypeCollection.T.
    TwinCAT.TypeSystem.DataTypeCollection. [1650]

Namespace: TwinCAT.TypeSystem.Generic. [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
```

### Syntax

C#
```
public class DataTypeCollection<T> : IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable, IDataTypeCollection<T>
where T : class, IDataType
```

### Type Parameters

T

The DataTypeCollection<T> type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypeCollection.T.</td>
<td>Initializes a new instance of the DataTypeCollection. [1650] class.</td>
</tr>
<tr>
<td>DataTypeCollection.T.IEnumerable.T.</td>
<td>Initializes a new instance of the DataTypeCollection.T. class.</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the count of contained <code>IDataType</code>'s.</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether this instance is read only.</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the <code>IDataType</code> at the specified index.</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the <code>IDataType</code> with the specified name.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds the specified item to the collection.</td>
</tr>
<tr>
<td>AddRange</td>
<td>Adds a range of types</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>Converts the <code>DataTypeCollection.T</code> into a <code>ReadOnlyCollection.T</code>.</td>
</tr>
<tr>
<td>Clear</td>
<td>Clears the collection.</td>
</tr>
<tr>
<td>Clone</td>
<td>Clones this instance.</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether this <code>DataTypeCollection</code> contains the specified <code>IDataType</code>.</td>
</tr>
<tr>
<td>ContainsType</td>
<td>Determines whether the container contains the specified <code>IDataType</code>.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the data types to the specified array, starting at the array index.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Gets the enumerator.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the Index of the specified <code>IDataType</code>.</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an <code>IDataType</code> into the <code>DataTypeCollection</code>.</td>
</tr>
<tr>
<td>LookupType</td>
<td>Determines the specified <code>IDataType</code>.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the specified <code>IDataType</code>.</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the <code>IDataType</code> object at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td>TryGetType</td>
<td>Tries to get the specified <code>IDataType</code> from the <code>IDataTypeCollection.T</code>.</td>
</tr>
</tbody>
</table>
TwinCAT.Ads Namespaces

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list [2458]</td>
<td>Internal list of data types</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem.Generic Namespace [2441]

### 6.12.1.1  DataTypeCollection.T. Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static constructor</td>
<td>Initializes a new instance of the <strong>DataTypeCollection</strong> [1650] class.</td>
</tr>
<tr>
<td>public constructor</td>
<td>Initializes a new instance of the <strong>DataTypeCollection.T</strong> [2442] class.</td>
</tr>
</tbody>
</table>

#### Reference

DataTypeCollection.T. Class [2442]

TwinCAT.TypeSystem.Generic Namespace [2441]

### 6.12.1.1.1  DataTypeCollection.T. Constructor

Initializes a new instance of the **DataTypeCollection** [1650] class.

**Namespace**: TwinCAT.TypeSystem.Generic [2441]

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.9bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public DataTypeCollection()
```

#### Reference

DataTypeCollection.T. Class [2442]

DataTypeCollection.T. Overload [2444]

TwinCAT.TypeSystem.Generic Namespace [2441]

### 6.12.1.1.2  DataTypeCollection.T. Constructor (IEnumerable.T.)

Initializes a new instance of the **DataTypeCollection.T** [2442] class.
Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public DataTypeCollection(
    IEnumerable<T> types
)
```

Parameters


Reference

DataTypeCollection.T. Class [2442]
DataTypeCollection.T. Overload [2444]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.1.2 DataTypeCollection.T. Properties

The DataTypeCollection.T. [2442] generic type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [2445]</td>
<td>Gets the count of contained IDataType [1986]s.</td>
</tr>
<tr>
<td>IsReadOnly [2446]</td>
<td>Gets a value indicating whether this instance is read only.</td>
</tr>
<tr>
<td>Item.Int32. [2447]</td>
<td>Gets or sets the IDataType [1986] at the specified index.</td>
</tr>
<tr>
<td>Item.String. [2448]</td>
<td>Gets the IDataType [1986] with the specified name.</td>
</tr>
</tbody>
</table>

Reference

DataTypeCollection.T. Class [2442]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.1.2.1 DataTypeCollection.T..Count Property

Gets the count of contained IDataType [1986]s.

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
## Syntax

### C#

```csharp
public int Count { get; }
```

### Property Value

Type: `Int32`

The count.

**Implements**

`ICollection<T>.Count`

### Reference

- `DataTypeCollection<T>.Class` [2442]
- `TwinCAT.TypeSystem.Generic Namespace` [2441]

### 6.12.1.2.2 DataTypeCollection<T>.IsReadOnly Property

Gets a value indicating whether this instance is read only.

**Namespace:** `TwinCAT.TypeSystem.Generic` [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

```csharp
public bool IsReadOnly { get; }
```

### Property Value

Type: `Boolean`

true if this instance is read only; otherwise, false.

**Implements**

`ICollection<T>.IsReadOnly`

### Reference

- `DataTypeCollection<T>.Class` [2442]
- `TwinCAT.TypeSystem.Generic Namespace` [2441]

### 6.12.1.2.3 DataTypeCollection<T>.Item Property

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item&lt;Int32&gt;</td>
<td>Gets or sets the <code>IDataType</code> at the specified index.</td>
</tr>
</tbody>
</table>
**Item.String**

**Description**

Gets the IDataType with the specified name.

---

**Reference**

DataTypeCollection.T. Class [2442]

TwinCAT.TypeSystem.Generic Namespace [2441]

---

**DataTypeCollection.T..Item Property (Int32)**

Gets or sets the IDataType at the specified index.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:

5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```
public T this[int index] { get; set; }
```

**Parameters**

- `index` Type: System.Int32
  
  The index.

**Return Value**

Type: T [2442]

T.

**Implements**

IList.T..Item.Int32.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

DataTypeCollection.T. Class [2442]

Item Overload [2446]

TwinCAT.TypeSystem.Generic Namespace [2441]
**DataTypeCollection.T..Item Property (String)**

Gets the `IDataType` with the specified name.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public T this[string name] { get; }
```

**Parameters**

- **name**
  - Type: `System.String`
  - The name.

**Return Value**

- **Type:** `T` ([2442])
- `T`

**Implements**

`IDataTypeCollection.T..Item.String` ([1996])

---

**Reference**

- [DataTypeCollection.T. Class ([2442])](#)
- [Item Overload ([2446])](#)
- [TwinCAT.TypeSystem.Generic Namespace ([2441])](#)

---

### 6.12.1.3 **DataTypeCollection.T. Methods**

The `DataTypeCollection.T` generic type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add [2449]</td>
<td>Adds the specified item to the collection.</td>
</tr>
<tr>
<td>AddRange [2450]</td>
<td>Adds a range of types</td>
</tr>
<tr>
<td>AsReadOnly [2450]</td>
<td>Converts the <code>DataTypeCollection.T</code> into a `ReadOnlyCollection.T.</td>
</tr>
<tr>
<td>Clear [2451]</td>
<td>Clears the collection.</td>
</tr>
<tr>
<td>Clone [2451]</td>
<td>Clones this instance.</td>
</tr>
<tr>
<td>Contains [2451]</td>
<td>Determines whether this <code>DataTypeCollection</code> contains the specified <code>IDataType</code> [1986].</td>
</tr>
<tr>
<td>ContainsType [2452]</td>
<td>Determines whether the container contains the specified <code>IDataType</code> [1986].</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the data types to the specified array, starting at the array index.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Gets the enumerator.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the Index of the specified <code>IDataType</code>.</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an <code>IDataType</code> into the <code>DataTypeCollection</code>.</td>
</tr>
<tr>
<td>LookupType</td>
<td>Determines the specified <code>IDataType</code>.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the specified <code>IDataType</code>.</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the <code>IDataType</code> object at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>TryGetType</td>
<td>Tries to get the specified <code>IDataType</code> from the <code>DataTypeCollection</code>.</td>
</tr>
</tbody>
</table>

**Reference**

`DataTypeCollection.T. Class` [2442]

`TwinCAT.TypeSystem.Generic Namespace` [2441]

### 6.12.1.3.1 `DataTypeCollection.T..Add Method`

Adds the specified item to the collection.

**Namespace:** `TwinCAT.TypeSystem.Generic` [2441]

**Assembly:** `TwinCAT.Ads (in TwinCAT.Ads.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void Add(
    T item
)
```

**Parameters**

item Type: `T` [2442]

The item.
Implements

ICollection<T>.Add(T)

Reference

DataTypeCollection<T>.Class [2442]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.1.3.2  DataTypeCollection<T>.AddRange Method

Adds a range of types

Namespace:  TwinCAT.TypeSystem.Generic [2441]
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public void AddRange(
  IEnumerable<T> types
)

Parameters

types Type: System.Collections.Generic.IEnumerable<T> [2442].
The types.

Reference

DataTypeCollection<T>.Class [2442]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.1.3.3  DataTypeCollection<T>.AsReadOnly Method

Converts the DataTypeCollection<T> [2442] into a ReadOnlyCollection<T>.

Namespace:  TwinCAT.TypeSystem.Generic [2441]
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public ReadOnlyDataTypeCollection<T> AsReadOnly()

Return Value

Type: ReadOnlyDataTypeCollection<T> [2499].
ReadOnlyDataTypeCollection<T> [2442].

Reference

DataTypeCollection<T>.Class [2442]
6.12.1.3.4  **DataTypeCollection.T..Clear Method**

Clears the collection.

**Namespace:**  TwinCAT.TypeSystem.Generic [➤ 2441]
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public void Clear()
```

**Implements**


**Reference**

DataTypeCollection.T. Class [➤ 2442]
TwinCAT.TypeSystem.Generic Namespace [➤ 2441]

6.12.1.3.5  **DataTypeCollection.T..Clone Method**

Clones this instance.

**Namespace:**  TwinCAT.TypeSystem.Generic [➤ 2441]
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public DataTypeCollection<T> Clone()
```

**Return Value**

Type:  DataTypeCollection [➤ 2442] T [➤ 2442].
DataTypeCollection<T>.

**Reference**

DataTypeCollection.T. Class [➤ 2442]
TwinCAT.TypeSystem.Generic Namespace [➤ 2441]

6.12.1.3.6  **DataTypeCollection.T..Contains Method**

Determines whether this DataTypeCollection [➤ 1650] contains the specified IDataType [➤ 1986].

**Namespace:**  TwinCAT.TypeSystem.Generic [➤ 2441]
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
Syntax

**C#**

```csharp
public bool Contains(
    T item
)
```

**Parameters**

- **item**
  Type: **T**
  The item.

**Return Value**

Type: **Boolean**
true if [contains] [the specified item]; otherwise, false.

**Implements**

`ICollection<T>.Contains(T)`

**Reference**

`DataTypeCollection.T. Class [2442]`

`TwinCAT.TypeSystem.Generic Namespace [2441]`

6.12.1.3.7 **DataTypeCollection.T..ContainsType Method**

Determines whether the container contains the specified `IDataType [1986]`.

**Namespace: TwinCAT.TypeSystem.Generic [2441]**

**Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14**

Syntax

**C#**

```csharp
public bool ContainsType(
    string name
)
```

**Parameters**

- **name**
  Type: **System.String**
  The name.

**Return Value**

Type: **Boolean**
true if contained; otherwise, false.

**Implements**

`IDataTypeCollection.T..ContainsType(String) [1997]`
6.12.1.3.8 **DataTypeCollection.T..CopyTo Method**

Copies the data types to the specified array, starting at the array index.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public void CopyTo(
    T[] array,
    int arrayIndex
)
```

**Parameters**

- `array` : Type: `T` [2442]. The array.

**Implements**

`ICollection<T>.CopyTo(T, Int32)`

6.12.1.3.9 **DataTypeCollection.T..GetEnumerator Method**

Gets the enumerator.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public IEnumerator<T> GetEnumerator()
```

**Return Value**

Type: `IEnumerator<T>` [2442]. A `IEnumerator<T>` that can be used to iterate through the collection.
6.12.1.3.10  **DataTypeCollection.T..IndexOf Method**

Determines the Index of the specified IDataType.

**Namespace**: TwinCAT.TypeSystem.Generic

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  

```csharp
public int IndexOf(
    T item
)
```

**Parameters**

- **item**: Type: T

**Return Value**

Type: Int32
The index of item if found in the list; otherwise, -1.

**Implements**

- IList.T..IndexOf(T)

**Reference**

- DataTypeCollection.T..Class
- TwinCAT.TypeSystem.Generic Namespace

6.12.1.3.11  **DataTypeCollection.T..Insert Method**

Inserts anIDataType into the DataTypeCollection.

**Namespace**: TwinCAT.TypeSystem.Generic

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Reference**

- DataTypeCollection.T..Class
- TwinCAT.TypeSystem.Generic Namespace
Syntax

**C#**

```csharp
public void Insert(
    int index,
    T item
)
```

**Parameters**

- **index**
  - Type: `System.Int32`
  - The index.

- **item**
  - Type: `T` [2442]
  - The item.

**Implements**

`IList<T>.Insert(Int32, T)`

**Reference**

- `DataTypeCollection.T. Class` [2442]
- `TwinCAT.TypeSystem.Generic Namespace` [2441]

### 6.12.1.3.12 `DataTypeCollection.T..LookupType Method`

Determines the specified `IDataType` [1986]

**Namespace:** `TwinCAT.TypeSystem.Generic` [2441]

**Assembly:** `TwinCAT.Ads (in TwinCAT.Ads.dll)` Version: 5.0.294+Branch.releases-5.0Sha.90bb9a1b43b6095934fddca3e72bc0e15da1c14

**Syntax**

**C#**

```csharp
public T LookupType(
    string name
)
```

**Parameters**

- **name**
  - Type: `System.String`
  - The name.

**Return Value**

- Type: `T` [2442]
  - The `IDataType` [1986] if found, otherwise NULL

**Reference**

- `DataTypeCollection.T. Class` [2442]
- `TwinCAT.TypeSystem.Generic Namespace` [2441]
### 6.12.1.3.13  DataTypeCollection.T..Remove Method

Removes the specified IDataType.  

**Namespace:** TwinCAT.TypeSystem.Generic  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public bool Remove(T item)
```

#### Parameters

- **item**  
  Type: T  
  The item.

#### Return Value

Type: Boolean  
true if item was successfully removed from the ICollection.T.; otherwise, false. This method also returns false if item is not found in the original ICollection.T.

#### Implements

ICollection.T..Remove(T)

#### Reference

DataTypeCollection.T..Class [2442]  
TwinCAT.TypeSystem.Generic Namespace [2441]

### 6.12.1.3.14  DataTypeCollection.T..RemoveAt Method

Removes the IDataType object at the specified index.  

**Namespace:** TwinCAT.TypeSystem.Generic  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public void RemoveAt(int index)
```

#### Parameters

- **index**  
  Type: System.Int32  
  The index.
6.12.1.3.15   **DataTypeCollection.T..TryGetType Method**

Tries to get the specified `IDataType` from the `IDataTypeCollection.T.`.

**Namespace:**   TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool TryGetType(
    string name,
    out T type
)
```

**Parameters**

- **name**
  - Type: `System.String`
  - The name.

- **type**
  - Type: `T`.
  - The type (Out parameter)

**Return Value**

- Type: `Boolean`
  - true if found

**Implements**

`IDataTypeCollection.T..TryGetType(String, T.)` [1998]

**Reference**

`DataTypeCollection.T. Class` [2442]

`TwinCAT.TypeSystem.Generic Namespace` [2441]

6.12.1.4   **DataTypeCollection.T. Fields**

The `DataTypeCollection.T.` generic type exposes the following members.
Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list [2458]</td>
<td>Internal list of data types</td>
</tr>
</tbody>
</table>

Reference

DataTypeCollection.T..Class [2442]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.1.4.1   DataTypeCollection.T..list Field

Internal list of data types

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected List<T> list

Field Value

Type: List<T> [2442].

Reference

DataTypeCollection.T..Class [2442]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.2   INamespace..TType..Interface

Namespace interface

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public interface INamespace<TType>
where TType : class, IDataType

Type Parameters

TType          DataType class used within this Namespace interface

The INamespace..TType..type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypes</td>
<td>Data types organized by the INamespace.TType.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name/identifier of the Namespace</td>
</tr>
</tbody>
</table>

Reference

INamespace.TType. Interface [2458]

TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.2.1 INamespace.TType. Properties

The INamespace.TType. [2458] generic type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypes</td>
<td>Data types organized by the INamespace.TType.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name/identifier of the Namespace</td>
</tr>
</tbody>
</table>

Reference

INamespace.TType. Interface [2458]

TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.2.1.1 INamespace.TType..DataTypes Property

Data types organized by the INamespace.TType. [2458]

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
IDataTypeCollection<TType> DataTypes { get; }

Property Value

Type: IDataTypeCollection[TType].1995, TType.2458.
The data types.

Reference

INamespace.TType. Interface [2458]
TwinCAT.TypeSystem.Generic Namespace [2441]
6.12.2.1.2  **INamespace.TType..Name Property**

Gets the name/identifier of the Namespace

**Namespace:**  TwinCAT.TypeSystem.Generic  [›  2441]
**Assembly:**  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
string Name { get; }
```

**Property Value**

Type: String
The name.

**Reference**

INamespace.TType. Interface  [›  2458]
TwinCAT.TypeSystem.Generic Namespace  [›  2441]

### 6.12.3  InstanceCollection.T. Class

Base class for **Instance**  [›  2052] object collections (abstract).

**Inheritance Hierarchy**

System.Object
  TwinCAT.TypeSystem.Generic_InstanceCollection.T.
    TwinCAT.TypeSystem.FieldCollection  [›  1945]
    TwinCAT.TypeSystem.Generic.SymbolCollection.T.  [›  2526]
    TwinCAT.TypeSystem.MemberCollection  [›  2281]
**Namespace:**  TwinCAT.TypeSystem.Generic  [›  2441]
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public abstract class InstanceCollection<T> : IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable, IInstanceCollection<T>
where T : class, IInstance
```

**Type Parameters**

T

The InstanceCollection.T. type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceCollection.T (InstanceCollection Mode)</td>
<td>Initializes a new instance of the InstanceCollection.T. class.</td>
</tr>
</tbody>
</table>

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the collection count.</td>
</tr>
<tr>
<td>InnerList</td>
<td>Gets the List of instances.</td>
</tr>
<tr>
<td>InnerPathDict</td>
<td>The Path dictionary</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether this instance is read only.</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets or sets the IInstance at the specified index.</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the IInstance with the specified instance path.</td>
</tr>
<tr>
<td>Mode</td>
<td>The mode this InstanceCollection.T. is working in.</td>
</tr>
</tbody>
</table>

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds the specified item.</td>
</tr>
<tr>
<td>AddRange</td>
<td>Adds the specified items to this collection.</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>Converts the InstanceCollection.T. to an ReadOnlyInstanceCollection.T.</td>
</tr>
<tr>
<td>Clear</td>
<td>Clears this instance.</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether this collection contains an IInstance with the specified InstanceName / InstancePath</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether this collection contains the specified IInstance.</td>
</tr>
<tr>
<td>ContainsName</td>
<td>Determines whether the specified instance name contains name.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies this InstanceCollection.T. to the specified array.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetEnumerator</td>
<td>Gets the enumerator.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetInstance [2475]</td>
<td>Gets the IInstance [2052] by instance path.</td>
</tr>
<tr>
<td>GetInstanceByName [2476]</td>
<td>Gets the name of the instance by.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf [2476]</td>
<td>Determines the index of the specified IInstance [2052].</td>
</tr>
<tr>
<td>Insert [2477]</td>
<td>Inserts the specified IInstance [2052] at the specified index.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove [2478]</td>
<td>Removes the specified item.</td>
</tr>
<tr>
<td>RemoveAt [2479]</td>
<td>Removes the IInstance [2052] at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetInstance [2479]</td>
<td>Tries to get the IInstance [2052] of the specified path.</td>
</tr>
<tr>
<td>TryGetInstanceByName [2480]</td>
<td>Tries to get Instances by name.</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem.Generic Namespace [2441]

### 6.12.3.1 InstanceCollection.T. Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(InstanceCollectionMode) [2463]</td>
<td></td>
</tr>
<tr>
<td>InstanceCollectionMode) [2463]</td>
<td></td>
</tr>
</tbody>
</table>

### Reference

InstanceCollection.T. Class [2460]

TwinCAT.TypeSystem.Generic Namespace [2441]
6.12.3.1.1 InstanceCollection.T. Constructor (InstanceCollectionMode)

Initializes a new instance of the InstanceCollection.T. class.

**Namespace:** TwinCAT>TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected InstanceCollection(
    InstanceCollectionMode mode
)
```

**Parameters**

- `mode`  
  Type: TwinCAT>TypeSystem>InstanceCollectionMode
  
  The mode.

**Reference**

- InstanceCollection.T.Class
- InstanceCollection.T.Overload
- TwinCAT>TypeSystem.Generic Namespace

6.12.3.1.2 InstanceCollection.T. Constructor (IEnumerable<T>, InstanceCollectionMode)

Initializes a new instance of the InstanceCollection.T. class.

**Namespace:** TwinCAT>TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
protected InstanceCollection(
    IEnumerable<T> coll,
    InstanceCollectionMode mode
)
```

**Parameters**

- `coll`  
  Type: System.Collections.Generic(IEnumerable<T>)
  
  The copy collection

- `mode`  
  Type: TwinCAT>TypeSystem>InstanceCollectionMode
  
  The mode.

**Reference**

- InstanceCollection.T.Class
- InstanceCollection.T.Overload
- TwinCAT>TypeSystem.Generic Namespace
6.12.3.2 InstanceCollection.T. Properties

The InstanceCollection.T. [2460] generic type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [2464]</td>
<td>Gets the collection count.</td>
</tr>
<tr>
<td>InnerList [2465]</td>
<td>Gets the List of instances.</td>
</tr>
<tr>
<td>InnerPathDict [2465]</td>
<td>The Path dictionary</td>
</tr>
<tr>
<td>IsReadOnly [2465]</td>
<td>Gets a value indicating whether this instance is read only.</td>
</tr>
<tr>
<td>Item.Int32. [2466]</td>
<td>Gets or sets the IInstance [2052] at the specified index.</td>
</tr>
<tr>
<td>Item.String. [2467]</td>
<td>Gets the IInstance [2052] with the specified instance path.</td>
</tr>
<tr>
<td>Mode [2468]</td>
<td>The mode this InstanceCollection.T. [2460] is working in.</td>
</tr>
</tbody>
</table>

**Reference**

InstanceCollection.T. Class [2460]

TwinCAT.TypeSystem.Generic Namespace [2441]

### 6.12.3.2.1 InstanceCollection.T..Count Property

Gets the collection count.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public int Count { get; }
```

**Property Value**

- **Type:** Int32
- The count.

**Implements**

ICollection.T..Count

**Reference**

InstanceCollection.T. Class [2460]

TwinCAT.TypeSystem.Generic Namespace [2441]
6.12.3.2.2 InstanceCollection.T..InnerList Property

 Gets the list of instances.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
protected IList<T> InnerList { get; }
```

**Property Value**

Type: `IList<T>` [2460].
The inner list.

**Reference**

InstanceCollection.T.. Class [2460]

TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.3.2.3 InstanceCollection.T..InnerPathDict Property

The path dictionary

**Namespace:** TwinCAT.TypeSystem.Generic [2441]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
protected IDictionary<string, T> InnerPathDict { get; }
```

**Property Value**

Type: `IDictionary<string, T>` [2460].

**Reference**

InstanceCollection.T.. Class [2460]

TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.3.2.4 InstanceCollection.T..IsReadOnly Property

Gets a value indicating whether this instance is read only.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public bool IsReadOnly { get; };
```

Property Value

Type: Boolean
true if this instance is read only; otherwise, false.

Implements

ICollection.T..IsReadOnly

Reference

InstanceCollection.T. Class [2460]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.3.2.5 InstanceCollection.T..Item Property

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32.</td>
<td>Gets or sets the</td>
</tr>
<tr>
<td>Item.String.</td>
<td>Gets the</td>
</tr>
</tbody>
</table>

Reference

InstanceCollection.T. Class [2460]
TwinCAT.TypeSystem.Generic Namespace [2441]

InstanceCollection.T..Item Property (Int32)

Gets or sets the |Instance| at the specified index.

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public T this[int index] { get; set; }
```

Parameters

index Type: System.Int32
The index.
Return Value

Type: T [2460]
T.

Implements

IList<T>.Item.Int32.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

InstanceCollection.T..Class [2460]

Item Overload [2466]

TwinCAT.TypeSystem.Generic Namespace [2441]

**InstanceCollection.T..Item Property (String)**

Gets the IInstance [2052] with the specified instance path.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public T this[string instanceSpecifier] { get; }
```

Parameters

- `instanceSpecifier` Type: System.String
  The instance path or Instance Name (dependent of Mode [2468] setting)

Return Value

Type: T [2460]
T.

Implements

IInstanceCollection.T..Item.String. [2059]
**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td></td>
</tr>
<tr>
<td>ArgumentException</td>
<td></td>
</tr>
</tbody>
</table>

**Remarks**

Dependent what this `InstanceCollection.T. [2460]` contains configured by the `InstanceCollectionMode [2075]` the instance specifier should be the `InstanceName [2054]` or the `InstancePath [2055]`.

**Reference**

- `InstanceCollection.T. Class [2460]`
- `Item Overload [2466]`
- `TwinCAT.TypeSystem.Generic Namespace [2441]`

**6.12.3.2.6 InstanceCollection.T..Mode Property**

The mode this `InstanceCollection.T. [2460]` is working in.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public InstanceCollectionMode Mode { get; }
```

**Property Value**

Type: `InstanceCollectionMode [2075]`

**Implements**

`IInstanceCollection.T..Mode [2060]`

**Reference**

- `InstanceCollection.T. Class [2460]`
- `TwinCAT.TypeSystem.Generic Namespace [2441]`

**6.12.3.3 InstanceCollection.T. Methods**

The `InstanceCollection.T. [2460]` generic type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Add] [2469]</td>
<td>Adds the specified item.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AddRange {2470}</td>
<td>Adds the specified items to this collection.</td>
</tr>
<tr>
<td>AsReadOnly {2470}</td>
<td>Converts the InstanceCollection.T. {2460} to an ReadOnlyInstanceCollection.T. {2505}</td>
</tr>
<tr>
<td>Clear {2471}</td>
<td>Clears this instance.</td>
</tr>
<tr>
<td>Contains(String) {2472}</td>
<td>Determines whether this collection contains an Instance {2052} with the specified InstanceName / InstancePath</td>
</tr>
<tr>
<td>Contains(T) {2472}</td>
<td>Determines whether this collection contains the specified Instance {2052}</td>
</tr>
<tr>
<td>ContainsName {2473}</td>
<td>Determines whether the specified instance name contains name.</td>
</tr>
<tr>
<td>CopyTo {2474}</td>
<td>Copies this InstanceCollection.T. {2460} to the specified array.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator {2474}</td>
<td>Gets the enumerator.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetInstance {2475}</td>
<td>Gets the Instance {2052} by instance path.</td>
</tr>
<tr>
<td>GetInstanceByName {2476}</td>
<td>Gets the name of the instance by.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf {2476}</td>
<td>Determines the index of the specified Instance {2052}.</td>
</tr>
<tr>
<td>Insert {2477}</td>
<td>Inserts the specified Instance {2052} at the specified index.</td>
</tr>
<tr>
<td>MemberwiseClone {2478}</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove {2478}</td>
<td>Removes the specified item.</td>
</tr>
<tr>
<td>RemoveAt {2479}</td>
<td>Removes the Instance {2052} at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetInstance {2479}</td>
<td>Tries to get the Instance {2052}. of the specified path.</td>
</tr>
<tr>
<td>TryGetInstanceByName {2480}</td>
<td>Tries to get Instances by name.</td>
</tr>
</tbody>
</table>

**Reference**

InstanceCollection.T.Class {2460}

TwinCAT.TypeSystem.Generic Namespace {2441}

6.12.3.3.1 InstanceCollection.T..Add Method

Adds the specified item.
Syntax

C#

```csharp
public void Add(
    T item
)
```

Parameters

item Type: `T`[2460]
The item.

Implements

`ICollection<T>.Add(T)`

Reference

InstanceCollection.T. Class [2460]

`TwinCAT.TypeSystem.Generic Namespace [2441]`

6.12.3.3.2 InstanceCollection.T..AddRange Method

Adds the specified items to this collection.

Namespace: `TwinCAT.TypeSystem.Generic [2441]`
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void AddRange(
    IEnumerable<T> items
)
```

Parameters

items Type: `System.Collections.Generic.IEnumerable<T>`[2460],
The items.

Reference

InstanceCollection.T. Class [2460]

`TwinCAT.TypeSystem.Generic Namespace [2441]`

6.12.3.3.3 InstanceCollection.T..AsReadOnly Method

Converts the `InstanceCollection.T. [2460]` to an `ReadOnlyInstanceCollection.T. [2505]`
## TwinCAT.Ads Namespaces

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public ReadOnlyInstanceCollection<T> AsReadOnly()
```

**Return Value**

Type: `ReadOnlyInstanceCollection<T>`.

### Reference

- `InstanceCollection<T>.Class` [2460]
- `TwinCAT.TypeSystem.Generic Namespace` [2441]

### 6.12.3.3.4 InstanceCollection<T>.Clear Method

Clears this instance.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void Clear()
```

**Implements**

- `ICollection<T>.Clear`

### Reference

- `InstanceCollection<T>.Class` [2460]
- `TwinCAT.TypeSystem.Generic Namespace` [2441]

### 6.12.3.3.5 InstanceCollection<T>.Contains Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String) [2472]</td>
<td>Determines whether this collection contains an <code>IInstance</code> [2052] with the specified InstanceName / InstancePath</td>
</tr>
<tr>
<td>Contains(T) [2472]</td>
<td>Determines whether this collection contains the specified <code>IInstance</code> [2052]</td>
</tr>
</tbody>
</table>
InstanceCollection.T..Contains Method (String)

Determines whether this collection contains an IInstance with the specified InstanceName / InstancePath.

**Namespace:**  TwinCAT.TypeSystem.Generic
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool Contains(
    string instanceSpecifier
)
```

**Parameters**

instanceSpecifier  
Type: System.String  
The instance path or Instance Name (dependent of Mode setting)

**Return Value**

Type: Boolean  
true if [contains] [the specified instance path]; otherwise, false.

**Implements**

IInstanceCollection.T..Contains(String)  

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>instancePath</td>
</tr>
<tr>
<td>ArgumentException</td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

InstanceCollection.T..Class  
Contains Overload  
TwinCAT.TypeSystem.Generic

---

**InstanceCollection.T..Contains Method (T)**

Determines whether this collection contains the specified IInstance.

**Reference**

InstanceCollection.T..Class  
Contains Overload  
TwinCAT.TypeSystem.Generic

### Contains Method

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool Contains(T item)
```

#### Parameters

- **item**
  - Type: `T`  
  - The item.

#### Return Value

- Type: `Boolean`
  - true if contains [the specified item]; otherwise, false.

**Implements**

- `ICollection<T>.Contains(T)`

**Reference**

- `InstanceCollection.T. Class`  
- `Contains Overload`  
- TwinCAT.TypeSystem.Generic Namespace

### Contains Name Method

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool ContainsName(string instanceName)
```

#### Parameters

- **instanceName**
  - Type: `System.String`
  - Name of the instance.

#### Return Value

- Type: `Boolean`
  - true if the specified instance name contains name; otherwise, false.
Implements

IInstanceCollection.T.ContainsName(String) [2062]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

InstanceCollection.T.Class [2460]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.3.3.7 InstanceCollection.T.CopyTo Method

Copies this InstanceCollection.T. [2460] to the specified array.

Namespace:  TwinCAT.TypeSystem.Generic [2441]
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void CopyTo(
    T[] array,
    int arrayIndex
)
```

Parameters

array  Type: .T. [2460].
The array.
arrayIndex  Type: System.Int32
Index of the array.

Implements

ICollection.T.CopyTo(T, Int32)

Reference

InstanceCollection.T.Class [2460]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.3.3.8 InstanceCollection.T.GetEnumerator Method

Gets the enumerator.
Namespace: TwinCAT.TypeSystem.Generic
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public IEnumerator<T> GetEnumerator()
```

Return Value

Type: `IEnumerator<T>`
A `IEnumerator<T>` that can be used to iterate through the collection.

Implements

`IEnumerable<T>.GetEnumerator`

Reference

InstanceCollection.T.Class
TwinCAT.TypeSystem.Generic Namespace

6.12.3.3.9 InstanceCollection.T..GetInstance Method

Gets the `IInstance` by instance path.

Namespace: TwinCAT.TypeSystem.Generic
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public T GetInstance(
    string instanceSpecifier
)
```

Parameters

`instanceSpecifier` Type: `System.String`
The instance path or Instance Name (dependent of `Mode` setting)

Return Value

Type: `T`

Implements

`IInstanceCollection<T>.GetInstance(String)`
Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>Path not found!;instancePath</td>
</tr>
</tbody>
</table>

Reference

**InstanceCollection.T..Class** [2460]

**TwinCAT.TypeSystem.Generic Namespace** [2441]

### 6.12.3.3.10 InstanceCollection.T..GetInstanceByName Method

Gets the name of the instance by.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public IList<T> GetInstanceByName(
    string instanceName
)
```

**Parameters**

- instanceName
  - Type: System.String
  - Name of the instance.

**Return Value**

Type: **IList<T>** [2460].

**Implements**

IInstanceCollection.T..GetInstanceByName(String) [2063]

**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>Name not found!;instanceName</td>
</tr>
</tbody>
</table>

Reference

**InstanceCollection.T..Class** [2460]

**TwinCAT.TypeSystem.Generic Namespace** [2441]

### 6.12.3.3.11 InstanceCollection.T..IndexOf Method

Determines the index of the specified **IInstance** [2052].
**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public int IndexOf(
    T item
)
```

**Parameters**

- **item**
  - Type: `T` [2460]
  - The item.

**Return Value**

Type: `Int32`
The index of `item` if found in the list; otherwise, -1.

**Implements**

`IList<T>.IndexOf(T)`

**Reference**

`InstanceCollection<T>.Class` [2460]

TwinCAT.TypeSystem.Generic Namespace [2441]

---

**6.12.3.3.12 InstanceCollection<T>.Insert Method**

Inserts the specified `IInstance` [2052] at the specified index.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public void Insert(
    int index,
    T instance
)
```

**Parameters**

- **index**
  - Type: `System.Int32`
  - The instance.

- **instance**
  - Type: `T` [2460]
  - The item.
Implements

IList<T>.Insert(Int32, T)

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>index or index</td>
</tr>
<tr>
<td>ArgumentNullException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

InstanceCollection<T>.Class [2460]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.3.3.13 InstanceCollection<T>.Remove Method

Removes the specified item.

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public bool Remove(T item)

Parameters

item Type: T [2460]
The item.

Return Value

Type: Boolean
ture if item was successfully removed from the ICollection<T>; otherwise, false. This method also returns false if item is not found in the original ICollection<T>.

Implements

ICollection<T>.Remove(T)

Reference

InstanceCollection<T>.Class [2460]
TwinCAT.TypeSystem.Generic Namespace [2441]
6.12.3.3.14 InstanceCollection.T..RemoveAt Method

Removes the IInstance [2052] at the specified index.

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public void RemoveAt(
    int index
)

Parameters

index Type: System.Int32
The index.

Implements

IList.T..RemoveAt(Int32)

Reference

InstanceCollection.T. Class [2460]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.3.3.15 InstanceCollection.T..TryGetInstance Method

Tries to get the IInstance [2052]. of the specified path.

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public bool TryGetInstance(
    string instanceSpecifier,
    out T symbol
)

Parameters

instanceSpecifier Type: System.String
The instance path or Instance Name (dependent of Mode [2468] setting)
symbol Type: T [2460].
The symbol.

Return Value

Type: Boolean
ture if the IInstance [2052] is found; otherwise, false
Implements

IInstanceCollection.T.TryGetInstance(String, T) [2063]

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>instancePath</td>
</tr>
<tr>
<td>ArgumentException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

InstanceCollection.T.Class [2460]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.3.3.16 InstanceCollection.T.TryGetInstanceByName Method

Tries to get Instances by name.

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public virtual bool TryGetInstanceByName(
    string instanceName,
    out IList<T> instances
)
```

Parameters

- `instanceName` Type: System.String
  Name of the instance.

- `instances` Type: System.Collections.Generic.IList<T>
  The instances found.

Return Value

Type: Boolean
true if the IInstance [2052] is found; otherwise, false

Implements

IInstanceCollection.T.TryGetInstanceByName(String, IList<T>) [2064]

Reference

InstanceCollection.T.Class [2460]
TwinCAT.TypeSystem.Generic Namespace [2441]
6.12.4 ISymbolProvider.TNamespace, TDataType, TSymbol. Interface

Symbol provider interface

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public interface ISymbolProvider<TNamespace, TDataType, TSymbol>
    where TDataType : class, IDataType
    where TSymbol : class, ISymbol
```

**Type Parameters**

- **TNamespace**
  - Namespace type
- **TDataType**
  - DataType type
- **TSymbol**
  - Symbol type

The ISymbolProvider.TNamespace, TDataType, TSymbol. type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypes</td>
<td>Gets all data types from all Namespaces</td>
</tr>
<tr>
<td>Namespaces</td>
<td>Get the Namespaces of DataTypes for this Symbol provider</td>
</tr>
<tr>
<td>RootNamespace</td>
<td>Gets the root (main) namespace of the Symbol provider.</td>
</tr>
<tr>
<td>RootNamespaceName</td>
<td>Gets the name of the root namespace</td>
</tr>
<tr>
<td>Symbols</td>
<td>Gets the (root) symbols of the Symbol provider.</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem.Generic Namespace

6.12.4.1 ISymbolProvider.TNamespace, TDataType, TSymbol. Properties

The ISymbolProvider.TNamespace, TDataType, TSymbol. Properties generic type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataTypes</td>
<td>Gets all data types from all Namespaces</td>
</tr>
<tr>
<td>Namespaces</td>
<td>Get the Namespaces of DataTypes for this Symbol provider</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RootNamespace</td>
<td>Gets the root (main) namespace of the Symbol provider.</td>
</tr>
<tr>
<td>RootNamespaceName</td>
<td>Gets the name of the root namespace</td>
</tr>
<tr>
<td>Symbols</td>
<td>Gets the (root) symbols of the Symbol provider.</td>
</tr>
</tbody>
</table>

### Reference

-ISymbolProvider.TNamespace, TDataType, TSymbol. Interface [2481]-
-TwinCAT.TypeSystem.Generic Namespace [2441]-

### 6.12.4.1.1 ISymbolProvider.TNamespace, TDataType, TSymbol..DataTypes Property

Gets all data types from all Namespaces

- **Namespace:** TwinCAT.TypeSystem.Generic [2441]
- **Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
IDataTypeCollection<TDataType> DataTypes { get; }
```

**Property Value**

Type: [IDataTypeCollection [1995], TDataType [2481]].
The data types.

### Reference

-ISymbolProvider.TNamespace, TDataType, TSymbol. Interface [2481]-
-TwinCAT.TypeSystem.Generic Namespace [2441]-

### 6.12.4.1.2 ISymbolProvider.TNamespace, TDataType, TSymbol..Namespaces Property

Get the Namespaces of DataTypes for this Symbol provider

- **Namespace:** TwinCAT.TypeSystem.Generic [2441]
- **Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
INamespaceCollection<TDataType> Namespaces { get; }
```
Property Value
Type: INamespaceCollection[2073].TDataType[2481].
ReadOnly collection of the namespaces.

Reference
ISymbolProvider.TNamespace, TDataType, TSymbol. Interface[2481]
TwinCAT.TypeSystem.Generic Namespace[2441]

6.12.4.1.3  ISymbolProvider.TNamespace, TDataType, TSymbol..RootNamespace Property

Gets the root (main) namespace of the Symbol provider.

Namespace:  TwinCAT.TypeSystem.Generic[2441]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
INamespace<TDataType> RootNamespace { get; }

Property Value
Type: INamespace[2458].TDataType[2481].
The root namespace.

Reference
ISymbolProvider.TNamespace, TDataType, TSymbol. Interface[2481]
TwinCAT.TypeSystem.Generic Namespace[2441]

6.12.4.1.4  ISymbolProvider.TNamespace, TDataType, TSymbol..RootNamespaceName Property

Gets the name of the root namespace

Namespace:  TwinCAT.TypeSystem.Generic[2441]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
string RootNamespaceName { get; }

Property Value
Type: String
The namespace.
6.12.4.1.5   ISymbolProvider.TNamespace, TDataType, TSymbol..Symbols Property

Gets the (root) symbols of the Symbol provider.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
ISymbolCollection<TSymbol> Symbols { get; }
```

**Property Value**

Type: ISymbolCollection[TSymbol], TSymbol

Read only collection of the Symbols

**Reference**

ISymbolProvider.TNamespace, TDataType, TSymbol. Interface

TwinCAT.TypeSystem.Generic Namespace

---

6.12.5   NamespaceCollection<T>. Class

Generic class for Namespace collections

**Inheritance Hierarchy**

System.Object  
TwinCAT.TypeSystem.Generic.NamespaceCollection<T>

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public class NamespaceCollection<T> : IList<INamespace<T>>,  
   ICollection<INamespace<T>>, IEnumerable<INamespace<T>>, IEnumerable  
where T : class, IDataType
```

**Type Parameters**

T

The NamespaceCollection<T> type exposes the following members.
### TwinCAT.Ads Namespaces

#### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NamespaceCollection.T.</td>
<td>Initializes a new instance of the NamespaceCollection.T. class.</td>
</tr>
</tbody>
</table>

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllTypes</td>
<td>Gets all types included in all namespaces.</td>
</tr>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T.</td>
</tr>
<tr>
<td>InnerAllTypes Dictionary FullPath -&gt; IDataType</td>
<td></td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T. is read-only.</td>
</tr>
<tr>
<td>Item.Int32.</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item.String.</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
</tbody>
</table>

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an item to the ICollection.T.</td>
</tr>
<tr>
<td>Clear</td>
<td>Removes all items from the ICollection.T.</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines whether the ICollection.T. contains a specific value.</td>
</tr>
<tr>
<td>ContainsNamespace</td>
<td>Determines whether the specified name contains namespace.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies to.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Determines the index of a specific item in the IList.T.</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts an item to the IList.T. at the specified index.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the first occurrence of a specific object from the ICollection.T.</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the IList.T. item at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryGetNamespace</td>
<td>Tries to get the namespace object</td>
</tr>
<tr>
<td>TryGetType</td>
<td>Tries to get the specified type.</td>
</tr>
<tr>
<td>TryGetTypeByName</td>
<td>Tries to get the data type by full name.</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem.Generic Namespace [2441]

#### 6.12.5.1 NamespaceCollection.T. Constructor

Initializes a new instance of the `NamespaceCollection.T` class.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public NamespaceCollection()
```

**Reference**

NamespaceCollection.T. Class [2484]

TwinCAT.TypeSystem.Generic Namespace [2441]

#### 6.12.5.2 NamespaceCollection.T. Properties

The `NamespaceCollection.T` generic type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllTypes</td>
<td>Gets all types included in all namespaces.</td>
</tr>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ICollection.T.</td>
</tr>
<tr>
<td>InnerAllTypes</td>
<td>Dictionary FullPath -&gt; IDataType</td>
</tr>
<tr>
<td>IsReadOnly</td>
<td>Gets a value indicating whether the ICollection.T. is read-only.</td>
</tr>
<tr>
<td>Item.Int32,</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item.String,</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
</tbody>
</table>

**Reference**

NamespaceCollection.T. Class [2484]
6.12.5.2.1 NamespaceCollection.T..AllTypes Property

Gets all types included in all namespaces.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll)

Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public IDataTypeCollection<T> AllTypes { get; }
```

**Property Value**

Type: `IDataTypeCollection<T>`

All types.

**Reference**

NamespaceCollection.T..Class

TwinCAT.TypeSystem.Generic Namespace

6.12.5.2.2 NamespaceCollection.T..Count Property

Gets the number of elements contained in the `ICollection<T>`.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll)

Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public int Count { get; }
```

**Return Value**

Type: `Int32`

The number of elements contained in the `ICollection<T>`.

**Implements**

`ICollection<T.Count`

**Reference**

NamespaceCollection.T..Class

TwinCAT.TypeSystem.Generic Namespace
6.12.5.2.3 NamespaceCollection.T..InnerAllTypes Property

Dictionary FullPath -> IDataType

Namespace: TwinCAT.TypeSystem.Generic
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
protected IDictionary<string, T> InnerAllTypes { get; } // Type: IDictionary, T

Property Value

Type: IDictionary, T.

Reference

NamespaceCollection.T.. Class
TwinCAT.TypeSystem.Generic Namespace

6.12.5.2.4 NamespaceCollection.T..IsReadOnly Property

Gets a value indicating whether the ICollection is read-only.

Namespace: TwinCAT.TypeSystem.Generic
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool IsReadOnly { get; } // Type: Boolean, true if the ICollection is read-only; otherwise, false.

Reference

NamespaceCollection.T.. Class
TwinCAT.TypeSystem.Generic Namespace
6.12.5.2.5  NamespaceCollection.T..Item Property

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32. [2489]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
<tr>
<td>Item.String. [2490]</td>
<td>Gets or sets the element at the specified index.</td>
</tr>
</tbody>
</table>

Reference

NamespaceCollection.T. Class [2484]

TwinCAT.TypeSystem.Generic Namespace [2441]

NamespaceCollection.T..Item Property (Int32)

Gets or sets the element at the specified index.

Namespace:  TwinCAT.TypeSystem.Generic [2441]
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public INamespace<T> this[int index] { get; set; }
```

Parameters

index  Type: System.Int32
The index.

Return Value

Type: INamespace [2458].

Implements

IList<T>.Item.Int32.

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotImplementedException</td>
<td></td>
</tr>
</tbody>
</table>

Reference

NamespaceCollection.T. Class [2484]
Item Overload [2489]
### NamespaceCollection.T..Item Property (String)

Gets or sets the element at the specified index.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public INamespace<T> this[string str] { get; }
```

#### Parameters

- **str**
  - **Type:** System.String
  - The STR.

#### Return Value

- **Type:** INamespace<T>
  - The STR.

#### Reference

- NamespaceCollection.T. Class
- Item Overload
- TwinCAT.TypeSystem.Generic Namespace

### 6.12.5.3 NamespaceCollection.T. Methods

The NamespaceCollection.T.[2484] generic type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add [2491]</td>
<td>Adds an item to the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>Clear [2492]</td>
<td>Removes all items from the ICollection&lt;T&gt;.</td>
</tr>
<tr>
<td>Contains [2492]</td>
<td>Determines whether the ICollection&lt;T&gt; contains a specific value.</td>
</tr>
<tr>
<td>ContainsNamespace [2493]</td>
<td>Determines whether the specified name contains namespace.</td>
</tr>
<tr>
<td>CopyTo [2493]</td>
<td>Copies to.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf [2494]</td>
<td>Determines the index of a specific item in the IList.T.</td>
</tr>
<tr>
<td>Insert [2495]</td>
<td>Inserts an item to the IList.T at the specified index.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove [2496]</td>
<td>Removes the first occurrence of a specific object from the ICollection.T.</td>
</tr>
<tr>
<td>RemoveAt [2496]</td>
<td>Removes the IList.T item at the specified index.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetNamespace</td>
<td>Tries to get the namespace object</td>
</tr>
<tr>
<td>TryGetType [2497]</td>
<td>Tries to get the specified type.</td>
</tr>
<tr>
<td>TryGetTypeByName [2498]</td>
<td>Tries to get the data type by full name.</td>
</tr>
</tbody>
</table>

Reference

NamespaceCollection.T..Class [2484]

TwinCAT.TypeSystem.Generic Namespace [2441]

### 6.12.5.3.1 NamespaceCollection.T..Add Method

Adds an item to the ICollection.T.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void Add(
    INamespace<T> item
)
```

**Parameters**

item

Type: TwinCAT.TypeSystem.Generic.INamespace [2458].T [2484].
The object to add to the ICollection.T.

**Implements**

ICollection.T.Add(T)
6.12.5.3.2 NamespaceCollection.T..Clear Method

Removes all items from the ICollection.T.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# public void Clear()

**Implements**

ICollection.T.Clear.

6.12.5.3.3 NamespaceCollection.T..Contains Method

Determines whether the ICollection.T contains a specific value.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C# public bool Contains(
    INamespace<T> item
)

**Parameters**

item Type: TwinCAT.TypeSystem.Generic.INamespace

The object to locate in the ICollection.T.

**Return Value**

Type: Boolean
true if item is found in the ICollection.T; otherwise, false.
Implements

IEnumerator<T>

Reference

NamespaceCollection.T..Contains(T) Method

Determines whether the specified name contains namespace.

Namespace: TwinCAT.TypeSystem.Generic
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C# public bool ContainsNamespace(
    string name
) Parameters

name Type: System.String
The name.

Return Value

Type: Boolean
true if the specified name contains namespace; otherwise, false.

Reference

NamespaceCollection.T..CopyTo Method

Copies to.

Namespace: TwinCAT.TypeSystem.Generic
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C# public void CopyTo(
    INamespace<T>[] array,
    int arrayIndex
)
Parameters

array                   Type: TwinCAT.TypeSystem.Generic.INamespace[T].T
                        The array.
arrayIndex             Type: System.Int32
                        Index of the array.

Implements

ICollection.T.CopyTo(T, Int32)

Reference

NamespaceCollection.T..GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: TwinCAT.TypeSystem.Generic
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sh.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public IEnumerator<INamespace<T>> GetEnumerator()

Return Value

Type: IEnumerator<INamespace<T>>.T
A IEnumerator<T> that can be used to iterate through the collection.

Implements

IEnumerable.T.GetEnumerator.

Reference

NamespaceCollection.T..IndexOf Method

Determines the index of a specific item in the IList<T>.

Namespace: TwinCAT.TypeSystem.Generic
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sh.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public int IndexOf(
    INamespace<T> item
)
```

Parameters

- **item**
  - Type: `TwinCAT.TypeSystem.Generic.INamespace<T>`. The object to locate in the `IList<T>`.

Return Value

- Type: `Int32`.
  - The index of item if found in the list; otherwise, -1.

Implements

- `IList<T>.IndexOf(T)`

Reference

- `NamespaceCollection.T.Class` [2484]
- `TwinCAT.TypeSystem.Generic.Namespace` [2441]

6.12.5.3.8 NamespaceCollection.T..Insert Method

Inserts an item to the `IList<T>` at the specified index.

Namespace: `TwinCAT.TypeSystem.Generic` [2441]

Assembly: `TwinCAT.Ads (in TwinCAT.Ads.dll)` Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public void Insert(
    int index,
    INamespace<T> item
)
```

Parameters

- **index**
  - Type: `System.Int32`.
    - The zero-based index at which item should be inserted.

- **item**
  - Type: `TwinCAT.TypeSystem.Generic.INamespace<T>`.
    - The object to insert into the `IList<T>`.

Implements

- `IList<T>.Insert(Int32, T)`
6.12.5.3.9 NamespaceCollection.T..Remove Method

Removes the first occurrence of a specific object from the ICollection.T.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool Remove(
    INamespace<T> item
)
```

**Parameters**

`item` Type: TwinCAT.TypeSystem.Generic.INamespace [2458].T [2484]. The object to remove from the ICollection.T.

**Return Value**

Type: Boolean

true if item was successfully removed from the ICollection.T.; otherwise, false. This method also returns false if item is not found in the original ICollection.T.

**Implements**

ICollection.T..Remove(T)

Reference

NamespaceCollection.T. Class [2484]

TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.5.3.10 NamespaceCollection.T..RemoveAt Method

Removes the IList.T. item at the specified index.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void RemoveAt(
    int index
)
```
### NamespaceCollection.T..TryGetNamespace Method

Tries to get the namespace object

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public bool TryGetNamespace(
    string name,
    out INamespace<T> nspace
)
```

**Parameters**

- `name` Type: System.String
  The name.

- `nspace` Type: TwinCAT.TypeSystem.Generic.INamespace
  The namespace object (out-parameter)

**Return Value**

Type: Boolean
true if found, false if not contained.

---

**Reference**

NamespaceCollection.T. Class [2484]

TwinCAT.TypeSystem.Generic Namespace [2441]

---

### NamespaceCollection.T..TryGetType Method

Tries to get the specified type.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Parameters**

- `index` Type: System.Int32
  The zero-based index of the item to remove.

**Implements**

IList<T>.RemoveAt(Int32)

**Reference**

NamespaceCollection.T. Class [2484]

TwinCAT.TypeSystem.Generic Namespace [2441]
### TryGetType Method

- **Syntax**
  ```csharp
  public bool TryGetType(
    string typeName,
    out T dataType
  )
  ```

- **Parameters**
  - `typeName`: Type: `System.String`  
    Data type name.
  - `dataType`: Type: `T [2484]`.  
    The found data type (out-parameter).

- **Return Value**
  - Type: `Boolean`  
    true if found, false if not contained.

- **Exceptions**
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td><code>typeName</code></td>
</tr>
<tr>
<td>ArgumentException</td>
<td></td>
</tr>
</tbody>
</table>

- **Reference**
  - `NamespaceCollection.T` Class [2484]
  - `TwinCAT.TypeSystem.Generic` Namespace [2441]

### TryGetTypeByFullName Method

- **Syntax**
  ```csharp
  public bool TryGetTypeByFullName(
    string fullname,
    out T dataType
  )
  ```

- **Parameters**
  - `fullname`: Type: `System.String`  
    DataTypes full name.
  - `dataType`: Type: `T [2484]`.  
    Found data type (out-parameter).
Return Value

Type: Boolean
true if found, false if not contained.

Reference

NamespaceCollection.T. Class [2484]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.6 ReadOnlyDataTypeCollection.T. Class

ReadOnly Data Type collection

Inheritance Hierarchy

System.Object
    TwinCAT.TypeSystem.Generic.ReadOnlyDataTypeCollection.<T>
      TwinCAT.TypeSystem.ReadOnlyDataTypeCollection [2291]

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public class ReadOnlyDataTypeCollection<T> : ReadOnlyCollection<T>, IDataTypeCollection<T>, ICollection<T>, IEnumerable<T>, IEnumerable
where T : class, IDataType

Type Parameters

T

The ReadOnlyDataTypeCollection.T. type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadOnlyDataTypeCollection&lt;T&gt; (ReadOnlyCollection&lt;T&gt;)</td>
<td>Initializes a new instance of the ReadOnlyDataTypeCollection.T. class.</td>
</tr>
<tr>
<td>ReadOnlyDataTypeCollection&lt;T&gt; (ReadOnlyDataTypeCollection&lt;T&gt;)</td>
<td>Initializes a new instance of the ReadOnlyDataTypeCollection.T. class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.T..)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection.T..)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the element with the specified type name.</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the IList&lt;T&gt; that the ReadOnlyCollection&lt;T&gt; wraps. (Inherited from ReadOnlyCollection&lt;T&gt;..)</td>
</tr>
</tbody>
</table>

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>Determines whether an element is in the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;T&gt;..)</td>
</tr>
<tr>
<td>ContainsType [2504]</td>
<td>Determines whether the specified name contains type.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire ReadOnlyCollection&lt;T&gt; to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection&lt;T&gt;..)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;T&gt;..)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection&lt;T&gt;.. (Inherited from ReadOnlyCollection&lt;T&gt;..)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetType [2505]</td>
<td>Tries to get the Type with the specified name out of the collection.</td>
</tr>
</tbody>
</table>

#### Reference

TwinCAT.TypeSystem.Generic Namespace [2441]

### 6.12.6.1 ReadOnlyDataTypeCollection<T>. Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadOnlyDataTypeCollection&lt;T&gt; (DataTypeCollection&lt;T&gt;) [2501]</td>
<td>Initializes a new instance of the ReadOnlyDataTypeCollection&lt;T&gt; [2499] class.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>2501</td>
<td>Initializes a new instance of the <code>ReadOnlyDataTypeCollection&lt;T&gt;</code> class.</td>
</tr>
</tbody>
</table>

**Reference**

- `ReadOnlyDataTypeCollection<T>` Class [2499]
- `TwinCAT.TypeSystem.Generic` Namespace [2441]

### 6.12.6.1.1 `ReadOnlyDataTypeCollection<T>. Constructor` (`DataTypeCollection<T>.`)  
Initializes a new instance of the `ReadOnlyDataTypeCollection<T>` class.

**Namespace:** `TwinCAT.TypeSystem.Generic` [2441]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public ReadOnlyDataTypeCollection<T>(DataTypeCollection<T> coll)
```

**Parameters**

- `coll`  
Type: `TwinCAT.TypeSystem.Generic.DataTypeCollection<T>`.  
The collection.

**Reference**

- `ReadOnlyDataTypeCollection<T>. Class` [2499]
- `ReadOnlyDataTypeCollection<T>. Overload` [2500]
- `TwinCAT.TypeSystem.Generic` Namespace [2441]

### 6.12.6.1.2 `ReadOnlyDataTypeCollection<T>. Constructor` (`ReadOnlyDataTypeCollection<T>.`)  
Initializes a new instance of the `ReadOnlyDataTypeCollection<T>` class.

**Namespace:** `TwinCAT.TypeSystem.Generic` [2441]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public ReadOnlyDataTypeCollection<T>(ReadOnlyDataTypeCollection<T> coll)
```
Parameters

coll  
Type: TwinCAT.TypeSystem.Generic.ReadOnlyDataTypeCollection [2499].T
[2499].

The coll.

Reference

ReadOnlyDataTypeCollection.T. Class [2499]
ReadOnlyDataTypeCollection.T. Overload [2500]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.6.2  ReadOnlyDataTypeCollection.T. Properties

The ReadOnlyDataTypeCollection.T. [2499] generic type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.T. [2499].)</td>
</tr>
<tr>
<td>Item.Int32.</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection.T. [2499].)</td>
</tr>
<tr>
<td>Item.String.</td>
<td>GETS the element with the specified type name. (Inherited from ReadOnlyCollection.T. [2499].)</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.T. [2499].)</td>
</tr>
</tbody>
</table>

Reference

ReadOnlyDataTypeCollection.T. Class [2499]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.6.2.1  ReadOnlyDataTypeCollection.T..Item Property

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32.</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection.T. [2499].)</td>
</tr>
<tr>
<td>Item.String.</td>
<td>GETS the element with the specified type name. (Inherited from ReadOnlyCollection.T. [2499].)</td>
</tr>
</tbody>
</table>

Reference

ReadOnlyDataTypeCollection.T. Class [2499]
TwinCAT.TypeSystem.Generic Namespace [2441]

ReadOnlyDataTypeCollection.T..Item Property (String)

Gets the element with the specified type name.
Namespace: TwinCAT.TypeSystem.Generic
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294*Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C# public T this[string name] { get; }

Parameters

name Type: System.String
The name.

Return Value

Type: T
T.

Implements

IDataTypeCollection.T..Item.String.

Reference

ReadOnlyDataTypeCollection.T. Class
Item Overload
TwinCAT.TypeSystem.Generic Namespace

6.12.6.3 ReadOnlyDataTypeCollection.T. Methods

The ReadOnlyDataTypeCollection.T. generic type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>Determines whether an element is in the ReadOnlyCollection.T. (Inherited from ReadOnlyCollection.T[2499].)</td>
</tr>
<tr>
<td>ContainsType</td>
<td>Determines whether the specified name contains type.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.T[2499].)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection.T. (Inherited from ReadOnlyCollection.T[2499].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the</td>
</tr>
<tr>
<td></td>
<td>first occurrence within the entire ReadOnlyCollection.T.. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>ReadOnlyCollection.T[].)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetType</td>
<td>Tries to get the Type with the specified name out of the collection.</td>
</tr>
</tbody>
</table>

**Reference**

ReadOnlyDataTypeCollection.T.. Class [2499]

TwinCAT.TypeSystem.Generic Namespace [2441]

### 6.12.6.3.1 ReadOnlyDataTypeCollection.T..ContainsType Method

Determines whether the specified name contains type.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool ContainsType(
    string name
)
```

**Parameters**

- `name` Type: System.String
  
  The name.

**Return Value**

- Type: Boolean
  
  true if the specified name contains type; otherwise, false.

**Implements**

IDataTypeCollection.T..ContainsType(String) [1997]

**Reference**

ReadOnlyDataTypeCollection.T.. Class [2499]

TwinCAT.TypeSystem.Generic Namespace [2441]
6.12.6.3.2  **ReadOnlyDataTypeCollection.T..TryGetType Method**

Tries to get the Type with the specified name out of the collection.

**Namespace:**  TwinCAT.TypeSystem.Generic [2441]
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public bool TryGetType(
    string name,
    out T type
)
```

### Parameters

- **name**
  - **Type:** System.String
  - The name.

- **type**
  - **Type:** T [2499]
  - The type.

### Return Value

- **Type:** Boolean
  - true if found

### Implements

IDataTypeCollection.T..TryGetType(String, T.) [1998]

### Reference

ReadOnlyDataTypeCollection.T. Class [2499]

TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.7  **ReadOnlyInstanceCollection.T. Class**

ReadOnly Instance collection

### Inheritance Hierarchy

System.Object
  - System.Collections.ObjectModel.ReadOnlyCollection<T>
  - TwinCAT.TypeSystem.Generic.ReadOnlyInstanceCollection<T>
  - TwinCAT.TypeSystem.Generic.ReadOnlySymbolCollection<T> [2522]
  - TwinCAT.TypeSystem.Generic.ReadOnlyFieldCollection [2317]
  - TwinCAT.TypeSystem.Generic.ReadOnlyMemberCollection [2322]

**Namespace:**  TwinCAT.TypeSystem.Generic [2441]
**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14
### Syntax

**C#**

```csharp
public class ReadOnlyInstanceCollection<T> : ReadOnlyCollection<T>,
    IInstanceCollection<T>, IList<T>, ICollection<T>, IEnumerable<T>,
    IEnumerable
where T : class, IInstance
```

### Type Parameters

**T**

The `ReadOnlyInstanceCollection<T>` type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReadOnlyInstanceCollection(T)</code></td>
<td>Initializes a new instance of the <code>ReadOnlyInstanceCollection&lt;T&gt;</code> class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Count</code></td>
<td>Gets the number of elements contained in the <code>ReadOnlyCollection&lt;T&gt;</code> instance. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td><code>Item.Int32</code></td>
<td>Gets the element at the specified index. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td><code>Item.String</code></td>
<td>Gets the element with the specified instance path.</td>
</tr>
<tr>
<td><code>Items</code></td>
<td>Returns the <code>IList&lt;T&gt;</code> that the <code>ReadOnlyCollection&lt;T&gt;</code> wraps. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td><code>Mode</code></td>
<td>Gets the <code>InstanceCollectionMode</code>.[2075]</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Contains(T)</code></td>
<td>Determines whether an element is in the <code>ReadOnlyCollection&lt;T&gt;</code>. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td><code>Contains(String)</code></td>
<td>Determines whether the <code>ReadOnlyInstanceCollection&lt;T&gt;</code> contains an instance with the specified instance path.</td>
</tr>
<tr>
<td><code>ContainsName</code></td>
<td>Determines whether the specified instance is contained.</td>
</tr>
<tr>
<td><code>CopyTo</code></td>
<td>Copies the entire <code>ReadOnlyCollection&lt;T&gt;</code> to a compatible one-dimensional <code>Array</code>, starting at the specified index of the target array. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>)</td>
</tr>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetEnumerator</code></td>
<td>Returns an enumerator that iterates through the <code>ReadOnlyCollection&lt;T&gt;</code>. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetInstance</td>
<td>Gets the IInstance by instance path. [2512]</td>
</tr>
<tr>
<td>GetInstanceByName</td>
<td>Gets the IInstance by instance name. [2513]</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection&lt;T&gt;. (Inherited from ReadOnlyCollection&lt;T&gt;..)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetInstance</td>
<td>Tries to get the instance with the specified instance path. [2513]</td>
</tr>
<tr>
<td>TryGetInstanceByName</td>
<td>Tries to get the instance by name. [2514]</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.TypeSystem.Generic Namespace [2441]

#### 6.12.7.1 ReadOnlyInstanceCollection<T>. Constructor

Initializes a new instance of the ReadOnlyInstanceCollection<T> [2505] class.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public ReadOnlyInstanceCollection(
    IInstanceCollection<T> coll
)
```

### Parameters

- **coll**
  - Type: TwinCAT.TypeSystem.IInstanceCollection [2057].
  - The coll.

### Reference

ReadOnlyInstanceCollection<T>. Class [2505]

TwinCAT.TypeSystem.Generic Namespace [2441]

#### 6.12.7.2 ReadOnlyInstanceCollection<T>. Properties

The ReadOnlyInstanceCollection<T> [2505] generic type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>ReadOnlyCollection&lt;T&gt;</code> instance. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>.)</td>
</tr>
<tr>
<td>Item&lt;Int32&gt;</td>
<td>Gets the element at the specified index. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>.)</td>
</tr>
<tr>
<td>Item&lt;String&gt;</td>
<td>Gets the element with the specified instance path.</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the <code>IList&lt;T&gt;</code> that the <code>ReadOnlyCollection&lt;T&gt;</code> wraps. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>.)</td>
</tr>
<tr>
<td>Mode</td>
<td>Gets the <code>_InstanceCollectionMode</code> [2075].</td>
</tr>
</tbody>
</table>

Reference

`ReadOnlyInstanceCollection<T>. Class [2505]`

`TwinCAT.TypeSystem.Generic Namespace [2441]`

6.12.7.2.1 `ReadOnlyInstanceCollection<T>..Item Property`

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item&lt;Int32&gt;</td>
<td>Gets the element at the specified index. (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code>.)</td>
</tr>
<tr>
<td>Item&lt;String&gt;</td>
<td>Gets the element with the specified instance path.</td>
</tr>
</tbody>
</table>

Reference

`ReadOnlyInstanceCollection<T>. Class [2505]`

`TwinCAT.TypeSystem.Generic Namespace [2441]`

`ReadOnlyInstanceCollection<T>..Item Property (String)`

Gets the element with the specified instance path.

Namespace: `TwinCAT.TypeSystem.Generic [2441]`

Assembly: `TwinCAT.Ads (in TwinCAT.Ads.dll)` Version:

5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public T this[string instancePath] { get; }
```

Parameters

instancePath Type: `System.String`

The instance path.
Return Value

Type: `T` [2505]
The instance if contained.

Implements

`IInstanceCollection<T>.Item.String` [2059]

Reference

`ReadOnlyInstanceCollection<T>.Class` [2505]

`Item Overload` [2508]

`TwinCAT.TypeSystem.Generic Namespace` [2441]

6.12.7.2.2 `ReadOnlyInstanceCollection<T>.Mode` Property

Gets the `InstanceCollectionMode` [2075].

Namespace: `TwinCAT.TypeSystem.Generic` [2441]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public InstanceCollectionMode Mode { get; }
```

Property Value

Type: `InstanceCollectionMode` [2075]
The mode.

Implements

`IInstanceCollection<T>.Mode` [2060]

Reference

`ReadOnlyInstanceCollection<T>.Class` [2505]

`TwinCAT.TypeSystem.Generic Namespace` [2441]

6.12.7.3 `ReadOnlyInstanceCollection<T>.Methods`

The `ReadOnlyInstanceCollection<T>` [2505] generic type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>contain(T)</td>
<td>Determines whether an element is in the <code>ReadOnlyCollection&lt;T&gt;</code> (Inherited from <code>ReadOnlyCollection&lt;T&gt;</code> [2505]).</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

#### Contains(String)

**Description**
Determines whether the `ReadOnlyInstanceCollection.T. [2505]` contains an instance with the specified instance path.

#### ContainsName

**Description**
Determines whether the specified instance is contained.

#### CopyTo

**Description**
Copies the entire `ReadOnlyCollection.T.` to a compatible one-dimensional `Array`, starting at the specified index of the target array. (Inherited from `ReadOnlyCollection.T. [2505]`.)

#### Equals

**Description**
Determines whether the specified object is equal to the current object. (Inherited from `Object.`)

#### Finalize

**Description**
Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from `Object`.)

#### GetEnumerator

**Description**
Returns an enumerator that iterates through the `ReadOnlyCollection.T.`. (Inherited from `ReadOnlyCollection.T. [2505]`.)

#### GetHashCode

**Description**
Serves as the default hash function. (Inherited from `Object`.)

#### GetInstance

**Description**
Gets the `IInstance [2052]` by instance path.

#### GetInstanceByName

**Description**
Gets the `IInstance [2052]` by instance name.

#### GetType

**Description**
Gets the `Type` of the current instance. (Inherited from `Object`.)

#### IndexOf

**Description**
Searches for the specified object and returns the zero-based index of the first occurrence within the entire `ReadOnlyCollection.T.`. (Inherited from `ReadOnlyCollection.T. [2505]`.)

#### MemberwiseClone

**Description**
Creates a shallow copy of the current `Object`. (Inherited from `Object`.)

#### ToString

**Description**
Returns a string that represents the current object. (Inherited from `Object`.)

#### TryGetInstance

**Description**
Tries to get the instance with the specified instance path.

#### TryGetInstanceByName

**Description**
Tries to get the instance by name.

### Reference

- `ReadOnlyInstanceCollection.T. Class [2505]`
- `TwinCAT.TypeSystem.Generic Namespace [2441]`

### 6.12.7.3.1 `ReadOnlyInstanceCollection.T..Contains Method`

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the <code>ReadOnlyCollection.T.</code>. (Inherited from <code>ReadOnlyCollection.T. [2505]</code>.)</td>
</tr>
<tr>
<td>Contains(String) [2511]</td>
<td>Determines whether the <code>ReadOnlyInstanceCollection.T. [2505]</code> contains an instance with the specified instance path.</td>
</tr>
</tbody>
</table>
ReadOnlyInstanceCollection.T..Contains Method (String)

Determines whether the `ReadOnlyInstanceCollection.T.` contains an instance with the specified instance path.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public bool Contains(
    string instancePath
)
```

**Parameters**

- **instancePath**: Type: `System.String`
  
  The instance path.

**Return Value**

Type: `Boolean`

true if contains the specified instance path; otherwise, false.

**Implements**

`IInstanceCollection.T..Contains(String)`

**Reference**

- `ReadOnlyInstanceCollection.T..Contains` ([2505](#))
- `Contains Overload` ([2510](#))
- `TwinCAT.TypeSystem.Generic Namespace` ([2441](#))

---

6.12.7.3.2  ReadOnlyInstanceCollection.T..ContainsName Method

Determines whether the specified instance is contained.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#
```csharp
public bool ContainsName(
    string instanceName
)
```

Parameters

instanceName Type: System.String
Name of the instance.

Return Value

Type: Boolean
true, if instance name is found.

Implements

IInstanceCollection.T.ContainsName(String) [2062]

Reference

ReadOnlyInstanceCollection.T.Class [2505]
TwinCAT.TypeSystem.Generic.Namespace [2441]

6.12.7.3.3 ReadOnlyInstanceCollection.T..GetInstance Method

Gets the IInstance by instance path.

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
```csharp
public T GetInstance(
    string instancePath
)
```

Parameters

instancePath Type: System.String
The instance path.

Return Value

Type: T [2505]
T.

Implements

IInstanceCollection.T.GetInstance(String) [2062]
6.12.7.3.4 **ReadOnlyInstanceCollection.T..GetInstanceByName Method**

Gets the `IInstance` by instance name.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public IList<T> GetInstanceByName(
    string instanceName
)
```

**Parameters**

- **instanceName**
  - Type: System.String
  - Name of the instance.

**Return Value**

Type: `IList<T>`, `IList<T>`.

**Implements**

`IInstanceCollection.T..GetInstanceByName(String)`

Reference

ReadOnlyInstanceCollection.T. Class [2505]

TwinCAT.TypeSystem.Generic Namespace [2441]

---

6.12.7.3.5 **ReadOnlyInstanceCollection.T..TryGetInstance Method**

Tries to get the instance with the specified instance path.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public bool TryGetInstance(
    string instancePath,
    out T instance
)
```

**Reference**

ReadOnlyInstanceCollection.T. Class [2505]

TwinCAT.TypeSystem.Generic Namespace [2441]
Parameters

instancePath
Type: System.String
The instance path.

instance
Type: T.2505
The instance.

Return Value

Type: Boolean
ture, if found, false if not contained.

Implements

IInstanceCollection.T.TryGetInstance(String, T.)2063

Reference

ReadOnlyInstanceCollection.T.Class2505
TwinCAT.TypeSystem.Generic Namespace2441

6.12.7.3.6 ReadOnlyInstanceCollection.T..TryGetInstanceByName Method

Tries to get the instance by name.

Namespace: TwinCAT.TypeSystem.Generic2441
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool TryGetInstanceByName(
    string instanceName,
    out IList<T> symbols
)

Parameters

instanceName
Type: System.String
Name of the instance.

symbols
Type: System.Collections.Generic.IList<T>.2505
The found symbols (out-parameter)

Return Value

Type: Boolean
ture, if found; false if not contained.

Implements

IInstanceCollection.T.TryGetInstanceByName(String, IList<T>).2064
Reference
ReadOnlyInstanceCollection.T. Class [2505]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.8 ReadOnlyNamespaceCollection.T. Class
Read Only namespace collection

Inheritance Hierarchy
System.Object
      TwinCAT.TypeSystem.Generic.ReadOnlyNamespaceCollection.T.

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#
public class ReadOnlyNamespaceCollection<T> : ReadOnlyCollection<INamespace<T>>,
   INamespaceCollection<T>, ICollection<INamespace<T>>, IEnumerable<INamespace<T>>, IEnumerable
where T : class, IDataType

Type Parameters

T

The ReadOnlyNamespaceCollection.T. type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllTypes</td>
<td>Gets all types included in all namespaces.</td>
</tr>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.INamespace&lt;T&gt;.)</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection.INamespace&lt;T&gt;.)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the element at the specified index.</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the IList&lt;T&gt; that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.INamespace&lt;T&gt;.)</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>Determines whether an element is in the <code>ReadOnlyCollection&lt;T&gt;</code> (Inherited from <code>ReadOnlyCollection&lt;INamespace&gt;</code>).[2458]T...</td>
</tr>
<tr>
<td>ContainsNamespace</td>
<td>Determines whether this collection contains a namespace with the specified name.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire <code>ReadOnlyCollection&lt;T&gt;</code> to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from <code>ReadOnlyCollection&lt;INamespace&gt;</code>).[2458]T...</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the <code>ReadOnlyCollection&lt;T&gt;</code> (Inherited from <code>ReadOnlyCollection&lt;INamespace&gt;</code>).[2458]T...</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire <code>ReadOnlyCollection&lt;T&gt;</code> (Inherited from <code>ReadOnlyCollection&lt;INamespace&gt;</code>).[2458]T...</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>TryGetNamespace</td>
<td>Tries to get the namespace with the specified name.</td>
</tr>
<tr>
<td>TryGetType</td>
<td>Tries to get the specified data type.</td>
</tr>
<tr>
<td>TryGetTypeByName</td>
<td>Tries to get the specified type (by fullName)</td>
</tr>
</tbody>
</table>

## Reference

TwinCAT.TypeSystem.Generic Namespace.[2441]  

#### 6.12.8.1 `ReadOnlyNamespaceCollection<T>. Constructor`

Initializes a new instance of the `ReadOnlyNamespaceCollection<T>`.[2515] class.

**Namespace:** TwinCAT.TypeSystem.Generic.[2441]  
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934f6dca3e72b0ea15da1c14  

**Syntax**

C#  

```csharp
public ReadOnlyNamespaceCollection(
    NamespaceCollection<T> coll
)
```
Parameters

coll

The coll.

Reference

ReadOnlyNamespaceCollection.T. Class [2515]

TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.8.2 ReadOnlyNamespaceCollection.T. Properties

The ReadOnlyNamespaceCollection.T. [2515] generic type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllTypes [2517]</td>
<td>Gets all types included in all namespaces.</td>
</tr>
</tbody>
</table>
| Count               | Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.INamespace [2458] T [2515]...)
| Item.Int32          | Gets the element at the specified index. (Inherited from ReadOnlyCollection.INamespace [2458] T [2515]...)
| Item.String [2518]  | Gets the element at the specified index.         |
| Items               | Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.INamespace [2458] T [2515]...)

Reference

ReadOnlyNamespaceCollection.T. Class [2515]

TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.8.2.1 ReadOnlyNamespaceCollection.T..AllTypes Property

Gets all types included in all namespaces.

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public IDataTypeCollection<T> AllTypes { get; }

Property Value

Type: IDataTypeCollection [1995] T [2515].
All types.
### 6.12.8.2.2  ReadOnlyNamespaceCollection.T..Item Property

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item.Int32</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection.INamespace[2458],[2515]...)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the element at the specified index.</td>
</tr>
</tbody>
</table>

**ReadOnlyNamespaceCollection.T..Item Property (String)**

Gets the element at the specified index.

**Namespace:**  TwinCAT.TypeSystem.Generic [2441]

**Assembly:**  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public INamespace<T> this[string name] { get; }
```

**Parameters**

- **name**  
  Type: System.String  
  The name.

**Return Value**

Type: INamespace[2458], T[2515].

**Reference**

- ReadOnlyNamespaceCollection.T.Class [2515]
- TwinCAT.TypeSystem.Generic Namespace [2441]
6.12.8.3  **ReadOnlyNamespaceCollection.T. Methods**

The `ReadOnlyNamespaceCollection.T` [› 2515] generic type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>Determines whether an element is in the <code>ReadOnlyCollection.T</code>. (Inherited from <code>ReadOnlyCollection_INamespace [› 2458] T [› 2515]</code> ...)</td>
</tr>
<tr>
<td>ContainsNamespace</td>
<td>Determines whether this collection contains a namespace with the specified name.</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire <code>ReadOnlyCollection.T</code> to a compatible one-dimensional <code>Array</code>, starting at the specified index of the target array. (Inherited from <code>ReadOnlyCollection_INamespace [› 2458] T [› 2515]</code> ...)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the <code>ReadOnlyCollection.T</code>. (Inherited from <code>ReadOnlyCollection_INamespace [› 2458] T [› 2515]</code> ...)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire <code>ReadOnlyCollection.T</code>. (Inherited from <code>ReadOnlyCollection_INamespace [› 2458] T [› 2515]</code> ...)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>TryGetNamespace</td>
<td>Tries to get the namespace with the specified name.</td>
</tr>
<tr>
<td>TryGetType</td>
<td>Tries to get the specified data type.</td>
</tr>
<tr>
<td>TryGetTypeByFullname</td>
<td>Tries to get the specified type (by fullName)</td>
</tr>
</tbody>
</table>

### Reference

- `ReadOnlyNamespaceCollection.T.Class` [› 2515]
- `TwinCAT.TypeSystem.Generic Namespace` [› 2441]

---

6.12.8.3.1  **ReadOnlyNamespaceCollection.T.ContainsNamespace Method**

Determines whether this collection contains a namespace with the specified name.

**Namespace:** TwinCAT.TypeSystem.Generic [› 2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:

5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
TwinCAT.Ads Namespaces

Syntax

C#

```c#
public bool ContainsNamespace(
    string name
)
```

Parameters

name Type: System.String
The name of the namespace

Return Value

Type: Boolean
true if the namespace is contained; otherwise, false.

Reference

ReadOnlyNamespaceCollection.T. Class [2515]
TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.8.3.2  ReadOnlyNamespaceCollection.T..TryGetNamespace Method

Tries to get the namespace with the specified name.

Namespace:  TwinCAT.TypeSystem.Generic [2441]
Assembly:  TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```c#
public bool TryGetNamespace(
    string name,
    out INamespace<T> nspace
)
```

Parameters

name Type: System.String
Namespace name.

nspace Type: TwinCAT.TypeSystem.Generic.INamespace [2458].T [2515].
The found namespace (out-parameter).

Return Value

Type: Boolean
true if found, false if not contained.

Reference

ReadOnlyNamespaceCollection.T. Class [2515]
TwinCAT.TypeSystem.Generic Namespace [2441]
6.12.8.3.3  ReadOnlyNamespaceCollection.T..TryGetType Method

Tries to get the specified data type.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public bool TryGetType(
    string typeName,
    out T dataType
)
```

**Parameters**

- **typeName**  
  Type: System.String  
  Name of the type.

- **dataType**  
  Type: T  
  Data Type (out-parameter).

**Return Value**

Type: Boolean  
true if found, false if not contained.

**Reference**

ReadOnlyNamespaceCollection.T. Class

TwinCAT.TypeSystem.Generic Namespace

6.12.8.3.4  ReadOnlyNamespaceCollection.T..TryGetTypeByFullName Method

Tries to get the specified type (by fullName)

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public bool TryGetTypeByFullName(
    string fullName,
    out T dataType
)
```

**Parameters**

- **fullName**  
  Type: System.String  
  FullName of the data type.

- **dataType**  
  Type: T  
  Found Data type (out-parameter).
Return Value

Type: Boolean
true if found, false if not contained.

Reference

ReadOnlyNamespaceCollection.T. Class [2515]
TwinCAT.TypeSystem.Generic.Namespace [2441]

6.12.9 ReadOnlySymbolCollection.T. Class

Read only symbol collection.

Inheritance Hierarchy

System.Object
    TwinCAT.TypeSystem.Generic.ReadOnlyInstanceCollection>T.
      TwinCAT.TypeSystem.Generic.ReadOnlySymbolCollection.T.
        TwinCAT.TypeSystem.ReadOnlySymbolCollection [2336]

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT Ads (in TwinCAT Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd2ca3e72bc0ea15da1c4

Syntax

C#
public class ReadOnlySymbolCollection<T> : ReadOnlyInstanceCollection<T>
where T : class, ISymbol

Type Parameters

T

The ReadOnlySymbolCollection.T. type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadOnlySymbolCollection&lt;T&gt; [2524]</td>
<td>Initializes a new instance of the ReadOnlySymbolCollection.T. class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.T..)</td>
</tr>
<tr>
<td>Item.String [2508]</td>
<td>Gets the element with the specified instance path. (Inherited from ReadOnlyInstanceCollection&lt;T&gt; [2505].)</td>
</tr>
<tr>
<td>Item.Int32.</td>
<td>Gets the element at the specified index. (Inherited from ReadOnlyCollection.T..)</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.T..)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mode</strong> [2509]</td>
<td>Gets the InstanceCollectionMode [2075]. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Contains(String)</strong> [2511]</td>
<td>Determines whether the ReadOnlyInstanceCollection.T. [2505] contains an instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
<tr>
<td><strong>Contains(T)</strong></td>
<td>Determines whether an element is in the ReadOnlyCollection.T. (Inherited from ReadOnlyCollection.T..)</td>
</tr>
<tr>
<td><strong>ContainsName</strong> [2511]</td>
<td>Determines whether the specified instance is contained. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
<tr>
<td><strong>CopyTo</strong></td>
<td>Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.T..)</td>
</tr>
<tr>
<td><strong>Empty</strong> [2525]</td>
<td>Returns an empty collection.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong></td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T..)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetInstance</strong> [2512]</td>
<td>Gets the IInstance [2052] by instance path. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
<tr>
<td><strong>GetInstanceByName</strong> [2513]</td>
<td>Gets the IInstance [2052] by instance name. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>IndexOf</strong></td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T..</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>TryGetInstance</strong> [2513]</td>
<td>Tries to get the instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
<tr>
<td><strong>TryGetInstanceByName</strong> [2514]</td>
<td>Tries to get the instance by name. (Inherited from ReadOnlyInstanceCollection.T. [2505].)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.TypeSystem.Generic Namespace [2441]
6.12.9.1  **ReadOnlySymbolCollection.T. Constructor**

Initializes a new instance of the `ReadOnlySymbolCollection.T.` class.

**Namespace:** TwinCAT.TypeSystem.Generic

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fd2ca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public ReadOnlySymbolCollection(
    IInstanceCollection<T> coll
)
```

**Parameters**

`coll`  
Type: `TwinCAT.TypeSystem.IInstanceCollection`. The coll.

**Reference**

`ReadOnlySymbolCollection.T. Class`  
`TwinCAT.TypeSystem.Generic Namespace`

6.12.9.2  **ReadOnlySymbolCollection.T. Properties**

The `ReadOnlySymbolCollection.T.` generic type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Gets the number of elements contained in the <code>ReadOnlyCollection.T.</code> instance. (Inherited from <code>ReadOnlyCollection.T.</code>)</td>
</tr>
<tr>
<td>Item.String</td>
<td>Gets the element with the specified instance path. (Inherited from <code>ReadOnlyInstanceCollection.T.</code>)</td>
</tr>
<tr>
<td>Item.Int32</td>
<td>Gets the element at the specified index. (Inherited from <code>ReadOnlyCollection.T.</code>)</td>
</tr>
<tr>
<td>Items</td>
<td>Returns the <code>IList.T.</code> that the <code>ReadOnlyCollection.T.</code> wraps. (Inherited from <code>ReadOnlyCollection.T.</code>)</td>
</tr>
<tr>
<td>Mode</td>
<td>Gets the <code>InstanceCollectionMode</code>. (Inherited from <code>ReadOnlyInstanceCollection.T.</code>)</td>
</tr>
</tbody>
</table>

**Reference**

`ReadOnlySymbolCollection.T. Class`  
`TwinCAT.TypeSystem.Generic Namespace`

6.12.9.3  **ReadOnlySymbolCollection.T. Methods**

The `ReadOnlySymbolCollection.T.` generic type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains(String)</td>
<td>Determines whether the ReadOnlyInstanceCollection{T}内陆含有指定实例路径的实例。（继承自ReadOnlyInstanceCollection{T}）</td>
</tr>
<tr>
<td>Contains(T)</td>
<td>Determines whether an element is in the ReadOnlyCollection{T}。（继承自ReadOnlyCollection{T}）。</td>
</tr>
<tr>
<td>ContainsName</td>
<td>Determines whether the specified instance is contained。（继承自ReadOnlyInstanceCollection{T}）</td>
</tr>
<tr>
<td>CopyTo</td>
<td>Copies the entire ReadOnlyCollection{T} to a compatible one-dimensional Array, starting at the specified index of the target array。（继承自ReadOnlyCollection{T}）。</td>
</tr>
<tr>
<td>Empty</td>
<td>Returns an empty collection。</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object。（Inherited from Object）</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection。（Inherited from Object）</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the ReadOnlyCollection{T}。（Inherited from ReadOnlyCollection{T}）。</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function。（Inherited from Object）。</td>
</tr>
<tr>
<td>GetInstance</td>
<td>Gets the IInstance by instance path。（Inherited from ReadOnlyInstanceCollection{T}）。</td>
</tr>
<tr>
<td>GetInstanceByName</td>
<td>Gets the IInstance by instance name。（Inherited from ReadOnlyInstanceCollection{T}）。</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance。（Inherited from Object）。</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection{T}。（Inherited from ReadOnlyCollection{T}）。</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object。（Inherited from Object）。</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object。（Inherited from Object）。</td>
</tr>
<tr>
<td>TryGetInstance</td>
<td>Tries to get the instance with the specified instance path。（Inherited from ReadOnlyInstanceCollection{T}）。</td>
</tr>
<tr>
<td>TryGetInstanceByName</td>
<td>Tries to get the instance by name。（Inherited from ReadOnlyInstanceCollection{T}）。</td>
</tr>
</tbody>
</table>

Reference

ReadOnlySymbolCollection{T} Class [2522]

TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.9.3.1  ReadOnlySymbolCollection{T}..Empty Method

Returns an empty collection.
Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static ReadOnlySymbolCollection<T> Empty()
```

Return Value

Type: ReadOnlySymbolCollection[T].

Reference

ReadOnlySymbolCollection.T. Class [2522]

TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.10 SymbolCollection.T. Class

Interface represents a collection of ISymbol objects.

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Generic.InstanceCollection[T].
  TwinCAT.TypeSystem.Generic.SymbolCollection.T.
  TwinCAT.TypeSystem.SymbolCollection[2396]

Namespace: TwinCAT.TypeSystem.Generic [2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class SymbolCollection<T> : InstanceCollection<T>, ISymbolCollection<T>, IInstanceCollection<T>, IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
where T : class, ISymbol
```

Type Parameters

T

The SymbolCollection.T. type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count [2464]</td>
<td>Gets the collection count. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>InnerList [2465]</td>
<td>Gets the List of instances. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>InnerPathDict [2465]</td>
<td>The Path dictionary (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsReadOnly [2465]</td>
<td>Gets a value indicating whether this instance is read only. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Item[Int32][2466]</td>
<td>Gets or sets the IInstance [2052] at the specified index. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Item[String][2467]</td>
<td>Gets the IInstance [2052] with the specified instance path. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Mode [2468]</td>
<td>The mode this InstanceCollection.T. [2460] is working in. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add [2469]</td>
<td>Adds the specified item. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>AddRange [2470]</td>
<td>Adds the specified items to this collection. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>AsReadOnly [2530]</td>
<td>Returns a Read only version of this collection (shallow copy).</td>
</tr>
<tr>
<td>Clear [2471]</td>
<td>Clears this instance. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Clone [2530]</td>
<td>Clones this instance.</td>
</tr>
<tr>
<td>Contains(String)[2472]</td>
<td>Determines whether this collection contains an IInstance [2052] with the specified InstanceName / InstancePath (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Contains(T)[2472]</td>
<td>Determines whether this collection contains the specified IInstance [2052] (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>ContainsName[2473]</td>
<td>Determines whether the specified instance name contains name. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>CopyTo [2474]</td>
<td>Copies this InstanceCollection.T. [2460] to the specified array. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Empty [2531]</td>
<td>Creates an Empty SymbolCollection.T.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator[2474]</td>
<td>Gets the enumerator. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetInstance [2475]</td>
<td>Gets the IInstance [2052] by instance path. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>GetInstanceByName[2476]</td>
<td>Gets the name of the instance by. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf [2476]</td>
<td>Determines the index of the specified IInstance [2052]. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Insert [2477]</td>
<td>Inserts the specified IInstance [2052] at the specified index. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
</tbody>
</table>
### SymbolCollection.T. Properties

The `SymbolCollection.T. [\ref{2526}]` generic type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Remove [\ref{2478}]</td>
<td>Removes the specified item. (Inherited from <code>InstanceCollection.T. [\ref{2460}]</code>.)</td>
</tr>
<tr>
<td>RemoveAt [\ref{2479}]</td>
<td>Removes the <code>IInstance [\ref{2052}]</code> at the specified index. (Inherited from <code>InstanceCollection.T. [\ref{2460}]</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>TryGetInstance [\ref{2479}]</td>
<td>Tries to get the <code>IInstance [\ref{2052}]</code> of the specified path. (Inherited from <code>InstanceCollection.T. [\ref{2460}]</code>.)</td>
</tr>
<tr>
<td>TryGetInstanceByName [\ref{2480}]</td>
<td>Tries to get <code>IInstance</code> by name. (Inherited from <code>InstanceCollection.T. [\ref{2460}]</code>.)</td>
</tr>
<tr>
<td>TryGetInstances [\ref{2531}]</td>
<td>Try to get instances with predicate function</td>
</tr>
</tbody>
</table>

### SymbolCollection.T. Methods

The `SymbolCollection.T. [\ref{2526}]` generic type exposes the following members.

#### Reference

- `TwinCAT.TypeSystem.Generic Namespace [\ref{2441}]`

### 6.12.10.1 SymbolCollection.T. Properties

#### Reference

- `SymbolCollection.T. Class [\ref{2526}]`

### 6.12.10.2 SymbolCollection.T. Methods

#### Reference

- `TwinCAT.TypeSystem.Generic Namespace [\ref{2441}]`
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add [2469]</td>
<td>Adds the specified item. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>AddRange [2470]</td>
<td>Adds the specified items to this collection. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>AsReadOnly [2530]</td>
<td>Returns a Read only version of this collection (shallow copy).</td>
</tr>
<tr>
<td>Clear [2471]</td>
<td>Clears this instance. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Clone [2530]</td>
<td>Clones this instance.</td>
</tr>
<tr>
<td>Contains(String)</td>
<td>Determines whether this collection contains an IInstance [2052] with the specified InstanceName / InstancePath (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Contains(T) [2472]</td>
<td>Determines whether this collection contains the specified IInstance [2052] (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>ContainsName [2473]</td>
<td>Determines whether the specified instance name contains name. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>CopyTo [2474]</td>
<td>Copies this InstanceCollection.T. [2460] to the specified array. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Empty [2531]</td>
<td>Creates an Empty SymbolCollection.T. [2526]</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator [2474]</td>
<td>Gets the enumerator. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetInstance [2475]</td>
<td>Gets the IInstance [2052] by instance path. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>GetInstanceByName</td>
<td>Gets the name of the instance by. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf [2476]</td>
<td>Determines the index of the specified IInstance [2052]. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>Insert [2477]</td>
<td>Inserts the specified IInstance [2052] at the specified index. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Remove [2478]</td>
<td>Removes the specified item. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>RemoveAt [2479]</td>
<td>Removes the IInstance [2052] at the specified index. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetInstance</td>
<td>Tries to get the IInstance [2052] of the specified path. (Inherited from InstanceCollection.T. [2460].)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TryGetInstanceByName</td>
<td>Tries to get Instances by name. (Inherited from InstanceCollection&lt;T&gt;.)</td>
</tr>
<tr>
<td>TryGetInstances</td>
<td>Try to get instances with predicate function</td>
</tr>
</tbody>
</table>

#### Reference

- SymbolCollection<T> Class [2526]
- TwinCAT.TypeSystem.Generic Namespace [2441]

#### 6.12.10.2.1 SymbolCollection<T>.AsReadOnly Method

Returns a Read only version of this collection (shallow copy).

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public ReadOnlySymbolCollection<T> AsReadOnly()
```

**Return Value**

Type: `ReadOnlySymbolCollection<T>`. `SymbolCollection<T>.AsReadOnly()`.

**Reference**

- SymbolCollection<T> Class [2526]
- TwinCAT.TypeSystem.Generic Namespace [2441]

#### 6.12.10.2.2 SymbolCollection<T>.Clone Method

Clones this instance.

**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
public SymbolCollection<T> Clone()
```

**Return Value**

6.12.10.2.3 SymbolCollection.T..Empty Method

Creates an Empty SymbolCollection.T.

Namespace: TwinCAT.TypeSystem.Generic
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public static SymbolCollection<T> Empty()

Return Value

Type: SymbolCollection<T>, SymbolCollection<T>.

Reference

SymbolCollection.T..Empty Method
SymbolCollection.T..TryGetInstances Method

6.12.10.2.4 SymbolCollection.T..TryGetInstances Method

Try to get instances with predicate function

Namespace: TwinCAT.TypeSystem.Generic
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool TryGetInstances(
    Func<T, bool> predicate,
    bool recurse,
    out IList<T> instances
)

Parameters

predicate Type: System.Func<T, Boolean>
The predicate function

recurse Type: System.Boolean
if set to true the symbol hierarchy will be searched recursively.

instances Type: System.Collections.Generic.IList<T>
The instances.
Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

SymbolCollection.T. Class [.] SymbolCollection.T. Class
TwinCAT.TypeSystem.Generic Namespace [.] TwinCAT.TypeSystem.Generic Namespace

6.12.11 SymbolIterationMask Enumeration

Mask Flagset to specify filters for SymbolIterator.T. [.] SymbolIterator.T.

Namespace: TwinCAT.TypeSystem.Generic [.] TwinCAT.TypeSystem.Generic
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

[FlagsAttribute]
public enum SymbolIterationMask

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>Uninitialized / None</td>
</tr>
<tr>
<td>Structures</td>
<td>1</td>
<td>Iterates over Subelements of Structs</td>
</tr>
<tr>
<td>Arrays</td>
<td>2</td>
<td>Iterates over Elements of Arrays</td>
</tr>
<tr>
<td>Unions</td>
<td>4</td>
<td>Iterates over Subelements of Unions</td>
</tr>
<tr>
<td>Pointer</td>
<td>8</td>
<td>Iterates over Pointer SubElements</td>
</tr>
<tr>
<td>References</td>
<td>16</td>
<td>Iterates over References</td>
</tr>
<tr>
<td>All</td>
<td>31</td>
<td>Iterates over All Complex/Combined types</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.TypeSystem.Generic Namespace [.] TwinCAT.TypeSystem.Generic

6.12.12 SymbolIterator.T. Class

Iterator class for enumerations of Symbols [.] Symbols.

Inheritance Hierarchy

System.Object
TwinCAT.TypeSystem.Generic.SymbolIterationMask
TwinCAT.Ads.TypeSystem.SymbolIterator [.] TwinCAT.Ads.TypeSystem.SymbolIterator

Namespace: TwinCAT.TypeSystem.Generic [.] TwinCAT.TypeSystem.Generic
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public class SymbolIterator<T> : IEnumerable<T>,
    IEnumerable
where T : class, ISymbol
```

Type Parameters

T

Concrete `ISymbol` type.

The SymbolIterator.T type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask [2538]</td>
<td>Gets or sets the <code>SymbolIterationMask</code> [2532]</td>
</tr>
<tr>
<td>SymbolRecursionDetection [2538]</td>
<td>Gets or sets a value indicating whether the iterator checks for Symbol recursions (true by default).</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetEnumerator [2539]</td>
<td>Gets the enumerator that enumerates through a collection</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
TwinCAT.Ads Namespaces
Name

Description

ToString

Returns a string that represents the current object. (Inherited from Object.)

Remarks
This iterator class can be used to iterate over collections of symbol trees (root symbols + sub symbols). By
constructor the user can choose if the iterator works recursively within the symbol tree and optionally a filter
function to select only specific symbols (predicate).
Examples
The following example shows how to determine, browse and filter symbols.
Browsing and filtering Symbols
using (AdsClient client = new AdsClient())
{
CancellationToken cancel = CancellationToken.None;
uint valueToRead = 0;
uint valueToWrite = 42;
client.Connect(AmsNetId.Local, 851);
// Load all Symbols + DataTypes
ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
ResultSymbols resultSymbols

= await loader.GetSymbolsAsync(cancel);

if (resultSymbols.Succeeded)
{
Symbol symbol = (Symbol)resultSymbols.Symbols["MAIN.nCounter"];
// Works for ALL Primitive 'ANY TYPES' Symbols
ResultWriteAccess resultWrite = await symbol.WriteValueAsync(valueToWrite, cancel);
ResultReadValueAccess resultRead = await symbol.ReadValueAsync(cancel);
if (resultRead.Succeeded)
valueToRead = (uint)resultRead.Value;
// Simple filtering of Symbols
Regex filterExpression = new Regex(pattern: @"^MAIN.*"); // Everything that starts with "MAIN"
// FilterFunction that filters for the InstancePath
Func<ISymbol, bool> filter = s => filterExpression.IsMatch(s.InstancePath);
SymbolIterator iterator = new SymbolIterator(symbols: resultSymbols.Symbols, recurse: true, pred
icate: filter);
foreach (ISymbol filteredSymbol in iterator)
{
Console.WriteLine(filteredSymbol.InstancePath);
}
}
}

Reference
TwinCAT.TypeSystem.Generic Namespace [} 2441]
System.Collections.Generic.IEnumerable.T.

2534

Version: 1.1

TC1000


## 6.12.12.1 SymbolIterator.T. Constructor

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SymbolIterator.T. (IInstanceCollection&lt;T&gt; symbols)</code></td>
</tr>
<tr>
<td><code>SymbolIterator.T. (IEnumerable&lt;T, Boolean&gt;)</code></td>
</tr>
<tr>
<td><code>SymbolIterator.T. (IInstanceCollection&lt;T, Func&lt;T, Boolean&gt;&gt;)</code></td>
</tr>
<tr>
<td><code>SymbolIterator.T. (IEnumerable&lt;T, Boolean, Func&lt;T, Boolean&gt;&gt;)</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
</table>

### Reference

- `SymbolIterator.T. Class [2532]`
- `TwinCAT.TypeSystem.Generic Namespace [2441]`

## 6.12.12.1.1 SymbolIterator.T. Constructor (IInstanceCollection.T.)


### Namespace: TwinCAT.TypeSystem.Generic [2441]

### Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Shal.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```csharp
public SymbolIterator(
    IInstanceCollection<T> symbols
)
```

### Parameters

- `symbols` Type: `TwinCAT.TypeSystem.IInstanceCollection<T> [2057], T [2532]`, The root symbols.

### Reference

- `SymbolIterator.T. Class [2532]`
- `SymbolIterator.T. Overload [2535]`
- `TwinCAT.TypeSystem.Generic Namespace [2441]`
6.12.12.1.2 SymbolIterator.T. Constructor (IEnumerable.T., Boolean)


**Namespace:** TwinCAT.TypeSystem.Generic [2441]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public SymbolIterator(
    IEnumerable<T> symbols,
    bool recurse
)
```

**Parameters**

- **symbols**  
  Type: System.Collections.Generic.IEnumerable<T> [2532].  
  The root collection

- **recurse**  
  Type: System.Boolean  
  if set to true, the iterator works recursively over all subsymbols.

**Reference**

SymbolIterator.T. Class [2532]
SymbolIterator.T. Overload [2535]
TwinCAT.TypeSystem.Generic Namespace [2441]


**Namespace:** TwinCAT.TypeSystem.Generic [2441]
**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#  
```csharp
public SymbolIterator(
    IInstanceCollection<T> symbols,
    Func<T, bool> predicate
)
```

**Parameters**

- **symbols**  
  Type: TwinCAT.TypeSystem.IInstanceCollection [2057].  
  The root symbols.

- **predicate**  
  Type: System.Func<T> [2532].  
  Boolean.
  The predicate.
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotSupportedException</td>
<td></td>
</tr>
</tbody>
</table>

### Reference

- SymbolIterator.T. Class [2532]
- SymbolIterator.T. Overload [2535]
- TwinCAT.TypeSystem.Generic Namespace [2441]


**Namespace:** TwinCAT.TypeSystem.Generic [2441]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### C#

```csharp
public SymbolIterator(
    IEnumerable<T> symbols,
    bool recurse,
    Func<T, bool> predicate
)
```

**Parameters**

- **symbols**
  - Type: System.Collections.Generic.IEnumerable<T> [2532]
  - Input collection (root objects).

- **recurse**
  - Type: System.Boolean
  - if set to true, the iterator works recursively over all subsymbols.

- **predicate**
  - Type: System.Func<T, Boolean> [2532]
  - The predicate.

### Reference

- SymbolIterator.T. Class [2532]
- SymbolIterator.T. Overload [2535]
- TwinCAT.TypeSystem.Generic Namespace [2441]

### 6.12.12.2 SymbolIterator.T. Properties

The SymbolIterator.T. [2532] generic type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask</td>
<td>Gets or sets the SymbolIterationMask [2532]</td>
</tr>
</tbody>
</table>
### SymbolRecursionDetection Property

Gets or sets a value indicating whether the iterator checks for Symbol recursions (true by default).

**Namespace**: TwinCAT.TypeSystem.Generic

**Assembly**: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

#### Syntax

**C#**

```csharp
public bool SymbolRecursionDetection { get; set; }
```

#### Property Value

Type: `Boolean`

true if recursion checking, false switched off check.

**Reference**

SymbolIterator.T..Class [2532]

TwinCAT.TypeSystem.Generic Namespace [2441]
6.12.12.3 SymbolIterator.T. Methods

The SymbolIterator.T. [2532] generic type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Gets the enumerator that enumerates through a collection</td>
</tr>
<tr>
<td>GetEnumerator()</td>
<td></td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

SymbolIterator.T. Class [2532]

TwinCAT.TypeSystem.Generic Namespace [2441]

6.12.12.3.1 SymbolIterator.T..GetEnumerator Method

Gets the enumerator that enumerates through a collection

Namespace: TwinCAT.TypeSystem.Generic [2441]

Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public IEnumerator<T> GetEnumerator()
```

Return Value

Type: [IEnumerator.T] [2532].

A [IEnumerator.T] that can be used to iterate through the collection.

Implements

IEnumerator.T.GetEnumerator.
6.13 TwinCAT.ValueAccess Namespace

Namespace for the common (non ADS dependent) value access.

Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotAccessException</td>
<td>Class CannotAccessException. This class cannot be inherited. Implements the SymbolException [2401]</td>
</tr>
<tr>
<td>ResultAccess</td>
<td>Result class for an asynchronous access operation.</td>
</tr>
<tr>
<td>ResultReadDynamicValueAccess</td>
<td>Asynchronous read access result returning a Dynamic value object (IDynamicValue [2011]). Implements the ResultReadValueAccess.T [2568]</td>
</tr>
<tr>
<td>ResultReadRawAccess</td>
<td>Asynchronous read access result object, reading raw (byte[]) data into memory locations. Implements the ResultReadValueAccess.T [2568]</td>
</tr>
<tr>
<td>ResultReadValueAccess</td>
<td>Asynchronous read access result returning an untyped Value object type. Implements the ResultAccess [2556]</td>
</tr>
<tr>
<td>ResultReadValueAccess.T</td>
<td>Result object of an asynchronous read of a specific value of type T. Implements the ResultAccess [2556]</td>
</tr>
<tr>
<td>ResultRpcMethodAccess</td>
<td>Asynchronous Invoke RPC Method result class. Implements the ResultReadValueAccess [2566]</td>
</tr>
<tr>
<td>ResultWriteAccess</td>
<td>Asynchronous write access result class.</td>
</tr>
</tbody>
</table>

Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAccessorRawValue</td>
<td>Helper Interface to access Symbol Values as byte Arrays</td>
</tr>
<tr>
<td>IAccessorValueFactory</td>
<td>Factory interfaces for Accessor implementations.</td>
</tr>
</tbody>
</table>

Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolNotificationTypes</td>
<td>Specifies the Notification type of ADS Notifications</td>
</tr>
<tr>
<td>ValueCreationModes</td>
<td>Creation mode for Values</td>
</tr>
</tbody>
</table>
6.13.1 CannotAccessValueException Class

Class CannotAccessValueException. This class cannot be inherited. Implements the SymbolException

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [2541]
      TwinCAT.TypeSystem.SymbolException [2401]
        TwinCAT.ValueAccess.CannotAccessValueException

Namespace: TwinCAT.ValueAccess [2540]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

[SerializableAttribute]
public sealed class CannotAccessValueException : SymbolException

The CannotAccessValueException type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotAccessValueException</td>
<td>Initializes a new instance of the CannotAccessValueException class.</td>
</tr>
<tr>
<td>CannotAccessValueException(ISymbol)</td>
<td>Initializes a new instance of the CannotAccessValueException class.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InstancePath</td>
<td>Gets the instance path. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbol [2411]</td>
<td>Gets the symbol. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>.GetObjectData [2412]</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.ValueAccess Namespace [2540]

TwinCAT.TypeSystem.SymbolException [2401]

### 6.13.1.1 CannotAccessValueException Constructor

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotAccessValueException(ISymbol [2543])</td>
<td>Initializes a new instance of the CannotAccessValueException [2541] class.</td>
</tr>
</tbody>
</table>

### Reference

CannotAccessValueException Class [2541]

TwinCAT.ValueAccess Namespace [2540]

### 6.13.1.1.1 CannotAccessValueException Constructor

Initializes a new instance of the CannotAccessValueException [2541] class.

**Namespace:** TwinCAT.ValueAccess [2540]

**Assembly:** TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
public CannotAccessException()
```

Reference

CannotAccessException Class [2541]

CannotAccessException Overload [2542]

TwinCAT.ValueAccess Namespace [2540]

6.13.1.1.2 CannotAccessException Constructor (ISymbol)

Initializes a new instance of the CannotAccessException [2541] class.

Namespace: TwinCAT.ValueAccess [2540]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public CannotAccessException(
    ISymbol symbol
)
```

Parameters

symbol Type: TwinCAT.TypeSystem.ISymbol [2176]
The symbol.

Reference

CannotAccessException Class [2541]

CannotAccessException Overload [2542]

TwinCAT.ValueAccess Namespace [2540]

6.13.1.2 CannotAccessException Properties

The CannotAccessException [2541] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstancePath</td>
<td>Gets the instance path. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Symbol [2411]</td>
<td>Gets the symbol. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

#### Reference

CannotAccessException Class [2541]

TwinCAT.ValueAccess Namespace [2540]

### 6.13.1.3 CannotAccessException Methods

The CannotAccessException [2541] type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData [2412]</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException [2401].)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

#### Reference

CannotAccessException Class [2541]

TwinCAT.ValueAccess Namespace [2540]

### 6.13.2 IAccessorRawValue Interface

Helper Interface to access Symbol Values as byte Arrays

**Namespace:** TwinCAT.ValueAccess [2540]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: S.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

public interface IAccessorRawValue

The IAccessorRawValue type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DefaultValueEncoding [2546]</td>
<td>Gets the default value encoding.</td>
</tr>
<tr>
<td>ValueFactory [2546]</td>
<td>Gets the value factory</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadArrayElementRawAsync [2547]</td>
<td></td>
</tr>
<tr>
<td>ReadRawAsync [2548]</td>
<td></td>
</tr>
<tr>
<td>TryReadArrayElementRaw [2548]</td>
<td></td>
</tr>
<tr>
<td>TryReadRaw [2549]</td>
<td></td>
</tr>
<tr>
<td>TryWriteArrayElementRaw [2550]</td>
<td></td>
</tr>
<tr>
<td>TryWriteRaw [2550]</td>
<td></td>
</tr>
<tr>
<td>WriteArrayElementRawAsync [2551]</td>
<td></td>
</tr>
<tr>
<td>WriteRawAsync [2552]</td>
<td></td>
</tr>
</tbody>
</table>

Reference

TwinCAT.ValueAccess Namespace [2540]

6.13.2.1 IAccessorRawValue Properties

The IAccessorRawValue [2544] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DefaultValueEncoding [2546]</td>
<td>Gets the default value encoding.</td>
</tr>
</tbody>
</table>
### TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueFactory</td>
<td>Gets the value factory</td>
</tr>
</tbody>
</table>

#### 6.13.2.1.1 IAccessorRawValue.DefaultValueEncoding Property

Gets the default value encoding.

**Namespace:** TwinCAT.ValueAccess
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
Encoding DefaultValueEncoding { get; }
```

**Property Value**

- **Type:** Encoding
- The default value encoding.

**Reference**

IAccessorRawValue Interface
TwinCAT.ValueAccess Namespace

#### 6.13.2.1.2 IAccessorRawValue.ValueFactory Property

Gets the value factory

**Namespace:** TwinCAT.ValueAccess
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
IAccessorValueFactory ValueFactory { get; }
```

**Property Value**

- **Type:** IAccessorValueFactory
- The value factory or null if only Raw Values only.

**Reference**

IAccessorRawValue Interface
TwinCAT.ValueAccess Namespace
6.13.2.2   IAccessorRawValue Methods

The IAccessorRawValue [2544] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadArrayElementRawAsync</td>
<td></td>
</tr>
<tr>
<td>ReadRawAsync</td>
<td></td>
</tr>
<tr>
<td>TryReadArrayElementRaw</td>
<td></td>
</tr>
<tr>
<td>TryReadRaw</td>
<td></td>
</tr>
<tr>
<td>TryWriteArrayElementRaw</td>
<td></td>
</tr>
<tr>
<td>TryWriteRaw</td>
<td></td>
</tr>
<tr>
<td>WriteArrayElementRawAsync</td>
<td></td>
</tr>
<tr>
<td>WriteRawAsync</td>
<td></td>
</tr>
</tbody>
</table>

Reference

IAccessorRawValue Interface [2544]
TwinCAT.ValueAccess Namespace [2540]

6.13.2.2.1   IAccessorRawValue.ReadArrayElementRawAsync Method

Namespace:  TwinCAT.ValueAccess [2540]
Assembly:   TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
Task<ResultReadRawAccess> ReadArrayElementRawAsync(
    IArrayInstance arrayInstance,
    int[] indices,
    Memory readBuffer,
    void cancel
)
```

Parameters

arrayInstance                Type: TwinCAT.TypeSystem.IArrayInstance [1964]
indices                       Type: System.Int32.
readBuffer Type: Memory

cancel Type: System.Void

Return Value
Type: Task<ResultReadRawAccess> [2564].

Reference
IAccessorRawValue Interface [2544]
TwinCAT.ValueAccess Namespace [2540]

6.13.2.2.2 IAccessorRawValue.ReadRawAsync Method
Namespace: TwinCAT.ValueAccess [2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
Task<ResultReadRawAccess> ReadRawAsync(
    ISymbol symbolInstance,
    Memory readBuffer,
    void cancel
)

Parameters
symbolInstance Type: TwinCAT.TypeSystem.ISymbol [2176]
readBuffer Type: Memory
cancel Type: System.Void

Return Value
Type: Task<ResultReadRawAccess> [2564].

Reference
IAccessorRawValue Interface [2544]
TwinCAT.ValueAccess Namespace [2540]

6.13.2.2.3 IAccessorRawValue.TryReadArrayElementRaw Method
Namespace: TwinCAT.ValueAccess [2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

**C#**

```csharp
int TryReadArrayElementRaw(
    IArrayInstance arrayInstance,
    int[] indices,
    Memory readBuffer,
    void timeStamp
)
```

**Parameters**

- `arrayInstance` Type: `TwinCAT.TypeSystem.IArrayInstance` [1964]
- `indices` Type: `System.Int32`
- `readBuffer` Type: `Memory`
- `timeStamp` Type: `System.Void`

**Return Value**

Type: `Int32`

**Reference**

- `IAccessorRawValue Interface` [2544]
- `TwinCAT.ValueAccess Namespace` [2540]

### 6.13.2.2.4 IAccessorRawValue.TryReadRaw Method

**Namespace:** `TwinCAT.ValueAccess` [2540]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

**C#**

```csharp
int TryReadRaw(
    ISymbol symbolInstance,
    Memory readBuffer,
    void timeStamp
)
```

**Parameters**

- `symbolInstance` Type: `TwinCAT.TypeSystem.ISymbol` [2176]
- `readBuffer` Type: `Memory`
- `timeStamp` Type: `System.Void`

**Return Value**

Type: `Int32`
Reference

IAccessorRawValue Interface [2544]
TwinCAT.ValueAccess Namespace [2540]

6.13.2.2.5 I_accessorRawValue.TryWriteArrayElementRaw Method

Namespace: TwinCAT.ValueAccess [2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C# int TryWriteArrayElementRaw( IArrayInstance arrayInstance, int[] indices, ReadOnlyMemory writeBuffer, void timeSignature )

Parameters

arrayInstance Type: TwinCAT.TypeSystem.IArrayInstance [1964]
indices Type: System.Int32
writeBuffer Type: ReadOnlyMemory
timeSignature Type: System.Void

Return Value

Type: Int32

Reference

IAccessorRawValue Interface [2544]
TwinCAT.ValueAccess Namespace [2540]

6.13.2.2.6 I_accessorRawValue.TryWriteRaw Method

Namespace: TwinCAT.ValueAccess [2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

Syntax

C# int TryWriteRaw(ISymbol symbolInstance, ReadOnlyMemory writeBuffer, void timeSignature)
### Parameters

- **symbolInstance**
  - Type: `TwinCAT.TypeSystem.ISymbol`

- **writeBuffer**
  - Type: `ReadOnlyMemory`

- **timeStamp**
  - Type: `System.Void`

### Return Value

- Type: `Int32`

### Reference

- `IAccessorRawValue Interface`
- `TwinCAT.ValueAccess Namespace`

---

### 6.13.2.2.7 `IAccessorRawValue.WriteArrayElementRawAsync Method`

#### Namespace

- `TwinCAT.ValueAccess`

#### Assembly

- `TwinCAT.Ads.Abstractions.dll` (Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14)

#### Syntax

**C#**

```csharp
Task<ResultWriteAccess> WriteArrayElementRawAsync(
    IArrayInstance arrayInstance,
    int[] indices,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

#### Parameters

- **arrayInstance**
  - Type: `TwinCAT.TypeSystem.IArrayInstance`

- **indices**
  - Type: `System.Int32`

- **writeBuffer**
  - Type: `ReadOnlyMemory`

- **cancel**
  - Type: `System.Void`

#### Return Value

- Type: `Task.ResultWriteAccess`

#### Reference

- `IAccessorRawValue Interface`
- `TwinCAT.ValueAccess Namespace`
### 6.13.2.8 IAccessorRawValue.WriteRawAsync Method

**Namespace:** TwinCAT.ValueAccess

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
Task<ResultWriteAccess> WriteRawAsync(
    ISymbol symbolInstance,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

**Parameters**

- **symbolInstance**
  - Type: TwinCAT.TypeSystem.ISymbol

- **writeBuffer**
  - Type: ReadOnlyMemory

- **cancel**
  - Type: System.Void

**Return Value**

- Type: Task<ResultWriteAccess>

**Reference**

IAccessorRawValue Interface

TwinCAT.ValueAccess Namespace

### 6.13.3 IAccessorValueFactory Interface

Factory interfaces for Accessor implementations.

**Namespace:** TwinCAT.ValueAccess

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public interface IAccessorValueFactory
```

The IAccessorValueFactory type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="CreatePrimitiveValue" alt="image" /></td>
<td>CreatePrimitiveValue</td>
</tr>
</tbody>
</table>


### Remarks

This interface is used by a custom Accessor class to create Value objects from memory representations.

### Reference

TwinCAT.ValueAccess Namespace [еныч. 2540]

#### 6.13.3.1 IAccessorValueFactory Methods

The `IAccessorValueFactory` [еныч. 2552] type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CreatePrimitiveValue</code> [еныч. 2553]</td>
<td></td>
</tr>
<tr>
<td><code>CreateValue(ISymbol, ReadOnlyMemory, Void)</code> [еныч. 2554]</td>
<td></td>
</tr>
<tr>
<td><code>CreateValue(ISymbol, ReadOnlyMemory, Void, Byte)</code> [еныч. 2555]</td>
<td></td>
</tr>
</tbody>
</table>

### Reference

`IAccessorValueFactory Interface` [еныч. 2552]

TwinCAT.ValueAccess Namespace [еныч. 2540]

#### 6.13.3.1.1 IAccessorValueFactory.CreatePrimitiveValue Method

**Namespace:** TwinCAT.ValueAccess [еныч. 2540]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

Object CreatePrimitiveValue(
    ISymbol symbol,
    ReadOnlyMemory sourceData,
    void timeStamp
)

Parameters

symbol Type: TwinCAT.TypeSystem.ISymbol

sourceData Type: ReadOnlyMemory

timeStamp Type: System.Void

Return Value

Type: Object

Reference

IAccessorValueFactory Interface
TwinCAT.ValueAccess Namespace

6.13.3.1.2 IAccessorValueFactory.CreateValue Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CreateValue(ISymbol, ReadOnlyMemory, Void)</td>
</tr>
<tr>
<td></td>
<td>CreateValue(ISymbol, ReadOnlyMemory, Void, Byte)</td>
</tr>
</tbody>
</table>

Reference

IAccessorValueFactory Interface
TwinCAT.ValueAccess Namespace

IAccessorValueFactory.CreateValue Method (ISymbol, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.ValueAccess
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

```csharp
Object CreateValue(
    ISymbol symbol,
    ReadOnlyMemory sourceData,
    void parent,
    byte timeStamp
)
```

Parameters

- **symbol**
  Type: TwinCAT.TypeSystem.ISymbol

- **sourceData**
  Type: ReadOnlyMemory

- **parent**
  Type: System.Void

- **timeStamp**
  Type: System.Byte

Return Value

Type: Object

Reference

- [IAccessorValueFactory Interface](#)

CreateValue Overload

TwinCAT.ValueAccess Namespace

**IAccessorValueFactory.CreateValue Method (ISymbol, ReadOnlyMemory, Void, Byte)**

Namespace: TwinCAT.ValueAccess

Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Return Value
Type: Object

Reference
IAccessorValueFactory Interface [2552]
CreateValue Overload [2554]
TwinCAT.ValueAccess Namespace [2540]

6.13.4 ResultAccess Class
Result class for an asynchronous access operation.

Inheritance Hierarchy
System.Object
TwinCAT.ValueAccess.ResultAccess
  TwinCAT.ValueAccess.ResultWriteAccess [2575]

Namespace: TwinCAT.ValueAccess [2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax
C#
public class ResultAccess

The ResultAccess type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultAccess(Int32)</td>
<td>Initializes a new instance of the ResultAccess class.</td>
</tr>
<tr>
<td>ResultAccess(Int32, DateTimeOffset)</td>
<td>Initializes a new instance of the ResultReadValueAccess struct.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime</td>
<td>The Timestamp / the date time of the communication access.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The ErrorCode of the communication access</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether this ResultAccess represents a failed access.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether this ResultAccess represents a succeeded access.</td>
</tr>
</tbody>
</table>
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td><code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>SetError [2561]</td>
<td>Sets the error to this ResultAccess object.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from</td>
</tr>
<tr>
<td></td>
<td><code>Object</code>.)</td>
</tr>
</tbody>
</table>

Remarks

This result is independent of any used protocol and is used by the different protocol providers (ADS, OPC, IOT etc.)

Reference

TwinCAT.ValueAccess Namespace [2540]

6.13.4.1 ResultAccess Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[2557]</td>
<td></td>
</tr>
<tr>
<td>DateTimeOffset)</td>
<td>[2558]</td>
</tr>
</tbody>
</table>

Reference

ResultAccess Class [2556]

TwinCAT.ValueAccess Namespace [2540]

6.13.4.1.1 ResultAccess Constructor (Int32)

Initializes a new instance of the `ResultAccess` [2556] class.

Namespace: TwinCAT.ValueAccess [2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

C#

public ResultAccess(
    int errorResult
)

Parameters

errorResult Type: System.Int32
The result code of the communication access..

Reference

ResultAccess Class [2556]
ResultAccess Overload [2557]
TwinCAT.ValueAccess Namespace [2540]

6.13.4.1.2 ResultAccess Constructor (Int32, DateTimeOffset)


Namespace: TwinCAT.ValueAccess [2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

public ResultAccess(
    int errorResult,
    DateTimeOffset time
)

Parameters

errorResult Type: System.Int32
The result code of the operation.

time Type: System.DateTimeOffset
The timestamp of the operation.

Reference

ResultAccess Class [2556]
ResultAccess Overload [2557]
TwinCAT.ValueAccess Namespace [2540]

6.13.4.2 ResultAccess Properties

The ResultAccess [2556] type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime</td>
<td>The Timestamp / the date time of the communication access.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The ErrorCode of the communication access</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether this ResultAccess represents a failed access.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether this ResultAccess represents a succeeded access.</td>
</tr>
</tbody>
</table>

## Reference

- ResultAccess Class [2556]
- TwinCAT.ValueAccess Namespace [2540]

### 6.13.4.2.1 ResultAccess.DateTime Property

The Timestamp / the date time of the communication access.

**Namespace:** TwinCAT.ValueAccess [2540]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public DateTimeOffset DateTime { get; }
```

**Property Value**

Type: DateTimeOffset

**Reference**

- ResultAccess Class [2556]
- TwinCAT.ValueAccess Namespace [2540]

### 6.13.4.2.2 ResultAccess.ErrorCode Property

The ErrorCode of the communication access

**Namespace:** TwinCAT.ValueAccess [2540]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public int ErrorCode { get; }
```

**Property Value**

Type: Int32
Remarks
A value of '0' represents success.

Reference
ResultAccess Class [2556]
TwinCAT.ValueAccess Namespace [2540]

6.13.4.2.3  ResultAccess.Failed Property

Gets a value indicating whether this ResultAccess [2556] represents a failed access.

Namespace:  TwinCAT.ValueAccess [2540]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool Failed { get; }

Property Value

Type: Boolean
true if failed; otherwise, false.

Reference
ResultAccess Class [2556]
TwinCAT.ValueAccess Namespace [2540]

6.13.4.2.4  ResultAccess.Succeeded Property

Gets a value indicating whether this ResultAccess [2556] represents a succeeded access.

Namespace:  TwinCAT.ValueAccess [2540]
Assembly:  TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#
public bool Succeeded { get; }

Property Value

Type: Boolean
true if succeeded; otherwise, false.

Reference
ResultAccess Class [2556]
TwinCAT.ValueAccess Namespace [2540]
6.13.4.3 ResultAccess Methods

The ResultAccess [2556] type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td><code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>SetError</code> [2561]</td>
<td>Sets the error to this ResultAccess [2556] object.</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

Reference

ResultAccess Class [2556]

TwinCAT.ValueAccess Namespace [2540]

6.13.4.3.1 ResultAccess.SetError Method

Sets the error to this ResultAccess [2556] object.

**Namespace**: TwinCAT.ValueAccess [2540]

**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public void SetError(
    int errorCode
)
```

**Parameters**

<table>
<thead>
<tr>
<th>errorCode</th>
<th>Type: System.Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The error code.</td>
</tr>
</tbody>
</table>

Reference

ResultAccess Class [2556]

TwinCAT.ValueAccess Namespace [2540]
6.13.5 **ResultReadDynamicValueAccess Class**

Asynchronous read access result returning an Dynamic value object (IDynamicValue [2011]). Implements the ResultReadValueAccess.T. [2568]

**Inheritance Hierarchy**

```
System.Object
   TwinCAT.ValueAccess.ResultAccess [2556]
         TwinCAT.ValueAccess.ResultReadDynamicValueAccess
```

Namespace: TwinCAT.ValueAccess
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public class ResultReadDynamicValueAccess : ResultReadValueAccess<IDynamicValue>
```

The ResultReadDynamicValueAccess type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime</td>
<td>The Timestamp / the date time of the communication access. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The ErrorCode of the communication access (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether this ResultAccess [2556] represents a failed access. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether this ResultAccess [2556] represents a succeeded access. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>Value</td>
<td>The value (Inherited from ResultReadValueAccess.T. [2568].)</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error to this ResultAccess [2556] object. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
6.13.5.1 ResultReadDynamicValueAccess Properties

The ResultReadDynamicValueAccess type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime</td>
<td>The Timestamp / the date time of the communication access. (Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The ErrorCode of the communication access (Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether this ResultAccess represents a failed access. (Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether this ResultAccess represents a succeeded access. (Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>Value</td>
<td>The value (Inherited from ResultReadValueAccess.T.)</td>
</tr>
</tbody>
</table>

6.13.5.2 ResultReadDynamicValueAccess Methods

The ResultReadDynamicValueAccess type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error to this ResultAccess object. (Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
6.13.6 ResultReadRawAccess Class

Asynchronous read access result object, reading raw (byte[]) data into memory locations. Implements the ResultReadValueAccess.T. [2568]

Inheritance Hierarchy

System.Object
  TwinCAT.ValueAccess.ResultAccess [2556]
    TwinCAT.ValueAccess.ResultReadRawAccess

Namespace: TwinCAT.ValueAccess [2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class ResultReadRawAccess : ResultReadValueAccess<Memory>
```

The ResultReadRawAccess type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime [2559]</td>
<td>The Timestamp / the date time of the communication access. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>ErrorCode [2559]</td>
<td>The ErrorCode of the communication access (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>Failed [2560]</td>
<td>Gets a value indicating whether this ResultAccess [2556] represents a failed access. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>Succeeded [2560]</td>
<td>Gets a value indicating whether this ResultAccess [2556] represents a succeeded access. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>Value [2570]</td>
<td>The value (Inherited from ResultReadValueAccess.T. [2568].)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SetError [2561]</td>
<td>Sets the error to this ResultAccess [2556] object. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Reference**

TwinCAT.ValueAccess Namespace [2540]


### 6.13.6.1 ResultReadRawAccess Properties

The ResultReadRawAccess [2564] type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime [2559]</td>
<td>The Timestamp / the date time of the communication access. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>ErrorCode [2559]</td>
<td>The ErrorCode of the communication access (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>Failed [2560]</td>
<td>Gets a value indicating whether this ResultAccess [2556] represents a failed access. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>Succeeded [2560]</td>
<td>Gets a value indicating whether this ResultAccess [2556] represents a succeeded access. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>Value [2570]</td>
<td>The value (Inherited from ResultReadValueAccess.T. [2568].)</td>
</tr>
</tbody>
</table>

**Reference**

ResultReadRawAccess Class [2564]

TwinCAT.ValueAccess Namespace [2540]

### 6.13.6.1.1 ResultReadRawAccess.Empty Property

Gets the empty ResultAnyValue [998] object.

**Namespace:** TwinCAT.ValueAccess [2540]

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

C#

```csharp
public static ResultReadRawAccess Empty { get; }
```
Property Value

Type: ResultReadRawAccess
The empty / unprocessed result.

Reference

ResultReadRawAccess Class
TwinCAT.ValueAccess Namespace

6.13.6.2 ResultReadRawAccess Methods

The ResultReadRawAccess type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td>gethashcode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>gettype</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>memberwiseclone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>seterror</td>
<td>Sets the error to this ResultAccess object.</td>
</tr>
<tr>
<td>tostring</td>
<td>Returns a string that represents the current object.</td>
</tr>
</tbody>
</table>

Reference

ResultReadRawAccess Class
TwinCAT.ValueAccess Namespace

6.13.7 ResultReadValueAccess Class

Asynchronous read access result returning an untyped Value object type 'object'. Implements the ResultAccess

Inheritance Hierarchy

System.Object
TwinCAT.ValueAccess.ResultAccess
TwinCAT.ValueAccess.ResultReadValueAccess
TwinCAT.ValueAccess.ResultRpcMethodAccess

Namespace: TwinCAT.ValueAccess
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
The ResultReadValueAccess type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime</td>
<td>The Timestamp / the date time of the communication access. (Inherited from ResultAccess)</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The ErrorCode of the communication access (Inherited from ResultAccess)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether this ResultAccess represents a failed access. (Inherited from ResultAccess)</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether this ResultAccess represents a succeeded access. (Inherited from ResultAccess)</td>
</tr>
<tr>
<td>Value</td>
<td>The value (Inherited from ResultReadValueAccess.T)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error to this ResultAccess object. (Inherited from ResultAccess)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.ValueAccess Namespace

TwinCAT.ValueAccess.ResultAccess

### 6.13.7.1 ResultReadValueAccess Properties

The ResultReadValueAccess type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime</td>
<td>The Timestamp / the date time of the communication access. (Inherited from ResultAccess)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The ErrorCode of the communication access (Inherited from ResultAccess)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether this ResultAccess represents a failed access.</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether this ResultAccess represents a succeeded access.</td>
</tr>
<tr>
<td>Value</td>
<td>The value (Inherited from ResultReadValueAccess.T.)</td>
</tr>
</tbody>
</table>

### Reference

ResultReadValueAccess Class [2556]

TwinCAT.ValueAccess Namespace [2540]

### 6.13.7.2 ResultReadValueAccess Methods

The ResultReadValueAccess type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error to this ResultAccess object.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
</tbody>
</table>

### Reference

ResultReadValueAccess Class [2556]

TwinCAT.ValueAccess Namespace [2540]

### 6.13.8 ResultReadValueAccess.T. Class

Result object of an asynchronous read of a specific value of type T. Implements the ResultAccess.
TwinCAT.ValueAccess Namespace \[2540\]
TwinCAT.ValueAccess.ResultReadValueAccess \[2556\]

Namespace: TwinCAT.ValueAccess \[2540\]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public class ResultReadValueAccess<T> : ResultAccess
```

Type Parameters

- **T**: The type of the value.

The `ResultReadValueAccess<T>` type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime [2559]</td>
<td>The Timestamp / the date time of the communication access. (Inherited from <code>ResultAccess \[2556\].</code>)</td>
</tr>
<tr>
<td>ErrorCode [2559]</td>
<td>The ErrorCode of the communication access (Inherited from <code>ResultAccess \[2556\].</code>)</td>
</tr>
<tr>
<td>Failed [2560]</td>
<td>Gets a value indicating whether this <code>ResultAccess \[2556\]</code> represents a failed access. (Inherited from <code>ResultAccess \[2556\].</code>)</td>
</tr>
<tr>
<td>Succeeded [2560]</td>
<td>Gets a value indicating whether this <code>ResultAccess \[2556\]</code> represents a succeeded access. (Inherited from <code>ResultAccess \[2556\].</code>)</td>
</tr>
<tr>
<td>Value [2570]</td>
<td>The value</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>SetError [2561]</td>
<td>Sets the error to this <code>ResultAccess \[2556\]</code> object. (Inherited from <code>ResultAccess \[2556\]</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

Reference

- TwinCAT.ValueAccess Namespace \[2540\]
- TwinCAT.ValueAccess.ResultAccess \[2556\]
6.1.8.1 ResultReadValueAccess.T. Properties

The `ResultReadValueAccess.T. [2568]` generic type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime [2559]</td>
<td>The Timestamp / the date time of the communication access. (Inherited from <code>ResultAccess [2556]</code>.)</td>
</tr>
<tr>
<td>ErrorCode [2559]</td>
<td>The ErrorCode of the communication access (Inherited from <code>ResultAccess [2556]</code>.)</td>
</tr>
<tr>
<td>Failed [2560]</td>
<td>Gets a value indicating whether this <code>ResultAccess [2556]</code> represents a failed access. (Inherited from <code>ResultAccess [2556]</code>.)</td>
</tr>
<tr>
<td>Succeeded [2560]</td>
<td>Gets a value indicating whether this <code>ResultAccess [2556]</code> represents a succeeded access. (Inherited from <code>ResultAccess [2556]</code>.)</td>
</tr>
<tr>
<td>Value [2570]</td>
<td>The value</td>
</tr>
</tbody>
</table>

### Reference

- `ResultReadValueAccess.T.Class [2568]`
- `TwinCAT.ValueAccess Namespace [2540]`

### 6.1.8.1.1 ResultReadValueAccess.T..Value Property

The value

**Namespace:** `TwinCAT.ValueAccess [2540]`

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public T Value { get; }
```

**Property Value**

Type: `T [2568]`

### Reference

- `ResultReadValueAccess.T.Class [2568]`
- `TwinCAT.ValueAccess Namespace [2540]`

### 6.1.8.2 ResultReadValueAccess.T. Methods

The `ResultReadValueAccess.T. [2568]` generic type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>SetError</code></td>
<td>Sets the error to this <code>ResultAccess</code> object. (Inherited from <code>ResultAccess</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

## Reference

- `ResultReadValueAccess.T.Class` [2568]
- `TwinCAT.ValueAccess Namespace` [2540]

### 6.13.9 ResultRpcMethodAccess Class

Asynchronous Invoke RPC Method result class. Implements the `ResultReadValueAccess` [2566]

## Inheritance Hierarchy

- `System.Object`
  - `TwinCAT.ValueAccess.ResultAccess` [2556]
    - `TwinCAT.ValueAccess.ResultReadValueAccess` [2568]
      - `Object`

**Namespace:** `TwinCAT.ValueAccess` [2540]

**Assembly:** `TwinCAT.Ads.Abstractions` (in `TwinCAT.Ads.Abstractions.dll`) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

## Syntax

**C#**

```csharp
public class ResultRpcMethodAccess : ResultReadValueAccess
```

The `ResultRpcMethodAccess` type exposes the following members.

## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ResultRpcMethodAccess</code></td>
<td>Initializes a new instance of the <code>ResultRpcMethod</code> struct. [1025]</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime</td>
<td>The Timestamp / the date time of the communication access. (Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The ErrorCode of the communication access (Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether this ResultAccess represents a failed access. (Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>OutParameters</td>
<td>Gets the out parameters.</td>
</tr>
<tr>
<td>ReturnValue</td>
<td>Gets the return value of theRpcMethod (optionally).</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether this ResultAccess represents a succeeded access. (Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>Value</td>
<td>The value (Inherited from ResultReadValueAccess.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error to this ResultAccess object. (Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Reference

TwinCAT.ValueAccess Namespace

TwinCAT.ValueAccess.ResultReadValueAccess

**6.13.9.1 ResultRpcMethodAccess Constructor**

Initializes a new instance of the ResultRpcMethod struct.

**Namespace:** TwinCAT.ValueAccess

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14
Syntax

**C#**

```csharp
public ResultRpcMethodAccess(
    Object returnValue,
    Object[] outParameters,
    int errorCode,
    DateTimeOffset timeStamp
)
```

**Parameters**

- **returnValue**
  - Type: `System.Object`
  - The value.

- **outParameters**
  - Type: `System.Object`
  - The out parameters.

- **errorCode**
  - Type: `System.Int32`
  - The error code.

- **timeStamp**
  - Type: `System.DateTimeOffset`
  - The time stamp.

**Reference**

- `ResultRpcMethodAccess Class [2571]`
- `TwinCAT.ValueAccess Namespace [2540]`

### 6.13.9.2 ResultRpcMethodAccess Properties

The `ResultRpcMethodAccess [2571]` type exposes the following members.

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DateTime [2559]</strong></td>
<td>The Timestamp / the date time of the communication access. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td><strong>ErrorCode [2559]</strong></td>
<td>The ErrorCode of the communication access (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td><strong>Failed [2560]</strong></td>
<td>Gets a value indicating whether this ResultAccess [2556] represents a failed access. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td><strong>OutParameters [2574]</strong></td>
<td>Gets the out parameters.</td>
</tr>
<tr>
<td><strong>ReturnValue [2574]</strong></td>
<td>Gets the return value of the RpcMethod (optionally).</td>
</tr>
<tr>
<td><strong>Succeeded [2560]</strong></td>
<td>Gets a value indicating whether this ResultAccess [2556] represents a succeeded access. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td><strong>Value [2570]</strong></td>
<td>The value (Inherited from ResultReadValueAccess.T. [2568].)</td>
</tr>
</tbody>
</table>

**Reference**

- `ResultRpcMethodAccess Class [2571]`
- `TwinCAT.ValueAccess Namespace [2540]`
6.13.9.2.1 ResultRpcMethodAccess.OutParameters Property

Gets the out parameters.

**Namespace:** TwinCAT.ValueAccess

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
public Object[] OutParameters { get; }
```

**Property Value**

Type: Object

The out parameters.

**Reference**

ResultRpcMethodAccess Class

TwinCAT.ValueAccess Namespace

6.13.9.2.2 ResultRpcMethodAccess.ReturnValue Property

Gets the return value of the RpcMethod (optionally).

**Namespace:** TwinCAT.ValueAccess

**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fdca3e72bc0ea15da1c14

**Syntax**

```csharp
public Object ReturnValue { get; }
```

**Property Value**

Type: Object

The return value or NULL

**Reference**

ResultRpcMethodAccess Class

TwinCAT.ValueAccess Namespace

6.13.9.3 ResultRpcMethodAccess Methods

The ResultRpcMethodAccess Class type exposes the following members.

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## TwinCAT.Ads Namespaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error to this <code>ResultAccess</code> object. (Inherited from <code>ResultAccess</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

### Reference

- ResultRpcMethodAccess Class [2571]
- TwinCAT.ValueAccess Namespace [2540]

## 6.13.10 ResultWriteAccess Class

Asynchronous write access result class.

### Inheritance Hierarchy

- System.Object
  - TwinCAT.ValueAccess.ResultAccess [2556]
    - TwinCAT.ValueAccess.ResultWriteAccess

### Namespace: TwinCAT.ValueAccess [2540]

### Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

### Syntax

**C#**

```
public class ResultWriteAccess : ResultAccess
```

The `ResultWriteAccess` type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultWriteAccess(IInt32)</td>
<td>Initializes a new instance of the <code>ResultWriteAccess</code> class.</td>
</tr>
<tr>
<td>ResultWriteAccess(IInt32, DateTimeOffset)</td>
<td>Initializes a new instance of the <code>ResultWriteAccess</code> struct.</td>
</tr>
</tbody>
</table>
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime</td>
<td>The Timestamp / the date time of the communication access. (Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>Empty</td>
<td>Gets the empty ResultWriteAccess object.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The ErrorCode of the communication access (Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>Failed</td>
<td>Gets a value indicating whether this ResultAccess represents a failed access. (Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>Succeeded</td>
<td>Gets a value indicating whether this ResultAccess represents a succeeded access. (Inherited from ResultAccess.)</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>SetLastError</td>
<td>Sets the error to this ResultAccess object. (Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

Reference

TwinCAT.ValueAccess Namespace

6.13.10.1 ResultWriteAccess Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultWriteAccess(int32)</td>
<td>Initializes a new instance of the ResultWriteAccess class.</td>
</tr>
<tr>
<td>ResultWriteAccess(int32, DateTimeOffset)</td>
<td>Initializes a new instance of the ResultWriteAccess struct.</td>
</tr>
</tbody>
</table>
6.13.10.1.1 ResultWriteAccess Constructor (Int32)

Initializes a new instance of the `ResultWriteAccess` class.

**Namespace**: TwinCAT.ValueAccess

**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public ResultWriteAccess(int errorResult)
```

**Parameters**

- `errorResult`  
  Type: `System.Int32`  
  The result code of the communication access.

Reference

- ResultWriteAccess Class [2575]
- TwinCAT.ValueAccess Namespace [2540]

6.13.10.1.2 ResultWriteAccess Constructor (Int32, DateTimeOffset)

Initializes a new instance of the `ResultWriteAccess` struct.

**Namespace**: TwinCAT.ValueAccess

**Assembly**: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

**Syntax**

```csharp
public ResultWriteAccess(int errorResult, DateTimeOffset timeStamp)
```

**Parameters**

- `errorResult`  
  Type: `System.Int32`  
  The error result.
- `timeStamp`  
  Type: `System.DateTimeOffset`  
  The time stamp.
6.13.10.2 ResultWriteAccess Properties

The ResultWriteAccess [2575] type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime [2559]</td>
<td>The Timestamp / the date time of the communication access. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>ErrorCode [2559]</td>
<td>The ErrorCode of the communication access (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>Failed [2560]</td>
<td>Gets a value indicating whether this ResultAccess [2556] represents a failed access. (Inherited from ResultAccess [2556].)</td>
</tr>
<tr>
<td>Succeeded [2560]</td>
<td>Gets a value indicating whether this ResultAccess [2556] represents a succeeded access. (Inherited from ResultAccess [2556].)</td>
</tr>
</tbody>
</table>

6.13.10.2.1 ResultWriteAccess.Empty Property

Gets the empty ResultWriteAccess [2575] object.

Namespace: TwinCAT.ValueAccess [2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
public static ResultWriteAccess Empty { get; }
```

Property Value

Type: ResultWriteAccess [2575]
The empty / unprocessed result.

Reference

ResultWriteAccess Class [2575]
6.13.10.3  ResultWriteAccess Methods

The ResultWriteAccess type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection. (Inherited from</td>
</tr>
<tr>
<td></td>
<td>Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>SetError</td>
<td>Sets the error to this ResultAccess object. (Inherited from ResultAccess)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from ResultAccess.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object)</td>
</tr>
</tbody>
</table>

Reference

ResultWriteAccess Class

TwinCAT.ValueAccess Namespace

6.13.11  SymbolNotificationTypes Enumeration

Specifies the Notification type of ADS Notifications

Namespace: TwinCAT.ValueAccess
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```csharp
[FlagsAttribute]
public enum SymbolNotificationTypes
```

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>None / Uninitialized</td>
</tr>
<tr>
<td>Value</td>
<td>1</td>
<td>ValueChanged notifications</td>
</tr>
<tr>
<td>RawValue</td>
<td>2</td>
<td>RawValueChanged notifications</td>
</tr>
<tr>
<td>Both</td>
<td>3</td>
<td>ValueChanged + RawValueChanged notifications</td>
</tr>
</tbody>
</table>
6.13.12 ValueCreationModes Enumeration

Creation mode for Values

**Namespace:** TwinCAT.ValueAccess Namespace 
**Assembly:** TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0e15da1c14

**Syntax**

C#

```csharp
[FlagsAttribute]
public enum ValueCreationModes
```

### Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>None / Uninitialized Mask</td>
</tr>
<tr>
<td>Primitives</td>
<td>1</td>
<td>Convert to .NET Primitives, wherever possible.</td>
</tr>
<tr>
<td>Enums</td>
<td>2</td>
<td>Use IEnumValue on EnumTypes instead of .NET Primitives</td>
</tr>
<tr>
<td>FullDynamics</td>
<td>4</td>
<td>Wraps all Primitives also in IValue Objects</td>
</tr>
<tr>
<td>PlcOpenTypes</td>
<td>8</td>
<td>Use PlcOpen Times (TIME, LTIME, DT, TOD, DATETIME) instead of .NET Primitives. They won't be wrapped into</td>
</tr>
<tr>
<td>Default</td>
<td>1</td>
<td>Default settings for the value creation mode (Translate to Primitives)</td>
</tr>
</tbody>
</table>

**Remarks**

This setting is used by the ValueFactory/ to create Read values. In default primitive mode all values will be transferred to Primitive .NET Symbols if possible. E.g PlcOpen.TIME --> TimeSpan, IEnumValue --> .NET Primitives. They won't be wrapped into

**Reference**

TwinCAT.ValueAccess Namespace [2540]
More Information:
www.beckhoff.com/tc1000