# Table of contents

1 Foreword ................................................................................................................................. 5
   1.1 Notes on the documentation ............................................................................................ 5
   1.2 Safety instructions ............................................................................................................. 6
   1.3 Notes on information security ............................................................................................ 7

2 Overview .................................................................................................................................. 8

3 Requirements ........................................................................................................................... 9

4 Installation and Activation ...................................................................................................... 10
   4.1 Installation by PackageManager (PowershellGallery, Internet access necessary) ........... 10
   4.2 Manual Installation without Internet access ................................................................. 11
   4.3 Check succeeded installation .......................................................................................... 12

5 TcXaeMgmt Version 3.X ........................................................................................................ 13
   5.1 TcXaeMgmt Module ........................................................................................................ 13
   5.2 Add-AdsNatRoute ........................................................................................................... 18
   5.3 Add-AdsRoute ................................................................................................................ 21
   5.4 Add-MqttRoute ................................................................................................................. 28
   5.5 Close-TcSession ............................................................................................................. 32
   5.6 Copy-AdsFile .................................................................................................................. 33
   5.7 Get-AdsRoute ................................................................................................................ 36
   5.8 Get-AdsState .................................................................................................................. 39
   5.9 Get-TcDataType ............................................................................................................... 42
   5.10 Get-TcLicense ............................................................................................................... 45
   5.11 Get-TcRouterInfo ........................................................................................................... 49
   5.12 Get-TcSession ................................................................................................................. 52
   5.13 Get-TcSymbol ................................................................................................................ 53
   5.14 Get-TcTargetInfo ........................................................................................................... 57
   5.15 Get-TcVersion ................................................................................................................. 60
   5.16 New-TcSession .............................................................................................................. 62
   5.17 Read-TcValue ................................................................................................................. 65
   5.18 Register-AdsHandle ....................................................................................................... 72
   5.19 Register-AdsNatRoute ................................................................................................... 75
   5.20 Remove-AdsNatRoute .................................................................................................... 77
   5.21 Remove-AdsRoute ......................................................................................................... 79
   5.22 Remove-MqttRoute ....................................................................................................... 82
   5.23 Send-TcReadWrite ......................................................................................................... 84
   5.24 Set-AdsState ................................................................................................................ 90
   5.25 Test-AdsRoute .............................................................................................................. 93
   5.26 Unregister-AdsHandle ................................................................................................. 96
   5.27 Write-TcValue .............................................................................................................. 100

6 TcXaeMgmt Version 5.X ........................................................................................................ 107
   6.1 About TcXaeMgmt ........................................................................................................... 107
   6.2 Add-AdsRoute .............................................................................................................. 115
   6.3 Add-MqttRoute ............................................................................................................. 124
   6.4 Close-TcSession ........................................................................................................... 128
<table>
<thead>
<tr>
<th></th>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.5</td>
<td>Copy-AdsFile</td>
<td>129</td>
</tr>
<tr>
<td>6.6</td>
<td>Get-AdsRoute</td>
<td>133</td>
</tr>
<tr>
<td>6.7</td>
<td>Get-AdsState</td>
<td>136</td>
</tr>
<tr>
<td>6.8</td>
<td>Get-AmsRouterEndpoint</td>
<td>140</td>
</tr>
<tr>
<td>6.9</td>
<td>Get-EcBoxes</td>
<td>141</td>
</tr>
<tr>
<td>6.10</td>
<td>Get-EcFrameStatistics</td>
<td>142</td>
</tr>
<tr>
<td>6.11</td>
<td>Get-EcMaster</td>
<td>144</td>
</tr>
<tr>
<td>6.12</td>
<td>Get-IoDevice</td>
<td>146</td>
</tr>
<tr>
<td>6.13</td>
<td>Get-IoFreeRun</td>
<td>149</td>
</tr>
<tr>
<td>6.14</td>
<td>Get-MqttRoute</td>
<td>151</td>
</tr>
<tr>
<td>6.15</td>
<td>Get-RTimeCpuSettings</td>
<td>152</td>
</tr>
<tr>
<td>6.16</td>
<td>Get-RTimeLatency</td>
<td>153</td>
</tr>
<tr>
<td>6.17</td>
<td>Get-RTimePerformance</td>
<td>156</td>
</tr>
<tr>
<td>6.18</td>
<td>Get-TcDataType</td>
<td>159</td>
</tr>
<tr>
<td>6.19</td>
<td>Get-TcEvent</td>
<td>162</td>
</tr>
<tr>
<td>6.20</td>
<td>Get-TcLicense</td>
<td>166</td>
</tr>
<tr>
<td>6.21</td>
<td>Get-TcRouterInfo</td>
<td>171</td>
</tr>
<tr>
<td>6.22</td>
<td>Get-TcSession</td>
<td>173</td>
</tr>
<tr>
<td>6.23</td>
<td>Get-TcSymbol</td>
<td>174</td>
</tr>
<tr>
<td>6.24</td>
<td>Get-TcTargetInfo</td>
<td>179</td>
</tr>
<tr>
<td>6.25</td>
<td>Get-TcVersion</td>
<td>182</td>
</tr>
<tr>
<td>6.26</td>
<td>New-TcSession</td>
<td>185</td>
</tr>
<tr>
<td>6.27</td>
<td>Read-TcValue</td>
<td>187</td>
</tr>
<tr>
<td>6.28</td>
<td>Register-AdsHandle</td>
<td>194</td>
</tr>
<tr>
<td>6.29</td>
<td>Register-AdsNatRoute</td>
<td>197</td>
</tr>
<tr>
<td>6.30</td>
<td>Remove-AdsRoute</td>
<td>200</td>
</tr>
<tr>
<td>6.31</td>
<td>Remove-MqttRoute</td>
<td>203</td>
</tr>
<tr>
<td>6.32</td>
<td>Reset-IoFreeRun</td>
<td>205</td>
</tr>
<tr>
<td>6.33</td>
<td>Restart-AdsComputer</td>
<td>208</td>
</tr>
<tr>
<td>6.34</td>
<td>Send-TcReadWrite</td>
<td>212</td>
</tr>
<tr>
<td>6.35</td>
<td>Set-AdsState</td>
<td>218</td>
</tr>
<tr>
<td>6.36</td>
<td>Set-AmsRouterEndpoint</td>
<td>222</td>
</tr>
<tr>
<td>6.37</td>
<td>Set-IoFreeRun</td>
<td>224</td>
</tr>
<tr>
<td>6.38</td>
<td>Stop-AdsComputer</td>
<td>227</td>
</tr>
<tr>
<td>6.39</td>
<td>Test-AdsRoute</td>
<td>231</td>
</tr>
<tr>
<td>6.40</td>
<td>Unregister-AdsHandle</td>
<td>234</td>
</tr>
<tr>
<td>6.41</td>
<td>Write-TcValue</td>
<td>238</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®, TwinCAT/BSD®, TC/BSD®, 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!

**DANGER**

Serious risk of injury!
Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

**WARNING**

Risk of injury!
Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

**CAUTION**

Personal injuries!
Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

**NOTE**

Damage to the environment or devices
Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

Tip or pointer

This symbol indicates information that contributes to better understanding.
1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.
2 Overview

What is Powershell

Excerpt from Wikipedia: “PowerShell is a task automation and configuration management framework from Microsoft, consisting of a command-line shell and associated scripting language built on the .NET Framework and .NET Core.”

“In PowerShell, administrative tasks are generally performed by cmdlets (pronounced command-lets), which are specialized .NET classes implementing a particular operation. Sets of cmdlets may be combined into scripts, executables (which are standalone applications), or by instantiating regular .NET classes (or WMI/COM Objects). These work by accessing data in different data stores, like the file system or registry, which are made available to the PowerShell runtime via PowerShell providers.” (link)

These Cmdlets are packaged and deployed in so called Powershell Modules.

What is the Powershell Ads Module (Extension) named ‘TcXaeMgmt’

The so called TcXaeMgmt module contains a number of useful Cmdlets and Providers for TwinCAT Management/Administration and accessing Ads Devices natively via Powershell.

- Route Management (Add-AdsRoute, Remove-AdsRoute) and Broadcast search (Get-AdsRoute)
- Testing Route Connections (Test-AdsRoute, Get-AdsState)
- Establish Ads Communication channels via Sessions and Connection (New-TcSession, Close-TcSession)
- Type-Safe Read/Write Value Access via ADS protocol (Read-TcValue, Write-TcValue)
- Symbol and DataType Browsing (Get-TcSymbol, Get-TcDataType)
- Further Administrative Tasks (Copy-AdsFile, Get-TcVersion, etc.)

The package is published via the Powershell Gallery at https://www.powershellgallery.com/packages/TcXaeMgmt/.

Versions

There exist two different series of the ‘TcXaeMgmt’ module:

<table>
<thead>
<tr>
<th>Versions</th>
<th>Powershell Version</th>
<th>TwinCAT Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.X [107]</td>
<td>Windows Powershell 5.1</td>
<td>&gt;= 3.1.4024.10</td>
<td>Platform independent version</td>
</tr>
</tbody>
</table>

Please be aware of the Differences Microsoft Powershell vs. Windows Powershell.
## 3 Requirements

<table>
<thead>
<tr>
<th>Version 5.X</th>
<th>Version 3.2.X</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Microsoft Powershell &gt;= 6.0 or Windows Powershell 5.1</td>
<td>• Windows 7 SP1 and newer</td>
</tr>
<tr>
<td>• Installed TwinCAT 3.1.4024.10 or newer (minimum RT / ADS level)</td>
<td>• Windows Powershell 4.0 and newer</td>
</tr>
<tr>
<td></td>
<td>• .NET Framework 4.5 and newer</td>
</tr>
<tr>
<td></td>
<td>• Installed TwinCAT 3 or TwinCAT 2 (minimum RT / ADS level)</td>
</tr>
</tbody>
</table>
4 Installation and Activation

4.1 Installation by Package Manager (Powershell Gallery, Internet access necessary)

Installation by Package Manager (Powershell Gallery)

For newer versions of Windows and Powershell the most easiest way to install the TcXaeMgmt Module is to use the Powershell Gallery. Powershell Gallery access is available without installation if one of the following setups is already on the system:

Windows 10 or newer
Windows Server 2016 or newer
Windows Management Framework (WMF) 5.0 or newer
PowerShell 6 or newer.

In this case just type

PS> Install-Module -Name TcXaeMgmt

from the Powershell console.

Further information about the package and its installation is available on the Gallery Website: https://www.powershellgallery.com/packages/TcXaeMgmt/

Installation of PowershellGet Module

In all other cases the PowershellGet module must be installed on the machine most likely. On Powershell the availability can be checked with the following command:

PS> get-module PowershellGet -listavailable

Directory: C:\Program Files\WindowsPowerShell\Modules

ModuleType Version Name ExportedCommands
---------- ------- ---- ----------------
Script 2.2.5 PowerShellGet {Find-Command, Find-DSCResource, Find-Module, Find-RoleCapability...}

Please assure, that at minimum Version 2.2.5 is available.

The Powershell version can be determined as follows:

PS> $PSVersionTable

Name Value
---- ----- 
PSVersion 5.0.10514.6
WSManStackVersion 3.0
SerializationVersion 1.1.0.1
CLAVersion 4.0.30319.42000
BuildVersion 10.0.10514.6
PSCompatibleVersions {1.0, 2.0, 3.0, 4.0...}
PSRemotingProtocolVersion 2.3

If the PowershellGet Module is not existing please follow the instructions of the following websites

- https://www.powershellgallery.com/packages/PackageManagement/

Now, when the PowershellGet Module is available, the command

PS> Install-Module -Name TcXaeMgmt
4.2 Manual Installation without Internet access

Because the TwinCAT XAE Management Powershell Module (TcXaeMgmt) is now available on the Powershell Gallery (https://www.powershellgallery.com/packages/TcXaeMgmt/) it is not necessary to activate the Powershell Module manually if Internet access available. For completeness and if no Internet is present the following steps show the manual installation process.

Check Installed Powershell Module

The TwinCAT Installation includes the setup for the Powershell TcXaeMgmt Module. It should already be existing under the folder

[TWINCATINSTALL]/AdsApi/Powershell/TcXaeMgmt

where [TWINCATINSTALL] indicates the TwinCAT root folder (c:\TwinCAT by default).

Check the Powershell Cmdlet Execution policy

What is left actually to the user is to activate that module in the Powershell environment.

PS> get-executionPolicy
Restricted

If the policy is not set to ‘Unrestricted’ or ‘RemoteSigned’, Powershell does not allow to process scripts or Cmdlets. For more information, please see

PS> get-help about_Execution_Policies

If the execution policy is restricted, it has to be set to ‘RemoteSigned’ by a Powershell console with administrative rights:

PS C:\tfs> Set-ExecutionPolicy RemoteSigned
Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about_Execution_Policies help topic at http://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N")?: y

Extend Powershell Module Search Path (PSModulePath)

As next step, the Powershell Search path for Powershell Modules must be extended, so that Powershell can find the TcXaeMgmt Cmdlets in the

[TWINCATINSTALL]/AdsApi/Powershell/

folder.

Please check the Enviroment variable ‘PSModulePath’:

PS> $env:PSModulePath
D:\Users\User\Documents\WindowsPowerShell\Modules\;C:\Program Files\WindowsPowerShell\Modules;C:\Windows\system32\WindowsPowerShell\v1.0\Modules;

If the TwinCAT AdsApi path is not contained in the PSModulePath Variable, it should be set via the Windows Control Panel → System Properties → Advanced → Environment Variables Dialog.

Please add

[TWINCATINSTALLDIR]\AdsApi\Powershell\

e.g with

C:\TwinCAT\AdsApi\Powershell

to the System wide ‘PSModulePath’ Variable. After a Powershell Console restart, the new setting should be available:
4.3 Check succeeded installation

To check that the TcXaeMgmt Module can be loaded successfully and is operational type:

```powershell
PS > get-module TcXaeMgmt -listavailable
```

Now the contained cmdlets can be accessed:

```powershell
PS> get-command -module TcXaeMgmt
```

Help is included for an overview of the features and concepts:

```powershell
PS > get-help about_TcXaeMgmt
```

Or for specific Cmdlet information:

```powershell
PS > get-help Read-TcValue -full
```
5 TcXaeMgmt Version 3.X

Windows Powershell Version of the ‘TcXaeMgmt’ Module. This runs on Windows and Windows Powershell Version >= 4.0.

Supported TwinCAT Versions are all Versions (Twincat 2 and TwinCAT 3) with minimum installation level ADS or RT.

If a TwinCAT Installation >= Version 3.1.4024 is running locally, please use the newer 'TcXaeMgmt' Version 5.X series [107].

5.1 TcXaeMgmt Module

**Topic**
PowerShell TwinCAT XAE Management Console (TcXaeMgmt)

**Short Description**
Describes the Powershell TwinCAT Management Console (TcXaeMgmt) module and how to use the contained cmdlets and functions.

**Long Description**
PowerShell TwinCAT Management Console is a PowerShell module that provides a number of useful cmdlets for TwinCAT System Management and for communicating with ADS devices with the ADS protocol.

This includes TwinCAT Route Management as finding routes (find targets, broadcast search), establishing and removing route connections (Add-AdsRoute, Remove-AdsRoute) and the test of registered routes (Test-AdsRoute) or communication (Get-AdsState).

Furthermore Ads Sessions can be established for further use (New-TcSession), Symbol information can be browsed (Get-TcSymbol, Get-TcDataType) and data read/write from/to to ADS Devices (Read-TcValue, Write-TcValue). Uploading and Downloading files to or from the TwinCAT Target systems is an additional feature (Copy-AdsFile).

**Powershell Compatibility**
Actually, the TwinCAT Management Console is written for Powershell 3.0 and higher.

**Preference Variables**

**Cmdlets**
To see what cmdlets are provided by the TcXaeMgmt Module, execute the command:
Get-Command -Module TcXaeMgmt -CommandType Cmdlet

The actual TcXaeMgmt cmdlets are listed below:

**Add-AdsNatRoute [18]**
Adds an AmsNAT Route to the destination system (obsolete).
**Add-AdsRoute [►21]**
Cmdlet for adding TwinCAT Routes.

**Add-MqttRoute [►28]**
Adds an MQTT route to the destination system.

**Close-TcSession [►32]**
Closes the specified session object.

**Copy-AdsFile [►33]**
Uploads / Downloads files from/to TwinCAT target.

**Get-AdsRoute [►36]**
List routes on a TwinCAT System / Broadcast search.

**Get-AdsState [►39]**
Gets the Ads State of a TwinCAT Target.

**Get-TcDataType [►42]**
Get the DataTypes from a TwinCAT target system / Device.

**Get-TcLicense [►45]**
Get TwinCAT License information.

**Get-TcRouterInfo [►49]**
Gets the router status information of the specified target system.

**Get-TcSession [►52]**
List the currently established Sessions.

**Get-TcSymbol [►53]**
Get the symbols from a TwinCAT target system / Device.

**Get-TcTargetInfo [►57]**
Get TwinCAT Device Target information.

**Get-TcVersion [►60]**
Get the TwinCAT Version of a target system.

**New-TcSession [►62]**
Create a new session to a TwinCAT Target.

**Read-TcValue [►65]**
Reads values from TwinCAT devices.
Register-AdsHandle [\[ 72\] ]
Registers and returns a symbol handle.

Register-AdsNatRoute [\[ 75\] ]
Changes an standard Route to an AmsNAT route on the target system (obsolete).

Remove-AdsNatRoute [\[ 77\] ]
Removes an AmsNAT Route from the destination system (obsolete).

Remove-AdsRoute [\[ 79\] ]
Remove an ADS Route.

Remove-MqttRoute [\[ 82\] ]
Remove a MQTT Route.

Send-TcReadWrite [\[ 84\] ]
Sends a Read/Write request to ADS Server / TwinCAT Devices.

Set-AdsState [\[ 90\] ]
Set the ADS State of a TwinCAT Target.

Test-AdsRoute [\[ 93\] ]
Test the specified route connection.

Unregister-AdsHandle [\[ 96\] ]
Unregisters a symbol handle.

Write-TcValue [\[ 100\] ]
Write values to TwinCAT devices.

First Steps

Getting Route
```powershell
PS> $route = get-adsroute TC3TEST*
PS> $route

Name                      NetId                  Address          Sub Version RTSystem
----                      -----                  -------          ---        -------
TC3TESTA1-CP67X 172.17.62.105.1.1 172.17.62.105 0.0            Unknown
```

Create Session
```powershell
PS> $session = New-TcSession -Route $route -Port 851
PS> $session

ID Address IsConnected EstablishedAt
-- ------- ----------- -------------
1      172.17.62.105.1.1:851 True 12/12/2016 12:22:02 PM
```
Read Ads Value (Struct)

```
PS> $v1 = Read-TcValue -SessionId 1 -Path "GVL.vgStruct"
PS> $v1

vBool   : True
vByte   : 123
vWord   : 12345
vDWord  : 12345678
vSInt   : -121
vUSInt  : 212
vInt    : -12121
vUInt   : 21212
vDInt   : -1212121
vReal   : 123,456
vLReal  : 1234567890,12346
vString : QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;:_
vTime   : 01:02:03.0040000
vTod    : 23:45:06.7890000
vDate   : 17.11.2005 00:00:00
vDT     : 17.11.2005 12:34:56
vAlias  : 8
vEnum   : 8
vRange  : 7
PSValue : ...
```

Read Ads Value (Boolean)

```
PS> $v2 = Read-TcValue -SessionId 1 -Path "Main.bChange"
PS> $v2
False
```

Read Ads Value (Array of Strings)

```
PS> $v3 = Read-TcValue -SessionId 1 -Path "GVL.vgaString"

Dimensions Elements
---------- --------
{TwinCAT.TypeSystem.Dimension} {QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;:_;:;MNBVCXYÄÖLKJHGFDSAÜPOIUZTREWQ}...
```

Read Array Of Structs

```
PS> $v4 = Read-TcValue -SessionId 1 -Path "GVL.vgastruct"

Dimensions Elements
---------- --------
{TwinCAT.TypeSystem.Dimension} (@vBool=True; vByte=123; vWord=12345; vDWord=12345678; vSInt=-121; vUSInt=212; vInt=-12121; vUInt=21212; vDInt=-1212121; vUD...)
```

Dump Array Elements

```
PS> $v4.Dimensions.ElementCount
2

PS> $v4.Elements

vBool   : True
vByte   : 123
vWord   : 12345
vDWord  : 12345678
vSInt   : -121
vUSInt  : 212
vInt    : -12121
vUInt   : 21212
vDInt   : -1212121
```

Version: 1.2
Browse Data Types (Query by Category)

PS> $session | Get-TcDataType | where Category -eq "Array"

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Category</th>
<th>Comment</th>
<th>ElementType</th>
<th>Dimensions</th>
<th>Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARRAY [-1..1] OF INT</td>
<td>6</td>
<td>Array</td>
<td></td>
<td>INT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARRAY [-10..-8] OF BOOL</td>
<td>3</td>
<td>Array</td>
<td></td>
<td>BOOL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARRAY [0..1] OF A_Alias</td>
<td>4</td>
<td>Array</td>
<td></td>
<td>A_Alias</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Browse DataTypes by name

PS> $session | Get-TcDataType -name "Array*"

Browse all Symbols recursively

PS> $session | Get-TcSymbol -recurse

... returns all symbols

Browse Symbols recursively by Symbol Path

Here specific array index 'TaskInfo[1]'

PS> $session | Get-TcSymbol -recurse -path "**TaskInfo\`{1}`**,"*.ProjectName"

<table>
<thead>
<tr>
<th>InstanceName</th>
<th>Comment</th>
<th>DataType</th>
<th>Size</th>
<th>InstancePath</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProjectName</td>
<td>STRING(63)</td>
<td>64</td>
<td></td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ProjectName</td>
</tr>
<tr>
<td>ObjId</td>
<td>OTCID</td>
<td>4</td>
<td></td>
<td>TwinCAT_SystemInfoVarList._TaskInfo[1].ObjId</td>
</tr>
<tr>
<td>CycleTime</td>
<td>UDINT</td>
<td>4</td>
<td></td>
<td>TwinCAT_SystemInfoVarList._TaskInfo[1].CycleTime</td>
</tr>
</tbody>
</table>
Browse only Symbols ending with path *.ProjectName

```powershell
PS> $project = Get-TcSymbol -Session $session -recurse -path "*.ProjectName"
```

<table>
<thead>
<tr>
<th>InstanceName</th>
<th>DataType</th>
<th>Size</th>
<th>InstancePath</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProjectName</td>
<td>STRING(63)</td>
<td>64</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ProjectName</td>
<td></td>
</tr>
</tbody>
</table>

**Ads Read ProjectName**

```powershell
PS> $project | Read-TcValue -Session $session
```

**Ads Write ProjectName**

```powershell
PS> $project | Write-TcValue -Session $session -Value "NewProjectName"
PS> $project | Read-TcValue -Session $session
```

**ReadWrite by Symbol Path**

```powershell
PS> Read-TcValue -SessionId 1 -Path "Main.bChange"
false

PS> Write-TcValue -SessionId 1 -Symbol "Main.bChange" -Value True
PS> Read-TcValue -SessionId 1 -Path "GVL.vgBool"
PS> Write-TcValue -SessionId 1 -Path "GVL.vgBool" -value $true
```

**ReadWrite by Piping**

```powershell
PS> $projectNameSymbol = $session | Get-TcSymbol -Recurse -path "*ProjectName"
PS> $projectNameSymbol | Read-TcValue -SessionId 1
PS> $projectNameSymbol | Write-TcValue -SessionId 1 -Value "NewProjectName"
PS> $projectNameSymbol | Read-TcValue -SessionId 1
```

## 5.2 Add-AdsNatRoute

**SYNOPSIS**

Adds an AmsNAT Route to the destination system (obsolete).

**SYNTAX**

```
Add-AdsNatRoute [-Name] <String> -Address <String> -NetId <AmsNetId> -NATNetId <AmsNetId> [-Destination <String>] [-WhatIf] [-Confirm] [CommonParameters]
```
DESCRIPTION

This Cmdlet adds an AmsNAT Route to the destination system.

It writes just the specified content to the StaticRoutes.xml and needs a TwinCAT Restart afterwards.

Please be aware that the Route is not added bidirectionally.

For TwinCAT Versions 1>= 3.1.4024.11 (or newer), the Add-AdsRoute Cmdlet should be used with the -NAT Parameter as Replacement.

Therefore, this 'Add-AdsNatRoute' Cmdlet is classified as 'obsolete' and of limited use and could be removed in future.

EXAMPLES

EXAMPLE 1

PS> Add-AdsNatRoute -Name MyRoute -NetId 1.2.3.4.1.1 -Address 1.2.3.4 -NATNetId 1.2.3.4.2.2

Adds a Route 'MyRoute' with RemoteNetId '1.2.3.4.1.1' and IPAddress '1.2.3.4' with AmsNAT translation '1.2.3.4.2.2' to the local system.

EXAMPLE 2

PS> Add-AdsNatRoute -Name Client01 -NetId 1.2.3.4.1.1 -Address Client01 -NATNetId 1.2.3.4.2.2 -Destination CX_1234

Adds a Route 'Client01' with RemoteNetId '1.2.3.4.1.1' and DnsName 'Client01' to 'CX_1234' with AmsNAT translation '1.2.3.4.2.2'.

PARAMETERS

-Address

The Address of the system to be added as route.

This can be an IPAddress or HostName.

  Type: String
  Parameter Sets: (All)
  Aliases:
  Required: True
  Position: Named
  Default value: None
  Accept pipeline input: False
  Accept wildcard characters: False

-Destination

The Destination Address, where the AmsNAT Route is added.

This Parameter allows RouteName, AmsNetId, IPAddress or HostName

  Type: String
  Parameter Sets: (All)
  Aliases:
  Required: False
  Position: Named
  Default value: None
  Accept pipeline input: False
  Accept wildcard characters: False
-Name

The Name of the AmsNAT Route to add.

Type: String
Parameter Sets: (All)
Aliases:

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-NATNetId

The NATNetId.

This is the NetId that is used as NetId Translation on the local/destination system.

Type: AmsNetId
Parameter Sets: (All)
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-NetId

The NetId of the system to be added as Route.

This will be the RemoteNetId.

Type: AmsNetId
Parameter Sets: (All)
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Confirm

Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
5.3 Add-AdsRoute

SYNOPSIS

Cmdlet for adding TwinCAT Routes.

SYNTAX

Address (Default)


AddressPSK


AddressSCA


AddressSSC


NetId


NetIdPSK

**DESCRIPTION**

Adds a Route to the destination target System (Temporary or static).

**EXAMPLES**

**EXAMPLE 1**

```powershell
PS> Get-AdsRoute -All -name "Tc3*"
Name            NetId            Address          Sub TcVersion RTSystem
----            -----            -------          --------       -------
TC3TestA1-CP67x 172.17.62.105.1.1 172.17.62.105 3.1.4021       Win7
TC3Test13-C6650 172.17.60.239.1.1 172.17.62.156 2.11.2246       Win7

PS> $cred = Get-Credential -Message "Get Credentials" -UserName "UserName"

PS> Add-AdsRoute -Credential $cred -name "TC3TestA1-CP67x" -temporary

Name            NetId            Address          Sub TcVersion RTSystem
----            -----            -------          --------       -------
TC3TestA1-CP67x 172.17.62.105.1.1 172.17.62.105 3.1.4021       Win7

PS> Get-AdsRoute -name "TC3TestA1-CP67x" | Test-AdsRoute
```
Search for Systems that start with the name "TC3*", then asks the user for Credentials and adds the Route as 'temporary' (with TC2 compatible security, clear text password).

Afterwards, the connection is checked.

**EXAMPLE 2**

```
PS> Get-AdsRoute -All -name "Tc3*"
Name           NetId            Address       Sub TcVersion RTSystem
----           -----            -------       --- -------- --------
TC3TestA1-CP67x 172.17.62.105.1.1 172.17.62.105 3.1.4021   Win7
TC3Test13-C6650 172.17.66.139.1.1 172.17.62.156 2.11.2246  Win7
```

```
PS> $cred = Get-Credential -Message "Get Credentials" -UserName "UserName"
```

```
PS> Add-AdsRoute -Credential $cred -name "TC3TestA1-CP67x" -selfSigned
```

```
Name           NetId            Address       Sub TcVersion RTSystem
----           -----            -------       --- -------- --------
TC3TestA1-CP67x 172.17.62.105.1.1 172.17.62.105 3.1.4021   Win7
```

```
PS> Get-AdsRoute -name "TC3TestA1-CP67x" | Test-AdsRoute
```

Search for Systems that start with the name "TC3*", then asks the user for Credentials and adds the Route with 'SelfSigned' AdsSecure settings.

Afterwards, the connection is checked.

**EXAMPLE 3**

```
PS> Add-AdsRoute -Address 172.17.62.105 -sca
```

```
Name           NetId            Address       Sub TcVersion RTSystem
--------------- -----------            -------       --- -------- --------
TC3TestA1-CP67x 172.17.62.105.1.1 172.17.62.105 3.1.4021   Win7
```

Search for the system with the specified IPAddress, and add the Route with Shared Certification Authority settings without password.

The precondition is, that valid certificates are already established on both (engineering and remote) systems, within their StaticRoutes.xml files.

**EXAMPLE 4**

```
PS> $cred = Get-Credential -Message "Get Credentials" -UserName "UserName"
```

```
PS> Add-AdsRoute -Credential $cred -NetId 172.17.62.105 -Nat 1.2.3.4.1.1
```

```
Name           NetId            Address       Sub TcVersion RTSystem
--------------- -----------            -------       --- -------- --------
TC3TestA1-CP67x 1.2.3.4.1.1 172.17.62.105        3.1.4024   Win10 (2004)
```

Add a route with a local network address translation (NAT AmsNetId) to project a remote AmsNetId (RemoteNetId) locally to a different address.

**PARAMETERS**

- **-Address**

The address for the ADS route.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

Type: String[]
Parameter Sets: Address, AddressPSK, AddressSCA, AddressSSC
Aliases: Name, TargetAddress
- **BinaryKey**

  The **BinaryKey** can be used instead of the credential **Password** on **SecureSettings.PreSharedKeys** (PSK).

  There is no function for this parameter on other security settings.

  Type: **Byte[]**  
  Parameter Sets: AddressPSK, NetIdPSK, RoutesPSK  
  Aliases:  
  Required: False  
  Position: Named  
  Default value: None  
  Accept pipeline input: False  
  Accept wildcard characters: False

- **Credential**

  Credentials of the route to be added to the destination system.

  **IMPORTANT:** Please be aware, that in the current version, the password is transferred as clear text through the network.

  Use this only in safe subnetworks.

  Type: **PSCredential**  
  Parameter Sets: Address, AddressPSK, AddressSSC, NetId, NetIdPSK, NetIdSSC, Routes, RoutesPSK, RoutesSSC  
  Aliases: TargetCredential  
  Required: True  
  Position: Named  
  Default value: None  
  Accept pipeline input: False  
  Accept wildcard characters: False

- **Destination**

  The **Destination Address**, where the route is added.

  Type: **String**  
  Parameter Sets: (All)  
  Aliases:  
  Required: False  
  Position: Named  
  Default value: None  
  Accept pipeline input: False  
  Accept wildcard characters: False

- **DestinationCredential**

  The credentials of the destination system, where to add the route.

  Local system by default.

  Type: **PSCredential**  
  Parameter Sets: (All)  
  Aliases:  
  Required: False  
  Position: Named  
  Default value: None
-HostName

If set, the route will be registered as HostName

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-IgnoreCN

Gets or sets the 'Ignore Common Name' mode for SharedCertificateAuthority (SCA) while adding the route.

The "CommonName" of the certificate must correspond to the name used when establishing the connection in the certificate.

This behavior can be deactivated by this option.

Type: SwitchParameter
Parameter Sets: AddressSCA, NetIdSCA, RoutesSCA
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-InputObject

The input Ads Routes.

Type: RouteTargetCollection
Parameter Sets: Routes, RoutesPSK, RoutesSCA, RoutesSSC
Aliases: Route, TargetRoute

Required: True
Position: 0
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-Nat

The 'Nat' parameter sets the local representation of the routes AmsNetId.

All (local) addressing to this netId will be translated to the remote/network AmsAddress of the route.

This Parameter can be used with TwinCAT Versions \( \geq 3.1.4024.11 \)

Type: AmsNetId
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
-NetId

The AmsNetID for the ADS route to add.

Type: AmsNetId
Parameter Sets: NetId, NetIdPSK, NetIdSCA, NetIdSSC
Aliases: TargetNetId

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-PreSharedKey

Gets or sets the PreSharedKey (PSK) mode for adding the route.

Because a valid certificate is expected at the target, it is not necessary to enter credentials.

Type: SwitchParameter
Parameter Sets: AddressPSK, NetIdPSK, RoutesPSK
Aliases: PSK

Required: True
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Quiet

The Quiet parameter suppresses the 'ShouldProcess' message and the ADSRoute will be added without further question.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-SelfSigned

Gets or sets the SelfSigned (SSC) mode for adding the route.

Type: SwitchParameter
Parameter Sets: AddressSSC, NetIdSSC, RoutesSSC
Aliases: SSC

Required: True
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-SharedCertAuth

Gets or sets the SharedCertificateAuthority (SCA) mode for adding the route.

Because a valid certificate is expected at the target, it is not necessary to enter credentials.

Type: SwitchParameter
Parameter Sets: AddressSCA, NetIdSCA, RoutesSCA
Aliases: SCA

Required: True
Position: Named
**-Temporary**

If set, the Route will be registered as temporary route.

Type: SwitchParameter  
Parameter Sets: (All)  
Aliases:

Required: False  
Position: Named  
Default value: False  
Accept pipeline input: False  
Accept wildcard characters: False

**-Timeout**

(Broadcast) Search Timeout for searching the unregistered target (Default 5000).

Type: Int32  
Parameter Sets: Address, AddressPSK, AddressSCA, AddressSSC  
Aliases:

Required: False  
Position: Named  
Default value: 5000  
Accept pipeline input: False  
Accept wildcard characters: False

**-Unidirectional**

Gets or sets the unidirectional setting.

The Unidirectional setting registers the ADS Route as 'one-way' channel.

That means that the engineering/source system (that's where the route request is initiated) can send requests to the remote target, but not in the opposite direction.

Type: SwitchParameter  
Parameter Sets: (All)  
Aliases:

Required: False  
Position: Named  
Default value: False  
Accept pipeline input: False  
Accept wildcard characters: False

**-Confirm**

Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter  
Parameter Sets: (All)  
Aliases: cf

Required: False  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

**-WhatIf**

Shows what would happen if the cmdlet runs.
The cmdlet is not run.

**Type:** SwitchParameter  
**Parameter Sets:** (All)  
**Aliases:** wi  
**Required:** False  
**Position:** Named  
**Default value:** None  
**Accept pipeline input:** False  
**Accept wildcard characters:** False

### CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkId=113216).

### INPUTS

**TwinCAT.RouteTargetCollection**  
The input Ads Routes.

### 5.4 Add-MqttRoute

**SYNOPSIS**

Adds an MQTT route to the destination system.

**SYNTAX**

**Default (Default)**

```
[-Confirm] [<CommonParameters>]
```

**Identity**

```
[-Credential <PSCredential>] [-[IdentityCaseSensitive] [-WhatIf] [-Confirm] [<CommonParameters>]
```

**Psk**

```
-Identity <String> -PreSharedKey <String> [-WhatIf] [-Confirm] [<CommonParameters>]
```

**SCA**

```
-CA <String> -Cert <String> -Key <String> [-WhatIf] [-Confirm] [<CommonParameters>]
```

**DESCRIPTION**

This Cmdlet adds an MQTT route to the destination system.

To add the route, the Address of a MQTT route must be specified.
EXAMPLES

EXAMPLE 1
> Add-MqttRoute -Address 1.2.3.4 -port 42

Adds an MQTT route to an MQTT Broker system with the IPAddress '1.2.3.4' and Port '42'.

EXAMPLE 2
> Add-MqttRoute -Address MqttSystem -port 42 -Destination CX_1234

Adds an MQTT route on the destination System 'CX_1234' with DnsHost Address 'MqttSystem' and Port '42'.

PARAMETERS

- Address

The Address of the MQTT Broker to add.

This can be the HostName or the IPAddress.

Type: String  
Parameter Sets: (All)  
Aliases:

Required: True  
Position: 0  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

- CA

Path to the Certificate Authority file.

Certificates of MQTT broker, signed by this CA will be accepted for connection.

The file must be already located on the target system.

Type: String  
Parameter Sets: SCA  
Aliases:

Required: True  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

- Cert

Path to the public key Certificate (X.509).

The file must be already located on the target system.

Type: String  
Parameter Sets: SCA  
Aliases:

Required: True  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False
- **Credential**

Credentials of the Preshared Key Identity (PSK).

This consists of Identity/UserName and the Password IMPORTANT: Please be aware, that in the current version, the password is transferred as clear text through the network.

Use this only in safe subnetworks.

Type: PSCredential
Parameter Sets: Identity
Aliases: TargetCredential

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

- **Destination**

The Destination Address, where the MQTT route is added remotely.

Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

- **Identity**

The identity name used to talk to the MQTT message broker (Preshared Key method).

Type: String
Parameter Sets: Psk
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

- **IdentityCaseSensitive**

Use the Identity case sensitive for key calculation.

Type: SwitchParameter
Parameter Sets: Identity
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

- **Key**

Path of the private Key file of the X.509 Certificate.

The file must be already located on the target system.

Type: String
Parameter Sets: SCA
Aliases:
Required: True  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

-**Port**

The TCP/IP Port of the MQTT Broker to add.

Type: Int32  
Parameter Sets: (All)  
Aliases:

Required: True  
Position: 1  
Default value: 0  
Accept pipeline input: False  
Accept wildcard characters: False

-**PreSharedKey**

The Preshared key used together with the identity for MQTT message broker communication.

Type: String  
Parameter Sets: Psk  
Aliases:

Required: True  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

-**Topic**

The MQTT Topic string under which this MQTT Consumer sends/receives data.

Type: String  
Parameter Sets: (All)  
Aliases:

Required: False  
Position: 2  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

-**Confirm**

Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter  
Parameter Sets: (All)  
Aliases: cf

Required: False  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

-**WhatIf**

Shows what would happen if the cmdlet runs.

The cmdlet is not run.
5.5 Close-TcSession

SYNOPSIS
Closes the specified session object.

SYNTAX
Default (Default)
Close-TcSession -Id <Int32> [CommonParameters]

Session
Close-TcSession -InputObject <ISession> [CommonParameters]

DESCRIPTION
This Cmdlet closes the specified Point-To-Point Connection to the TwinCAT Target that is represented by the returned session object.

All registered SessionProvider types of Sessions can be used here (e.g. ADS, MQTT, OPC).

EXAMPLES
EXAMPLE 1
PS > Close-AdsSession $session

PARAMETERS

-Id
The session object to close is specified by this session ID.

Type: Int32
Parameter Sets: Default
Aliases:
Required: True
Position: Named
Default value: 0

CommonParameters
This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkId=113216).
Accept pipeline input: False
Accept wildcard characters: False

-InputObject

The Session object to close.

Type: ISession
Parameter Sets: Session
Aliases: Session
Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

TwinCAT.ISession

The Session object to close.

5.6 Copy-AdsFile

SYNOPSIS

Uploads / Downloads files from/to TwinCAT target.

SYNTAX

NetId (Default)

[-NetId <AmsNetId>] [<CommonParameters>]

Route

-InputObject <IRoute> [<CommonParameters>]

AddressStr

-Address <String> [<CommonParameters>]

SessionId

-SessionId <Int32> [<CommonParameters>]
DESCRIPTION
This Cmdlet implements ADS file transfer operations with TwinCAT Systems.

EXAMPLES

EXAMPLE 1
PS > Copy-AdsFile -address CX_00001 -path CurrentConfig.xml -Destination c:\tmp\Config1.xml -Directory BootDir

Downloads the CurrentConfig.xml from the BootDir of the target system to 'c:\tmp\Config1.xml'

EXAMPLE 2
PS > Copy-AdsFile -address CX_00001 -upload -path c:\tmp\Config1.xml -destination CurrentConfig.xml -Directory BootDir

Uploads the file "c:\tmp\Config1.xml" on local system to the Target BootFolder of system CX_00001

EXAMPLE 3
PS > Copy-AdsFile -address CX_00001 -path c:\ReadMe.txt -destination d:\tmp\ReadMe.txt

Downloads the File "C:\ReadMe.txt" form System CX_0001 to the local system and store it under d:\tmp\ReadMe.txt

PARAMETERS

-Address
The address of the system where the file is Downloaded from / Uploaded to (Default: Local) This can be the RouteName, NetId, the HostName or the IPAddress.
  Type: String
  Parameter Sets: AddressStr
  Aliases:
    Required: True
    Position: Named
    Default value: None
    Accept pipeline input: False
    Accept wildcard characters: True

-Destination
The Destination pathspecifier, where the file is stored.
If the Cmdlet is in Download mode, this has to be the FullPath of the target location.
In case of 'Uploading' this can be the FileName or a FullPath dependent of the StandardDirectory Parameter.
  Type: String
  Parameter Sets: (All)
  Aliases:
    Required: False
    Position: 1
    Default value: None
    Accept pipeline input: False
    Accept wildcard characters: False

-Directory
The Directory specifier on the remote system.
The Default is "Generic".

Possible values: Generic, BootDir, TargetDir, ConfigDir, InstallDir, RepositoryDir, UserPath1, UserPath2, UserPath3, UserPath4, UserPath5, UserPath6, UserPath7, UserPath8, UserPath9

Type: PathSpecifier
Parameter Sets: (All)
Aliases:
Accepted values: Generic, BootDir, TargetDir, ConfigDir, InstallDir, RepositoryDir, UserPath1, UserPath2, UserPath3, UserPath4, UserPath5, UserPath6, UserPath7, UserPath8, UserPath9

Required: False
Position: Named
Default value: Generic
Accept pipeline input: False
Accept wildcard characters: False

-Force

Forces to create the Directory on the target side (and overwrites any preexisting file).

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-InputObject

The address (Route) of the system where the file is Downloaded from / Uploaded to (Default: Local)

Type: IRoute
Parameter Sets: Route
Aliases: Route

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-NetId

The address (AmsNetId) of the system where the file is Downloaded from / Uploaded to (Default: Local)

Type: AmsNetId
Parameter Sets: NetId
Aliases:

Required: False
Position: Named
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

-Path

The source path specifier, where the file is taken from.

If this Cmdlet is in Download mode, this is the specifier or FullPath of the (remote) file, dependant of the StandardFolder Parameter.

In case of 'Uploading' this is the FullPath of the file to be transferred.

Type: String
Parameter Sets: (All)
Aliases:
-SessionId

The target system address is derived from the Session Information where the file is Downloaded from / Uploaded to.  
Type: Int32  
Parameter Sets: SessionId  
Aliases: Id  
Required: True  
Position: Named  
Default value: -1  
Accept pipeline input: False  
Accept wildcard characters: False

-Upload

Switches the Cmdlet to Upload mode.  
If not set, the Cmdlet is in 'Download' mode.  
Type: SwitchParameter  
Parameter Sets: (All)  
Required: False  
Position: Named  
Default value: False  
Accept pipeline input: False  
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

TwinCAT.IRoute

The address (Route) of the system where the file is Downloaded from / Uploaded to (Default: Local)

5.7 Get-AdsRoute

SYNOPSIS

List routes on a TwinCAT System / Broadcast search.

SYNTAX

GetRoutes (Default)
[<CommonParameters>]

5.7 Get-AdsRoute

SYNOPSIS

List routes on a TwinCAT System / Broadcast search.

SYNTAX

GetRoutes (Default)
[<CommonParameters>]

5.7 Get-AdsRoute

SYNOPSIS

List routes on a TwinCAT System / Broadcast search.

SYNTAX

GetRoutes (Default)
[<CommonParameters>]
LocalSystem
Get-AdsRoute [-InputObject <String>] [-Local] [<CommonParameters>]

DESCRIPTION
This Cmdlet can list the routes configured on a TwinCAT local/remote system, or start determining all TwinCAT Systems within the current subnet.

EXAMPLES

EXAMPLE 1
PS> Get-AdsRoute
Name          NetId          Address       Sub     TcVersion     RTSystem
-------------- ----------- -------------- ------ ---------- ----------
CP-15ECA0      172.17.62.128.1.1 172.17.62.178  [UNKNOWN]  [UNKNOWN]
CP-15ECA1      172.17.62.105.1.1 172.17.62.105  [UNKNOWN]  [UNKNOWN]

Lists all registered local routes.
Because only the local port 10000 is addressed, the TcVersion and RTSystem is unknown (the Cmdlet doesn't contact the targets and doesn't produce additional roundtrips.

EXAMPLE 2
PS> Get-AdsRoute -All
Name          NetId          Address       Sub     Version     RTSystem
-------------- ----------- -------------- ------ -------- ----------
CX-1CEEDA      5.16.136.222.1.1 172.17.62.139  3.1.4020  Win7
CX-20BC62      5.32.188.98.1.1  172.17.62.90   3.1.4020  CE6.0
CX-10A87B      5.16.168.123.1.1 172.17.62.140  2.11.2254 CE7.0
CP-15ECA0      172.17.62.128.1.1 172.17.62.178  3.1.4021  Win7
CX-0A7F60      5.10.127.96.1.1   172.17.62.148  3.1.4020  XP
CX2030-B4018   172.17.60.157.1.1 172.17.60.159  2.11.2256 Win7
CP_11BB16      5.17.187.22.1.1  172.17.60.180   2.11.2038 CE6.0
CX-12BC85      172.17.60.165.1.1 172.17.62.191  2.11.2237 CE7.0
CX-124218      5.16.66.24.1.1   172.17.60.192  3.1.4021  Win7
CX-1D2AA       172.17.62.180.1.1 172.17.62.180  3.1.4021  Win8
CX_0AB4F0      5.10.180.240.1.1 172.17.60.195   2.11.2243 XP
CP_1DFA0A      172.17.62.118.1.1 172.17.62.118  3.1.4021  Win7
CX-AF0001      172.17.62.75.1.1   172.17.62.70   3.1.4020  Win10

Start a Broadcast search from the local system and lists the devices within the connected network.

EXAMPLE 3
PS> Get-AdsRoute -Name "Tc3*"
Name          NetId          Address       Sub     Version     RTSystem
-------------- ----------- -------------- ------ -------- ----------
TC3TESTA1-CP67X 172.17.62.105.1.1 172.17.62.105  0.0   Unknown

Get the (actual) route assigned to the local system that has the name pattern "Tc3"

EXAMPLE 4
PS> Get-AdsRoute -All | where TcVersion -lt "3.1.0.0"
Name          NetId          Address       Sub     Version     RTSystem
-------------- ----------- -------------- ------ -------- ----------
TC3Test17-C6930 172.17.62.98.1.1 172.17.62.98  2.11.2234  Win7
CX2030-B4018   172.17.60.157.1.1 172.17.60.159  2.11.2256  Win7
CX-10A87B      5.16.168.123.1.1  172.17.62.140  2.11.2254  CE7.0
TC3Test13-C6650 172.17.60.239.1.1 172.17.62.156  2.11.2246  Win7
ECATTest01     172.17.61.6.1.1   172.17.61.31   2.11.2239  Win7
CX-12BC85      172.17.60.165.1.1 172.17.62.191  2.11.2237  CE7.0
CX_0AB4F0      5.10.180.240.1.1 172.17.60.195   2.11.2243  XP
PARAMETERS

-Address

The Name / Address of the route to get.

The address of the route can be coded as NetId, the HostName or the IPAddress in string representation.

Wildcards are permitted.

Type: String[
Parameter Sets: GetRoutes
Aliases: Name

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-All

Broadcast switch.

If activated a broadcast search is triggered within the local network.

The search can be constrained additionally by the -Address/-Name parameter.

Searching by Address (direct access of targets if no wildcards, otherwise using Broadcast search): - HostName: Searching the target by dns resolution and then via IP (fallback broadcast search filtering DeviceName/Hostname, not working over subnets!) - IPAddress: Directly accessing via IP (works also over subnets) - AmsNetId: Working via Broadcast search (not working over subnet segments!) Searching by Name: Works always via Broadcast search, wildcards permitted

Type: SwitchParameter
Parameter Sets: GetRoutes
Aliases: Broadcast

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-InputObject

The Destination address specifies the target, where the routes are determined.

Use this to get the registered routes of a remote system.

The Destination system can be specified by RouteName (route name on local system), AmsNetId, IPAddress or HostName

Type: String
Parameter Sets: (All)
Aliases: Destination

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
- **Local**

If set, the local system route will be returned.

By default a list of the actual registered routes will be returned.

Type: SwitchParameter
Parameter Sets: LocalSystem
Aliases: Self

Required: True
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

- **StaticRoutes**

Indicates that only static routes will be returned.

By default this Cmdlet returns the actual registered routes.

Type: SwitchParameter
Parameter Sets: GetRoutes
Aliases: Static

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

- **Timeout**

(Broadcast) Search Timeout (Default 5000 ms)

Type: Int32
Parameter Sets: GetRoutes
Aliases:

Required: False
Position: Named
Default value: 5000
Accept pipeline input: False
Accept wildcard characters: False

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkID=113216).

5.8  **Get-AdsState**

**SYNOPSIS**

Gets the Ads State of a TwinCAT Target.

**SYNTAX**

AddressStr
[<CommonParameters>]

Route
[<CommonParameters>]

DESCRIPTION
This command let gets the ADS state of a TwinCAT target.

EXAMPLES

EXAMPLE 1
PS > Get-AdsState 1.2.3.4.5.6

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
<th>OK</th>
<th>Time (ms)</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORK01</td>
<td>Config</td>
<td>True</td>
<td>0</td>
<td>1.2.3.4.5.6</td>
</tr>
</tbody>
</table>

EXAMPLE 2
PS > Get-AdsState

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
<th>OK</th>
<th>Time (ms)</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORK01</td>
<td>Config</td>
<td>True</td>
<td>0</td>
<td>1.2.3.4.5.6</td>
</tr>
</tbody>
</table>

EXAMPLE 3
PS > Get-AdsState 1.2.3.4,CX_0130C7

EXAMPLE 4
PS > get-route | get-adsState

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
<th>OK</th>
<th>Time (ms)</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORK01</td>
<td>Config</td>
<td>True</td>
<td>0</td>
<td>1.2.3.4.5.6</td>
</tr>
<tr>
<td>CX_0130C7</td>
<td>Config</td>
<td>True</td>
<td>0</td>
<td>5.1.48.199.1.1</td>
</tr>
</tbody>
</table>

PARAMETERS

-Address
The address(es) where to get the State.
This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.
Type: String[]
Parameter Sets: AddressStr
Aliases:
Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
**-Force**

Force Mode

<table>
<thead>
<tr>
<th>Type:</th>
<th>SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required:</td>
<td>False</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>False</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

**-InputObject**

The target systems, where to get the AdsState from.

<table>
<thead>
<tr>
<th>Type:</th>
<th>RouteTargetCollection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>Route</td>
</tr>
<tr>
<td>Aliases:</td>
<td>Destination, Source</td>
</tr>
<tr>
<td>Required:</td>
<td>False</td>
</tr>
<tr>
<td>Position:</td>
<td>1</td>
</tr>
<tr>
<td>Default value:</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>True (ByPropertyName, ByValue)</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

**-NetId**

The NetID of the target system.

<table>
<thead>
<tr>
<th>Type:</th>
<th>AmsNetId</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>NetIdPort</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required:</td>
<td>False</td>
</tr>
<tr>
<td>Position:</td>
<td>1</td>
</tr>
<tr>
<td>Default value:</td>
<td>172.17.60.167.1.1</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

**-Port**

The AmsPort of the target system.

<table>
<thead>
<tr>
<th>Type:</th>
<th>Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required:</td>
<td>False</td>
</tr>
<tr>
<td>Position:</td>
<td>2</td>
</tr>
<tr>
<td>Default value:</td>
<td>10000</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

**-Quiet**

The quiet mode.

<table>
<thead>
<tr>
<th>Type:</th>
<th>SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required:</td>
<td>False</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>False</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>
- **Timeout**

Communication timeout in ms.

<table>
<thead>
<tr>
<th>Type: Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: (All)</td>
</tr>
<tr>
<td>Aliases:</td>
</tr>
<tr>
<td>Required: False</td>
</tr>
<tr>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: -1</td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
</tr>
</tbody>
</table>

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkID=113216).

**INPUTS**

**TwinCAT.RouteTargetCollection**

The target systems, where to get the AdsState from.

### 5.9 Get-TcDataType

**SYNOPSIS**

Get the DataTypes from a TwinCAT target system / Device.

**SYNTAX**

**NetIdPort (Default)**

Get-TcDataType [[-Name] <String[]>] [-NetId <AmsNetId>] -Port <Int32> [CommonParameters]

**Route**

Get-TcDataType [[-Name] <String[]>] -Route <IRoute> -Port <Int32> [CommonParameters]

**AddressStr**

Get-TcDataType [[-Name] <String[]>] -Address <String> -Port <Int32> [CommonParameters]

**Session**

Get-TcDataType [[-Name] <String[]>] -InputObject <ISession> [CommonParameters]

**SessionId**

Get-TcDataType [[-Name] <String[]>] -SessionId <Int32> [CommonParameters]

**DESCRIPTION**

This Cmdlet get the DataTypes from a target system if symbolic information is provided by the device (Symbol Server running).
The DataTypes can be determined via different Providers (e.g. ADS, MQTT, OPC, see the `-Provider` parameter.)

**EXAMPLES**

**EXAMPLE 1**

```powershell
PS> Get-TcDataType -port 851
Name | Size | Category | BaseType
---- | ---- | -------- | --------
BYTE | 1   | Primitive | BYTE
WORD | 2   | Primitive | WORD
DINT | 4   | Primitive | DINT
UDINT | 4  | Primitive | UDINT
DWORD | 4  | Primitive | DWORD
E_ByteEnum | 1 | Enum | BYTE
FB_Test | 12424 | Struct |
PLC.PlcAppSystemInfo | 256 | Struct |
PLC.PlcTaskSystemInfo | 128 | Struct |
POINTER TO BYTE | 4 | Pointer | BYTE
R_Range | 2 | Alias | INT (-6..12)
REFERENCE TO BOOL | 4 | Reference | BOOL
ST_SimpleStruct | 166 | Struct |
STRING(80) | 81 | String |
```

Get the data types from the local system (Port 851):

**EXAMPLE 2**

```powershell
PS> $types = Get-TcDataType -Name 'ST_*' -NetId 1.2.3.4.5.6 -Port 851
```

Gets the DataTypes with name pattern 'ST_*' from the NetId / Port address symbol server.

**EXAMPLE 3**

```powershell
PS> $session = New-TcSession -Name 'CX_123456' -port 851
PS> Get-TcDataType -Session $session | where ByteSize -gt 1KB
```

Gets an ADS-Session/Connection to the target system CX_123456 on port 851, downloads the datatype information and returns all the DataTypes that are larger than 1KB of Size.

**PARAMETERS**

**-Address**

The address where to load the datatype descriptions.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

Type: String
Parameter Sets: AddressStr
Aliases:
- Required: True
- Position: Named
- Default value: None
- Accept pipeline input: False
- Accept wildcard characters: True

**-InputObject**

The session object to use for datatype upload.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Type</th>
<th>Parameter Sets</th>
<th>Required</th>
<th>Position</th>
<th>Default value</th>
<th>Accept pipeline input</th>
<th>Accept wildcard characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Name</td>
<td>The data type name(s) to get. Wildcards are permitted.</td>
<td>String[]</td>
<td>(All)</td>
<td>False</td>
<td>0</td>
<td>None</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>-NetId</td>
<td>The NetID address of the target system where to load the datatypes (Local by default).</td>
<td>AmsNetId</td>
<td>NetIdPort</td>
<td>False</td>
<td>Named</td>
<td>172.17.60.167.1.1</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>-Port</td>
<td>The Port where to load the datatype descriptions.</td>
<td>Int32</td>
<td>NetIdPort, Route, AddressStr</td>
<td>True</td>
<td>Named</td>
<td>10000</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>-Route</td>
<td>The Route object where to load the datatypes from (RouteTarget.Local by default).</td>
<td>IRoute</td>
<td>Route</td>
<td>True</td>
<td>Named</td>
<td>None</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>
-SessionId

The unique session Identifier that represents the session to use for the datatype upload.

Type: Int32
Parameter Sets: SessionId
Aliases: Id

Required: True
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

TwinCAT.ISession
The session object to use for datatype upload.

5.10 Get-TcLicense

SYNOPSIS

Get TwinCAT License information.

SYNTAX

NetIdPort (Default)
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] [-NetId <AmsNetId>] [-Status <LicenseStatus>] [<CommonParameters>]

Route
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] -Route <IRoute> [-Status <LicenseStatus>] [<CommonParameters>]

AddressStr
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] -Address <String> [-Status <LicenseStatus>] [<CommonParameters>]

Session
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] -InputObject <ISession> [-Status <LicenseStatus>] [<CommonParameters>]

SessionId
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] -SessionId <Int32> [-Status <LicenseStatus>] [<CommonParameters>]
DESCRIPTION

This Cmdlet gets information about TwinCAT licenses from the target system.

To contact the target system, it must be available as actual route or the local system.

EXAMPLES

EXAMPLE 1

PS> Get-TcLicense

Get the the valid licenses from the local system.

EXAMPLE 2

PS> $session = New-TcSession -Route TC3TESTA1-CP67X -Port 30
PS> $session | Get-TcLicense -Status All -name "scope"

<table>
<thead>
<tr>
<th>Name</th>
<th>Valid</th>
<th>ValidityCode</th>
<th>ExpireTime</th>
<th>Available</th>
<th>Used</th>
<th>VolumeNo</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC3 Scope Server</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 Scope View Professional</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Create a session to the License Server on target 'TC3TESTA1-CP67X' and return all valid and invalid licenses that contain 'scope' in their name.

EXAMPLE 3

PS> Get-TcLicense -Route TC3TESTA1-CP67X -Status Valid

<table>
<thead>
<tr>
<th>Name</th>
<th>Valid</th>
<th>ValidityCode</th>
<th>ExpireTime</th>
<th>Available</th>
<th>Used</th>
<th>VolumeNo</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC3 C++ / MatSim</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 CNC</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 Target For Matlab Simulink</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 CNC Axis</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 Serial-Communication</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 NC FTP Axes Pack unlimited</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 PLC / C++ / MatSim</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 Kinematic Transformation L4</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 NC Camming</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 PLC-HMI Web</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 NC Flying Saw</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 CNC Spline</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 SMS-SMTP</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 Hydraulic Positioning</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TC3 Kinematic Transformation L1</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Connect to the License Server on target 'TC3TESTa1-CP67X' and return all valid licenses.

EXAMPLE 4

> Get-TcLicense -NetId 172.17.60.153.1.1 -Status Invalid | format-list

<table>
<thead>
<tr>
<th>Id</th>
<th>: 4c256767-e6e6-4af5-bd68-9f7abad0c200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>: TC3 ADS</td>
</tr>
<tr>
<td>ExpireTime</td>
<td>: 8/17/2017 12:00:00 AM</td>
</tr>
<tr>
<td>ValidityCode</td>
<td>: Expired</td>
</tr>
<tr>
<td>Valid</td>
<td>: False</td>
</tr>
<tr>
<td>AvailableLicenses: 0</td>
<td></td>
</tr>
<tr>
<td>UsedLicenses</td>
<td>: 0</td>
</tr>
<tr>
<td>VolumeNo</td>
<td>: 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Id</th>
<th>: 66689887-ccb0-452c-ac9a-039d997c6e66</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>: TC3 PLC</td>
</tr>
<tr>
<td>ExpireTime</td>
<td>: 8/17/2017 12:00:00 AM</td>
</tr>
<tr>
<td>ValidityCode</td>
<td>: Expired</td>
</tr>
<tr>
<td>Valid</td>
<td>: False</td>
</tr>
<tr>
<td>AvailableLicenses: 0</td>
<td></td>
</tr>
</tbody>
</table>
Connect to NetId 172.17.60.153.1.1, determine all invalid licenses and format the result into a list.

EXAMPLE 5

> Get-TcLicense -OrderId TE*

<table>
<thead>
<tr>
<th>OrderID</th>
<th>Name</th>
<th>Valid</th>
<th>ValidityCode</th>
<th>ExpireTime</th>
<th>Available</th>
<th>Used</th>
<th>VolumeNo</th>
</tr>
</thead>
<tbody>
<tr>
<td>TE1400</td>
<td>TC3 Target For Matlab Simulink</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TE1500</td>
<td>TC3 Valve-Diagram-Editor</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TE1120</td>
<td>TC3 XCAD Interface</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TE1110</td>
<td>TC3 Cam-Design-Tool</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TE1111</td>
<td>TC3 EtherCAT Simulation</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TE1410</td>
<td>TC3 Interface For Matlab Simulink</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TE1300</td>
<td>TC3 Scope View Professional</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Get the valid licenses from local system and filter them for OrderIds starting with TE*.

PARAMETERS

- **Address**

  The address where to load the licenses.

  This can be the RouteName, NetId, the HostName or the IPAddress.

  Wildcards are permitted.

  Type: String
  Parameter Sets: AddressStr
  Aliases:
  Required: True
  Position: Named
  Default value: None
  Accept pipeline input: True (ByValue)
  Accept wildcard characters: True

- **InputObject**

  The session object to use for license upload.

  This must target port 30 (AmsPort.R0_LicenseServer).

  Type: ISession
  Parameter Sets: Session
  Aliases: Session

  Required: True
  Position: Named
  Default value: None
  Accept pipeline input: True (ByValue)
  Accept wildcard characters: False

- **Name**

  The name of the license to get.
Wildcards are permitted.

Type: String[]
Parameter Sets: (All)
Aliases:

- Required: False
- Position: Named
- Default value: None
- Accept pipeline input: False
- Accept wildcard characters: True

---

**-NetId**

The NetID address of the target system where to load the licenses (Local by default).

Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

- Required: False
- Position: Named
- Default value: 172.17.60.167.1.1
- Accept pipeline input: True (ByValue)
- Accept wildcard characters: False

---

**-OrderId**

The OrderID of the license.

Wildcards are permitted.

Type: String[]
Parameter Sets: (All)
Aliases:

- Required: False
- Position: Named
- Default value: None
- Accept pipeline input: False
- Accept wildcard character: True

---

**-Route**

The Route object where to load the licenses from (RouteTarget.Local by default).

Type: IRoute
Parameter Sets: Route
Aliases: Destination

- Required: True
- Position: Named
- Default value: None
- Accept pipeline input: True (ByValue)
- Accept wildcard characters: False

---

**-SessionId**

The unique session Identifier that represents the session to use for the license upload.

Type: Int32
Parameter Sets: SessionId
Aliases: Id

- Required: True
- Position: Named
- Default value: -1
- Accept pipeline input: False
- Accept wildcard characters: False
- **Status**
The Status parameter selects the Licenses to return.

Available is 'Valid' (the valid licenses), 'Invalid' (the invalid licenses) and 'All' ('Valid' + 'Invalid') licenses.

The Default is 'All'

Possible values: None, Valid, Invalid, All

Type: LicenseStatus
Parameter Sets: (All)
Aliases:
Accepted values: None, Valid, Invalid, All

Required: False
Position: Named
Default value: All
Accept pipeline input: False
Accept wildcard characters: False

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkID=113216).

**INPUTS**

**TwinCAT.Ads.AmsNetId**
The NetID address of the target system where to load the licenses (Local by default).

**TwinCAT.IRoute**
The Route object where to load the licenses from (RouteTarget.Local by default).

**System.String**
The address where to load the licenses.
This can be the RouteName, NetId, the HostName or the IPAddress.
Wildcards are permitted.

**TwinCAT.ISession**
The session object to use for license upload.
This must target port 30 (AmsPort.R0_LicenseServer).

### 5.11 Get-TcRouterInfo

**SYNOPSIS**
Gets the router status information of the specified target system.

**SYNTAX**

```
NetIdPort (Default)
Get-TcRouterInfo [[-NetId] <AmsNetId[]>] [-Async] [-Timeout <Int32>] [CommonParameters]
```
**Route**

Get-TcRouterInfo [-InputObject] <RouteTargetCollection> [-Async] [-Timeout <Int32>] [<CommonParameters>]

**AddressStr**

Get-TcRouterInfo [-Address] <String[]> [-Async] [-Timeout <Int32>] [<CommonParameters>]

**Session**

Get-TcRouterInfo -Session <ISession[]> [-Async] [-Timeout <Int32>] [<CommonParameters>]

**SessionId**

Get-TcRouterInfo -SessionId <Int32[]> [-Async] [-Timeout <Int32>] [<CommonParameters>]

**DESCRIPTION**

This Cmdlet gets status information from the specified target system.

To contact the target system, it must be available as actual route or must be the local system.

The status information contains the amount of overall router memory and the used memory.

Furthermore the number of active connections and the size of the actual router mailbox will be shown.

**EXAMPLES**

**EXAMPLE 1**

```
PS > Get-TcRouterInfo

Target          Version       Level OS  Image  Device  CPUArch
-------          -------       ------   ----    ------   --------
TC3TESTA1-CP67X 3.1.4021.54   CP      Win7    IntelX86
```

Get router information from the local system.

**PARAMETERS**

**-Address**

Target names/addresses.

These can consist of RouteName, NetID, HostName or IPAddress.

Wildcards are permitted.

Type: String[]
Parameter Sets: AddressStr
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

**-Async**

Starts the write on different threads.

Only for internal use and test purposes.
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

- **-InputObject**

The route object where to get the Target information from..

Type: RouteTargetCollection
Parameter Sets: Route
Aliases: Destination, Route

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

- **-NetId**

NetId(s) of the target system.

Type: AmsNetId[]
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: 1
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

- **-Session**

The Session to use for the value read.

Type: ISession[]
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName)
Accept wildcard characters: False

- **-SessionId**

Specifies the Session (with unique ID) to use for the value read.

Type: Int32[]
Parameter Sets: SessionId
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

- **-Timeout**

Timeout of the separate ADS Read operations
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 2500
Accept pipeline input: False
Accept wildcard characters: False

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkId=113216).

**INPUTS**

**TwinCAT.RouteTargetCollection**

The route object where to get the Target information from..

**TwinCAT.ISession[]**

The Session to use for the value read.

### 5.12 Get-TcSession

**SYNOPSIS**

List the currently established Sessions.

**SYNTAX**

Default (Default)

Get-TcSession [-Force] [CommonParameters]

**Id**

Get-TcSession -Id <Int32> [-Force] [CommonParameters]

**DESCRIPTION**

This Cmdlet lists all actually Point-To-Point connections to TwinCAT Targets in form of their session representation.

Different types of Sessions can be accessed via the registered types of SessionProviders (e.g. ADS, MQTT, OPC).

**EXAMPLES**

**EXAMPLE 1**

PS > Get-TcSession
**PARAMETERS**

- **Force**

Forces the Cmdlet to determine also the internal used sessions.

Type: SwitchParameter  
Parameter Sets: (All)  
Aliases:  

Required: False  
Position: Named  
Default value: False  
Accept pipeline input: False  
Accept wildcard characters: False

- **Id**

Specifies the ID of the session to get.

Type: Int32  
Parameter Sets: Id  
Aliases: SessionID  

Required: True  
Position: Named  
Default value: 0  
Accept pipeline input: False  
Accept wildcard characters: False

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkID=113216).

---

**5.13 Get-TcSymbol**

**SYNOPSIS**

Get the symbols from a TwinCAT target system / Device.

**SYNTAX**

**NetIdPort (Default)**


**Route**


**AddressStr**


**Session**

### SessionId

```powershell
Get-TcSymbol [-Path] <String[]> [-SessionId <Int32>] [-Recurse] [-Force] [<CommonParameters>]
```

### DESCRIPTION

This Cmdlet get the symbolic information from a target system if symbols are provided.

The information can be determined via different Providers (e.g. ADS, MQTT, OPC).

### EXAMPLES

#### EXAMPLE 1

```powershell
$session = New-TcSession -NetId 1.2.3.4.5.6 -Port 851
$session | Get-TcSymbol "TwinCAT_SystemInfoVarList._AppInfo" -recurse
```

<table>
<thead>
<tr>
<th>InstanceName</th>
<th>DataType</th>
<th>Size</th>
<th>InstancePath</th>
</tr>
</thead>
<tbody>
<tr>
<td>_AppInfo</td>
<td>PLC.PlcAppSystemInfo</td>
<td>256</td>
<td>TwinCAT_SystemInfoVarList._AppInfo</td>
</tr>
<tr>
<td>ObjId</td>
<td>OTCID</td>
<td>4</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ObjId</td>
</tr>
<tr>
<td>TaskCnt</td>
<td>UDINT</td>
<td>4</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.TaskCnt</td>
</tr>
<tr>
<td>OnlineChangeCnt</td>
<td>UDINT</td>
<td>4</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.OnlineChangeCnt</td>
</tr>
<tr>
<td>Flags</td>
<td>DWORD</td>
<td>4</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.Flags</td>
</tr>
<tr>
<td>AdsPort</td>
<td>UINT</td>
<td>2</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.AdsPort</td>
</tr>
<tr>
<td>BootDataLoaded</td>
<td>BOOL</td>
<td>1</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.BootDataLoaded</td>
</tr>
<tr>
<td>OldBootData</td>
<td>BOOL</td>
<td>1</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.OldBootData</td>
</tr>
<tr>
<td>AppTimestamp</td>
<td>DT</td>
<td>4</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.AppTimestamp</td>
</tr>
<tr>
<td>KeepOutputsOnBP</td>
<td>BOOL</td>
<td>1</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.KeepOutputsOnBP</td>
</tr>
<tr>
<td>ShutdownInProgress</td>
<td>BOOL</td>
<td>1</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ShutdownInProgress</td>
</tr>
<tr>
<td>LicensesPending</td>
<td>BOOL</td>
<td>1</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.LicensesPending</td>
</tr>
<tr>
<td>BSODOccurred</td>
<td>BOOL</td>
<td>1</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.BSODOccurred</td>
</tr>
<tr>
<td>TComSrvPtr</td>
<td>ITComObjectServer</td>
<td>4</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.TComSrvPtr</td>
</tr>
<tr>
<td>AppName</td>
<td>STRING(63)</td>
<td>64</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.AppName</td>
</tr>
<tr>
<td>ProjectName</td>
<td>STRING(63)</td>
<td>64</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ProjectName</td>
</tr>
</tbody>
</table>

Get the root symbolic information from the local system (Port 851):

#### EXAMPLE 2

```powershell
$session = New-TcSession -Name 'CX_123456' -port 851
$session | Get-TcSymbol -recurse | where InstanceName -like 'Project*' | where InstanceName
```

<table>
<thead>
<tr>
<th>InstanceName</th>
<th>DataType</th>
<th>Size</th>
<th>InstancePath</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProjectName</td>
<td>STRING(63)</td>
<td>64</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ProjectName</td>
</tr>
</tbody>
</table>

Create a session to the target system ‘1.2.3.4.5.6’ Port: 851 and get the symbol ‘TwinCAT_SystemInfoVarList._AppInfo’ and its subsymbols recursively.

#### EXAMPLE 3

```powershell
$session = New-TcSession -Name 'CX_123456' -port 851
$session | Get-TcSymbol -recurse | where InstanceName -like 'Project*' |
```

<table>
<thead>
<tr>
<th>InstanceName</th>
<th>DataType</th>
<th>Size</th>
<th>InstancePath</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProjectName</td>
<td>STRING(63)</td>
<td>64</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ProjectName</td>
</tr>
</tbody>
</table>

Gets an ADS-Session/Connection to the target system CX_123456 on port 851, downloads the symbol information recursively and returns all Instances where the instance name is like the pattern 'Project*'.

---

54 Version: 1.2 TE1000
PARAMETERS

- **Address**
The address for the target system where to get the symbol.
This can be the RouteName, NetId, the HostName or the IPAddress.
Wildcards are permitted.

<table>
<thead>
<tr>
<th>Type</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>AddressStr</td>
</tr>
<tr>
<td>Aliases</td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td>True</td>
</tr>
<tr>
<td>Position</td>
<td>Named</td>
</tr>
<tr>
<td>Default value</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline input</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters</td>
<td>True</td>
</tr>
</tbody>
</table>

- **Force**
Active only in recursive mode - ignored otherwise.
This parameter forces the Cmdlet to output all symbols - even Array Elements.
Please take care because the output can be very lengthy dependent on the Size of the Array.

<table>
<thead>
<tr>
<th>Type</th>
<th>SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases</td>
<td>ArrayElements</td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
</tr>
<tr>
<td>Position</td>
<td>Named</td>
</tr>
<tr>
<td>Default value</td>
<td>False</td>
</tr>
<tr>
<td>Accept pipeline input</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters</td>
<td>False</td>
</tr>
</tbody>
</table>

- **InputObject**
The session object that is used to get the symbols.

<table>
<thead>
<tr>
<th>Type</th>
<th>ISession</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>Session</td>
</tr>
<tr>
<td>Aliases</td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td>True</td>
</tr>
<tr>
<td>Position</td>
<td>Named</td>
</tr>
<tr>
<td>Default value</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline input: True (ByValue)</td>
<td></td>
</tr>
<tr>
<td>Accept wildcard characters</td>
<td>False</td>
</tr>
</tbody>
</table>

- **NetId**
The target system NetId.

<table>
<thead>
<tr>
<th>Type</th>
<th>AmsNetId</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>NetIdPort</td>
</tr>
<tr>
<td>Aliases</td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
</tr>
<tr>
<td>Position</td>
<td>Named</td>
</tr>
<tr>
<td>Default value</td>
<td>172.17.60.167.1.1</td>
</tr>
<tr>
<td>Accept pipeline input</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters</td>
<td>False</td>
</tr>
</tbody>
</table>

- **Path**
The instance path of the Symbol(s).
Wildcards are permitted.

Type: String[]
Parameter Sets: (All)
Aliases:

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-Port

The target system port.

Type: Int32
Parameter Sets: NetIdPort, Route, AddressStr
Aliases:

Required: True
Position: Named
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False

-Recurse

Gets the symbol recursively.

Often used in conjunction with Wildcards in -Path

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Route

The target system route.

Type: IRoute
Parameter Sets: Route
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-SessionId

The unique id of the session object that is used to get the symbols.

Type: Int32
Parameter Sets: SessionId
Aliases: Id

Required: True
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkId=113216).

INPUTS

TwinCAT.ISession

The session object that is used to get the symbols.

5.14 Get-TcTargetInfo

SYNOPSIS

Get TwinCAT Device Target information.

SYNTAX

NetIdPort (Default)

Get-TcTargetInfo [[-NetId] <AmsNetId[]>] [-Timeout <Int32>] [-Async] [CommonParameters]

Route

Get-TcTargetInfo [-InputObject] <RouteTargetCollection> [-Timeout <Int32>] [-Async] [CommonParameters]

AddressStr

Get-TcTargetInfo [-Address] <String[]> [-Timeout <Int32>] [-Async] [CommonParameters]

Session

Get-TcTargetInfo -Session <ISession[]> [-Timeout <Int32>] [-Async] [CommonParameters]

SessionId

Get-TcTargetInfo -SessionId <Int32[]> [-Timeout <Int32>] [-Async] [CommonParameters]

DESCRIPTION

This Cmdlet gets information from the specified target system.

To contact the target system, it must be available as actual route.

The information contains the TargetName, TwinCAT Version, Running Operating system, CPU Architecture and Image Information.

EXAMPLES

EXAMPLE 1

PS > Get-TcTargetInfo

Target  Version  Level OS  Image Device CPUArch
Get the target information of the local system.

EXAMPLE 2

PS > get-adsRoute | Get-TcTargetInfo

<table>
<thead>
<tr>
<th>Target</th>
<th>Version</th>
<th>Level</th>
<th>OS</th>
<th>Image</th>
<th>Device</th>
<th>CPUArch</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP-15ECA0</td>
<td>3.1.4021.50</td>
<td>CP</td>
<td>Win7</td>
<td></td>
<td>IntelX86</td>
<td></td>
</tr>
<tr>
<td>TC3TESTA1-CP67X</td>
<td>3.1.4021.54</td>
<td>CP</td>
<td>Win7</td>
<td></td>
<td>IntelX86</td>
<td></td>
</tr>
</tbody>
</table>

Get the target information of the actual connected routes.

PARAMETERS

- **Address**

Target names/addresses.

These can consist of RouteName, NetID, HostName or IPAddress.

Wildcards are permitted.

Type: String[
Parameter Sets: AddressStr
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

- **Async**

Starts the write on different threads.

Only for internal use and test purposes.

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

- **InputObject**

The route object where to get the Target information from..

Type: RouteTargetCollection
Parameter Sets: Route
Aliases: Destination, Route

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
-NetId

NetId(s) of the target system.

Type: AmsNetId[]
Parameter Sets: NetIdPort
Aliases: NetIdPort

Required: False
Position: 1
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

-Session

The Session to use for the value read.

Type: ISession[]
Parameter Sets: Session
Aliases: Session

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName)
Accept wildcard characters: False

-SessioinId

Specifies the Session (with unique ID) to use for the value read.

Type: Int32[]
Parameter Sets: SessionId
Aliases: SessionId

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Timeout

The ADS timeout in milliseconds.

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkId=113216).

INPUTS

TwinCAT.RouteTargetCollection

The route object where to get the Target information from..
5.15 Get-TcVersion

SYNOPSIS
Get the TwinCAT Version of a target system.

SYNTAX

NetIdPort (Default)
Get-TcVersion [[-NetId] <AmsNetId[]>] [-Timeout <Int32>] [-Async] [CommonParameters]

Route
Get-TcVersion [-Timeout <Int32>] [-InputObject] <RouteTargetCollection> [-Async] [CommonParameters]

AddressStr
Get-TcVersion [-Timeout <Int32>] [-Address] <String[]> [-Async] [CommonParameters]

Session
Get-TcVersion [-Timeout <Int32>] -Session <ISession[]> [-Async] [CommonParameters]

SessionId
Get-TcVersion [-Timeout <Int32>] -SessionId <Int32[]> [-Async] [CommonParameters]

DESCRIPTION
This Cmdlet gets the TwinCAT version of the specified target version and returns the version object.

EXAMPLES

EXAMPLE 1
PS > Get-TcVersion
Major  Minor  Build  Revision
-----  -----  ------  --------
3      1      4021   50

Get the TwinCAT version of the local system.

EXAMPLE 2
PS > Get-AdsRoute | Get-TcVersion
Major  Minor  Build  Revision
-----  -----  ------  --------
3      1      4021   50
3      1      4021   54

Get the TwinCAT version actual routes.
PARAMETERS

- **Address**

Addresses where to determine the Version information.

The Addresses can consist of NetId, IPAddress or HostName.

Wildcards are permitted.

Type: String[]
Parameter Sets: AddressStr
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

- **Async**

Starts the write on different threads.

Only for internal use and test purposes.

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

- **InputObject**

The target routes where to determine the Version information.

Type: RouteTargetCollection
Parameter Sets: Route
Aliases: Destination, Route

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

- **NetId**

The target address.

Type: AmsNetId[]
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: 1
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

- **Session**

The Session to use for the Cmdlet.
Type: ISession[]
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName)
Accept wildcard characters: False

**-SessionId**

Specifies the Session (with unique ID) to use for the Cmdlet.

Type: Int32[]
Parameter Sets: SessionId
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

**-Timeout**

The Ads timeout in milliseconds.

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkID=113216).

**INPUTS**

**TwinCAT.RouteTargetCollection**

The target routes where to determine the Version information.

**TwinCAT.ISession[]**

The Session to use for the Cmdlet.

## 5.16 New-TcSession

**SYNOPSIS**

Create a new session to a TwinCAT Target.
SYNTAX

**NetIdPort (Default)**

```
New-TcSession [-NetId] <AmsNetId> [-Port] <Int32> [<CommonParameters>]
```

**Route**

```
New-TcSession [-InputObject] <IRoute> [-Port] <Int32> [<CommonParameters>]
```

**AddressStr**

```
New-TcSession [-Provider <String>] [-Address] <String> [-Port] <Int32> [<CommonParameters>]
```

DESCRIPTION

Creates a new Point-To-Point Connection to a TwinCAT Target that is represented by the returned session object.

Different types of Sessions can be accessed by the registered types of SessionProviders (e.g. ADS, MQTT, OPC).

EXAMPLES

**EXAMPLE 1**

```
PS> $route = Get-AdsRoute -Name "Tc3*"
PS> $session = New-TcSession -Route $route -Port 851
PS> $session

ID Address IsConnected EstablishedAt
-- ------- ----------- -------------
 5 172.17.62.105.1.1:851 True 12/12/2016 12:22:02 PM
```

Establishes a new ADS Session/Connection to the specified route destination that has the name pattern "tc3*" via port 851 (PLC1)

**EXAMPLE 2**

```
PS> New-TcSession -NetId '172.17.62.105.1.1' -port 851
PS> $session

ID Address IsConnected EstablishedAt
-- ------- ----------- -------------
 5 172.17.62.105.1.1:851 True 12/12/2016 12:22:02 PM ///
```

Establishes a new Ads Session/Connection to the specified NetId/Port address.

**EXAMPLE 3**

```
PS> New-TcSession -Name 'CX_123456' -port 851
PS> $session

ID Address IsConnected EstablishedAt
-- ------- ----------- -------------
 5 172.17.62.105.1.1:851 True 12/12/2016 12:22:02 PM ///
```

Establishes a new Ads Session/Connection to the target system with the Name/HostName 'CX_123456' (Port 851).

PARAMETERS

**-Address**

The target address of the new session.
This can be the NetId, the HostName or the IPAddress.

**Wildcards are permitted.**

*Type: String*  
*Parameter Sets: AddressStr*  
*Aliases: Name*

Required: True  
Position: 1  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: True

**-InputObject**

The route target object.

*Type: IRoute*  
*Parameter Sets: Route*  
*Aliases: Destination, Route*

Required: True  
Position: 1  
Default value: None  
Accept pipeline input: True (ByValue)  
Accept wildcard characters: False

**-NetId**

The NetID Address

*Type: AmsNetId*  
*Parameter Sets: NetIdPort*  
*Aliases:*

Required: False  
Position: 1  
Default value: 172.17.60.167  
Accept pipeline input: False  
Accept wildcard characters: False

**-Port**

The AmsPort Address of the new session.

*Type: Int32*  
*Parameter Sets: NetIdPort, Route*  
*Aliases:*

Required: True  
Position: 2  
Default value: 10000  
Accept pipeline input: False  
Accept wildcard characters: False

*Type: Int32*  
*Parameter Sets: AddressStr*  
*Aliases:*

Required: False  
Position: 2  
Default value: 10000  
Accept pipeline input: False  
Accept wildcard characters: False

**-Provider**

Selects the session provider registered on the System (ADS by default)
Type: String
Parameter Sets: AddressStr
Aliases:

- Required: False
- Position: Named
- Default value: None
- Accept pipeline input: False
- Accept wildcard characters: False

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkID=113216).

**INPUTS**

TwinCAT.IRoute

The route target object.

**5.17 Read-TcValue**

**SYNOPSIS**

Reads values from TwinCAT devices.

**SYNTAX**

**NetIdPortSymbol** (Default)


**NetIdPortIndexed**


**NetIdPortIndexedTyped**

Read-TcValue [-NetId <AmsNetId[]>] -Port <Int32> [-IndexGroup] <UInt32> [IndexOffset] <UInt32>] [-ValueType] <Type> [-Async] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]

**RouteIndexed**


**RouteIndexedTyped**

Read-TcValue -Route <RouteTargetCollection> -Port <Int32> [-IndexGroup] <UInt32> [IndexOffset] <UInt32>] [-ValueType] <Type> [-Async] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]
### RouteSymbol

```
Read-TcValue -Route <RouteTargetCollection> -Port <Int32> [-Path] <String> [-Extended] [-Async] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [<CommonParameters>]
```

### AddressIndexed

```
Read-TcValue -Address <String[]> -Port <Int32> [-IndexGroup] <UInt32> [-IndexOffset] <UInt32> [-Async] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [<CommonParameters>]
```

### AddressIndexedTyped

```
Read-TcValue -Address <String[]> -Port <Int32> [-IndexGroup] <UInt32> [-IndexOffset] <UInt32> [-ValueType] <Type> [-Async] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [<CommonParameters>]
```

### AddressSymbol

```
```

### SessionIndexed

```
Read-TcValue -Session <ISession[]> [-IndexGroup] <UInt32> [-IndexOffset] <UInt32> [-Size] <Int32> [-Async] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [<CommonParameters>]
```

### SessionIndexedTyped

```
Read-TcValue -Session <ISession[]> [-IndexGroup] <UInt32> [-IndexOffset] <UInt32> [-ValueGroup] <Type> [-Async] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [<CommonParameters>]
```

### SessionSymbol

```
```

### SessionIdIndexed

```
Read-TcValue -SessionId <Int32[]> [-IndexGroup] <UInt32> [-IndexOffset] <UInt32> [-Size] <Int32> [-Async] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [<CommonParameters>]
```

### SessionIdIndexedTyped

```
Read-TcValue -SessionId <Int32[]> [-IndexGroup] <UInt32> [-IndexOffset] <UInt32> [-ValueType] <Type> [-Async] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [<CommonParameters>]
```

### SessionIdSymbol

```
Read-TcValue -SessionId <Int32[]> [-Path] <String> [-Extended] [-Async] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [<CommonParameters>]
```

### InputObject

```
```
DESCRIPTION
This Cmdlet read values from TwinCAT Devices.
The devices can be accessed via different ValueProviders.

EXAMPLES

EXAMPLE 1
PS> $session = New-TcSession -NetId '1.2.3.4.5.6' -Port 851
PS> $symbol = $session | get-TcSymbol -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'
PS> $symbol | Read-TcValue
ADS_DynSymbols

Create an ADS Session/Connection, determine the 'ProjectName' Symbol from the running PLC Project, read the current value of the symbol and print it to the console.

EXAMPLE 2
PS> Read-TcValue -IndexGroup 0x4040 -IndexOffset 0x1247a8 -NetId 172.17.62.105.1.1 -port 851 -size 0xff | format-hex

Reads 256 Bytes via IndexGroup/IndexOffset from the specified target system and prints the out formatted as hexdump.

EXAMPLE 3
PS> Read-TcValue -session $session -IndexGroup 0x4040 -IndexOffset 0x1247a8 -ValueFormat String ADS_DynSymbols

Reads a string typed value from IndexGroup / IndexOffset.
In this example the ProjectName of the running PLC Project resides at that ProcessImage Address.

EXAMPLE 4
PS> $route = Get-AdsRoute -Name 'CX-123456'
PS> $session = $route | New-TcSession -Port 851
PS> $handle = $session | Send-TcReadWrite -IndexGroup SymbolHandleByName -WriteValue "GVL.vgInt" -ReadType Int32 -force
PS> $session | Read-TcValue -IndexGroup SymbolValueByHandle -IndexOffset $handle -ValueFormat Int16

Create a session to the PLC (Port 851) of a target system, determine the SymbolHandle by InstancePath and use this handle to read its 'Int16' Value (INT on PLC System).
PARAMETERS

- **Address**

The Address(es) of the system(s) where to read the value.

The Address can consist of NetId, IPAddress or HostName.

Wildcards are permitted.

<table>
<thead>
<tr>
<th>Type: String[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: AddressIndexed, AddressIndexedTyped, AddressSymbol</td>
</tr>
<tr>
<td>Aliases:</td>
</tr>
<tr>
<td>Required: True</td>
</tr>
<tr>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: None</td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: True</td>
</tr>
</tbody>
</table>

- **Async**

Starts the read on different threads.

Only for test purposes.

<table>
<thead>
<tr>
<th>Type: SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: (All)</td>
</tr>
<tr>
<td>Aliases:</td>
</tr>
<tr>
<td>Required: False</td>
</tr>
<tr>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: False</td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
</tr>
</tbody>
</table>

- **Encoding**

Specifies the Encoding for strings.

The Default is Encoding.Default (ANSI with actual code page)

<table>
<thead>
<tr>
<th>Type: Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: (All)</td>
</tr>
<tr>
<td>Aliases:</td>
</tr>
<tr>
<td>Required: False</td>
</tr>
<tr>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: System.Text.SBCSCodePageEncoding</td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
</tr>
</tbody>
</table>

- **Extended**

Switch on 'ExtendedMode', what means that primitive values are not resolved to their primitive managed (powershell) counterparts, but still contain rich metadata as DynamicValues.

<table>
<thead>
<tr>
<th>Type: SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: NetIdPortSymbol, RouteSymbol, AddressSymbol, SessionSymbol, SessionIdSymbol</td>
</tr>
<tr>
<td>Aliases: FullMetadata</td>
</tr>
<tr>
<td>Required: False</td>
</tr>
<tr>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: False</td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
</tr>
</tbody>
</table>

- **Force**

Force reading value.
This flag bypasses the FailFastInterceptor to retry communication in every case.

**Type:** SwitchParameter  
**Parameter Sets:** (All)  
**Aliases:**

- **Required:** False  
- **Position:** Named  
- **Default value:** False  
- **Accept pipeline input:** False  
- **Accept wildcard characters:** False

---

**-IndexGroup**

The IndexGroup of the Symbol to read from target system.

**Only for IndexGroup/IndexOffset access.**

**Type:** UInt32  
**Parameter Sets:** NetIdPortIndexed, NetIdPortIndexedTyped, RouteIndexed, RouteIndexedTyped, AddressIndexed, AddressIndexedTyped, SessionIndexed, SessionIndexedTyped, SessionIdIndexed, SessionIdIndexedTyped  
**Aliases:** IG

- **Required:** True  
- **Position:** 1  
- **Default value:** 0  
- **Accept pipeline input:** False  
- **Accept wildcard characters:** False

---

**-IndexOffset**

The IndexOffset of the Symbol to read from the target system.

**Only for IndexGroup/IndexOffset access.**

**Type:** UInt32  
**Parameter Sets:** NetIdPortIndexed, NetIdPortIndexedTyped, RouteIndexed, RouteIndexedTyped, AddressIndexed, AddressIndexedTyped, SessionIndexed, SessionIndexedTyped, SessionIdIndexed, SessionIdIndexedTyped  
**Aliases:** IO

- **Required:** False  
- **Position:** 2  
- **Default value:** 0  
- **Accept pipeline input:** False  
- **Accept wildcard characters:** False

---

**-InputObject**

The symbol object to read value from.

**Type:** ISymbol  
**Parameter Sets:** InputObject  
**Aliases:** Symbol

- **Required:** True  
- **Position:** 1  
- **Default value:** None  
- **Accept pipeline input:** True (ByValue)  
- **Accept wildcard characters:** False

---

**-NetId**

The NetId part of the AmsAddress for the value read.

**Type:** AmsNetId[]  
**Parameter Sets:** NetIdPortSymbol, NetIdPortIndexed, NetIdPortIndexedTyped  
**Aliases:**

- **Required:** False  
- **Position:** Named
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

-Path

The instance path of the symbol to read (symbolic access).

This parameter supports wildcards.

Type: String
Parameter Sets: NetIdPortSymbol, RouteSymbol, AddressSymbol, SessionSymbol, SessionIdSymbol
Aliases:
Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-Port

The address Port to use for the value read.

Type: Int32
Parameter Sets: NetIdPortSymbol, NetIdPortIndexed, NetIdPortIndexedTyped, RouteIndexed, RouteIndexed Typed, RouteSymbol, AddressIndexed, AddressIndexedTyped, AddressSymbol
Aliases:
Required: True
Position: Named
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False

-Route

Specifies the target system(s) to read value from.

Type: RouteTargetCollection
Parameter Sets: RouteIndexed, RouteIndexedTyped, RouteSymbol
Aliases: Destination
Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Session

The Session to use for the value read.

Type: ISession[]
Parameter Sets: SessionIndexed, SessionIndexedTyped, SessionSymbol
Aliases:
Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName, ByValue)
Accept wildcard characters: False

-SessionId

Specifies the Session (with unique ID) to use for the value read.

Type: Int32[]
Parameter Sets: SessionIdIndexed, SessionIdIndexedTyped, SessionIdSymbol
Aliases:
**-Size**

The 'Size' of Value (in bytes) to read.

This parameter is available only when Indexed Group without ValueType parameter is performed.

Type: Int32
Parameter Sets: NetIdPortIndexed, RouteIndexed, AddressIndexed, SessionIndexed, SessionIdIndexed
Aliases: ReadSize, Length

Required: True
Position: 3
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

**-Timeout**

Communication Timeout in milliseconds

Type: Int32
Parameter Sets: (All)

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

**-ValueType**

The dataType of the Value for a 'ReadAny' access.

Only usable with IndexGroup/IndexOffset access.

Type: Type
Parameter Sets: NetIdPortIndexedTyped, RouteIndexedTyped, AddressIndexedTyped, SessionIndexedTyped, SessionIdIndexedTyped
Aliases: Type, ReadType

Required: True
Position: 3
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkId=113216).

**INPUTS**

**TwinCAT.ISession[]**

The Session to use for the value read.
Register-AdsHandle

SYNOPSIS
Registers and returns a symbol handle.

SYNTAX

NetIdPortSymbol (Default)
Register-AdsHandle [-NetId <AmsNetId>] [-Port <Int32>] [-Path] <String[]> [CommonParameters]

RouteSymbol
Register-AdsHandle -Route <IRoute> -Port <Int32> [-Path] <String[]> [CommonParameters]

AddressSymbol
Register-AdsHandle -Address <String> -Port <Int32> [-Path] <String[]> [CommonParameters]

SessionSymbol
Register-AdsHandle -Session <ISession> [-Path] <String[]> [CommonParameters]

SessionIdSymbol
Register-AdsHandle -SessionId <Int32> [-Path] <String[]> [CommonParameters]

InputObject
Register-AdsHandle [-InputObject] <ISymbol[]> [CommonParameters]

DESCRIPTION
This Cmdlet registers a symbol handle at the connected system.

The handle is returned as AdsHandleInfo.

EXAMPLES

EXAMPLE 1
PS> $session = New-TcSession -NetId '1.2.3.4.5.6' -Port 851
PS> $handleInfo = $session | get-AdsHandle -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'
PS> $handleInfo = register-AdsHandle -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName' -Session $s
PS> $handleInfo

<table>
<thead>
<tr>
<th>InstancePath</th>
<th>Result Handle</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT_SystemInfoVarList._AppInfo.ProjectName</td>
<td>NoError 0x428000FC (1115685116)</td>
</tr>
</tbody>
</table>

PS> Read-TcValue -Session $session -IndexGroup SymbolValueByHandle -IndexOffset $handleInfo.Handle -Type String
MyProject
Opens a new device session, registers a Symbol Handle to the ProjectName of the running PLC Project, Reads the value by handle unregisters the handle and closes the session again.

**PARAMETERS**

- **Address**

  The Address of the target system where to register the symbol handle.
  
  The Address can consist of RouteName, NetId, IPAddress or HostName.

  Wildcards are permitted.

  Type: String
  Parameter Sets: AddressSymbol
  Aliases:

  Required: True
  Position: Named
  Default value: None
  Accept pipeline input: False
  Accept wildcard characters: True

- **InputObject**

  The symbol object.

  Type: ISymbol[]
  Parameter Sets: InputObject
  Aliases: Symbol

  Required: True
  Position: 1
  Default value: None
  Accept pipeline input: True (ByValue)
  Accept wildcard characters: False

- **NetId**

  The NetId address of the Target system

  Type: AmsNetId
  Parameter Sets: NetIdPortSymbol
  Aliases:

  Required: False
  Position: Named
  Default value: 172.17.60.167.1.1
  Accept pipeline input: False
  Accept wildcard characters: False

- **Path**

  The instance path to the symbol.

  Type: String[]
  Parameter Sets: NetIdPortSymbol, RouteSymbol, AddressSymbol, SessionSymbol, SessionIdSymbol
  Aliases:

  Required: True
  Position: 1
  Default value: None
  Accept pipeline input: False
  Accept wildcard characters: True
-Port

The address Port to use (always in combination with the NetId).

ArgumentCompleter is supported.

<table>
<thead>
<tr>
<th>Type:</th>
<th>Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>NetIdPortSymbol, RouteSymbol, AddressSymbol</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required:</td>
<td>True</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>10000</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

-Route

Specifies the target system.

<table>
<thead>
<tr>
<th>Type:</th>
<th>IRoute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>RouteSymbol</td>
</tr>
<tr>
<td>Aliases:</td>
<td>Destination</td>
</tr>
<tr>
<td>Required:</td>
<td>True</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

-Session

The Session to use (instead of addressing the target system).

<table>
<thead>
<tr>
<th>Type:</th>
<th>ISession</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>SessionSymbol</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required:</td>
<td>True</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>True (ByPropertyName, ByVal)</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

-SessionId

Specifies the Session (with unique ID) to use instead of specifying the target address.

<table>
<thead>
<tr>
<th>Type:</th>
<th>Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>SessionIdSymbol</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required:</td>
<td>True</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>-1</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkId=113216).
INPUTS

_TwinCAT.ISession_

The Session to use (instead of addressing the target system).

_TwinCAT.TypeSystem.ISymbol[]_

The symbol object.

5.19 **Register-AdsNatRoute**

SYNOPSIS

Changes an standard Route to an AmsNAT route on the target system (obsolete).

SYNTAX

**Name (Default)**

```
Register-AdsNatRoute [-Name] <String> -NATNetId <AmsNetId> [-Destination <String>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

**NetId**

```
Register-AdsNatRoute [-NetId] <AmsNetId> -NATNetId <AmsNetId> [-Destination <String>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet Changes an standard Route to an AmsNAT route on the target system.

The route must be preexisting and the cmdlet adds the RemoteNetId/AmsNAT information to the StaticRoutes.xml of the destination system.

Afterwards the destination system needs a TwinCAT Restart.

For TwinCAT Versions \( \geq \) 3.1.4024.11 (or newer), the Add-AdsRoute Cmdlet should be used with the -NAT Parameter as Replacement.

Therefore, this 'Register-AdsNatRoute' Cmdlet is classified as 'obsolete' and of limited use and could be removed in future.

EXAMPLES

**EXAMPLE 1**

```
PS> Register-AdsNatRoute -Name MyRoute -NATNetId 1.2.3.4.2.2
```

Adds an AmsNAT address translation to the existing route 'MyRoute' on the local system (e.g. from '1.2.3.4.1.1' to '1.2.3.4.2.2').

**EXAMPLE 2**

```
PS> Register-AdsNatRoute -NetId 1.2.3.4.1.1 -NATNetId 1.2.3.4.2.2 -Destination CX_1234
```

Adds an AmsNAT address translation to the existing route with NetId '1.2.3.4.1.1' to NATNetId '1.2.3.4.2.2' on System 'CX_1234'.
PARAMETERS

-Destination
The Destination system, where the AmsNAT translation is added.
This Parameter allows RouteName, AmsNetId, IPAddress or HostName
Type: String
Parameter Sets: (All)
Aliases:
Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Name
The Name of the Route where to add an AmsNAT entry.
Type: String
Parameter Sets: Name
Aliases:
Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-NATNetId
The NATNetId (the local representation of the remote system).
Type: AmsNetId
Parameter Sets: (All)
Aliases:
Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-NetId
The NetID which specifies the existing route where to add an AmsNAT entry.
This NetId becomes the 'RemoteNetId' afterwards.
Type: AmsNetId
Parameter Sets: NetId
Aliases:
Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Confirm
Prompts you for confirmation before running the cmdlet.
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf
Required: False
5.20 Remove-AdsNatRoute

SYNOPSIS
Removes an AmsNAT Route from the destination system (obsolete).

SYNTAX
Address (Default)
Remove-AdsNatRoute [-Address <String>] [-Destination <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

Name
Remove-AdsNatRoute [-Name <String>] [-Destination <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

NetId
Remove-AdsNatRoute [-NATNetId <AmsNetId>] [-Destination <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

DESCRIPTION
This Cmdlet removes an AmsNAT Route from the destination system.

It edits just the specified content of the destination system StaticRoutes.xml and needs a TwinCAT Restart afterwards.

Please be aware that the Route is not removed bidirectionally.

This 'Remove-AdsNatRoute' Cmdlet is classified as 'obsolete' and of limited use and could be removed in future.
Please use the `Remove-AdsRoute` Cmdlet instead.

**EXAMPLES**

### EXAMPLE 1

```powershell
PS> Remove-AdsRouteNatRoute -Name MyRoute
```

Removes the AmsNAT route 'MyRoute' from the local system.

### EXAMPLE 2

```powershell
PS> Remove-AdsRouteNatRoute -NATNetId 1.2.3.4.2.2 -Destination CX_1234
```

Removes the AmsNAT Route with Translation NetId '1.2.3.4.2.2' from the system 'CX_1234'

**PARAMETERS**

- **Address**

  The Address of the AmsNAT route to delete.

  This can be the IPAddress or the Dns HostName of the route system.

  ```powershell
  Type: String
  Parameter Sets: Address
  Aliases:
  Required: True
  Position: 0
  Default value: None
  Accept pipeline input: False
  Accept wildcard characters: False
  ```

- **Destination**

  The destination address, where to Remove the specified Mqtt route.

  This can be the NetId, the HostName or the IPAddress.

  ```powershell
  Type: String
  Parameter Sets: (All)
  Aliases:
  Required: False
  Position: Named
  Default value: None
  Accept pipeline input: False
  Accept wildcard characters: False
  ```

- **Name**

  The Name of the AmsNAT route to remove.

  ```powershell
  Type: String
  Parameter Sets: Name
  Aliases:
  Required: True
  Position: 0
  Default value: None
  Accept pipeline input: False
  Accept wildcard characters: False
  ```

- **NATNetId**

  The AmsNAT NetId (translation NetID) of the route to remove.
Type: AmsNetId  
Parameter Sets: NetId  
Aliases:

Required: True  
Position: 0  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

-Confirm

Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter  
Parameter Sets: (All)  
Aliases: cf

Required: False  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

Type: SwitchParameter  
Parameter Sets: (All)  
Aliases: wi

Required: False  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkId=113216).

5.21 Remove-AdsRoute

SYNOPSIS

Remove an ADS Route.

SYNTAX

Address (Default)

Remove-AdsRoute [-Destination <String>] [-Address <String[]>] [-Quiet] [-Credentials <PSCredential>] [-WhatIf] [-Confirm] [<CommonParameters>]
### NetId

Remove-AdsRoute [-Destination <String>] [-NetId] <AmsNetId> [-Quiet] [-Credentials <PSCredential>] [-WhatIf] [-Confirm] [CommonParameters]

### Route

Remove-AdsRoute [-Destination <String>] [-InputObject] <RouteTargetCollection> [-Quiet] [-Credentials <PSCredential>] [-WhatIf] [-Confirm] [CommonParameters]

#### DESCRIPTION

Removes static or temporary routes from the local system or from remote systems. If access is available, the route is removed on both endpoints of the Route.

#### EXAMPLES

**EXAMPLE 1**

```powershell
PS> Get-AdsRoute
Name          NetId           Address        Sub TcVersion RTSystem
----          -----           -------        --- -------- --------
CP-15ECA0     172.17.62.128 1.1 172.17.62.178 0.0 Unknown
TC3TESTA1-CP67X 172.17.62.105 1.1 172.17.62.105 0.0 Unknown
PS> Remove-AdsRoute -Name "CP-15ECA0","TC3TESTA1-CP67X"
```

Removes the Routes "CP-15ECA0" and "TC3TESTA1-CP67X" from the local system.

**EXAMPLE 2**

```powershell
PS> Get-AdsRoute | Remove-AdsRoute -silent
```

Removes all registered routes from the local system.

#### PARAMETERS

- **-Address**
  
  The address for the ADS route to remove.
  
  This can be the NetId, the HostName or the IPAddress.

  Wildcards are permitted.

  Type: String[]
  Parameter Sets: Address
  Aliases: Name

  Required: True
  Position: 0
  Default value: None
  Accept pipeline input: False
  Accept wildcard characters: True

- **-Credentials**
  
  Destination system route credentials (only if removing remotely).

  Type: PSCredential
  Parameter Sets: (All)
  Aliases:

  Required: False
  Position: False
-Destination

The destination address, where to Remove the specified route.

This can be the NetId, the HostName or the IPAddress

Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-InputObject

A collection of routes to remove (Pipeline support).

Type: RouteTargetCollection
Parameter Sets: Route
Aliases:

Required: True
Position: 0
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-NetId

The NetID of the route to remove.

Type: AmsNetId
Parameter Sets: NetId
Aliases:

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Quiet

The Quiet parameter suppresses the 'ShouldProcess' message and the routes will be removed without further question.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Confirm

Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf
Remove-MqttRoute

SYNOPSIS
Remove a MQTT Route.

SYNTAX

DESCRIPTION
Removes a MQTT Route of the specified system.

EXAMPLES

EXAMPLE 1
PS> Remove-MqttRoute -address 1.2.3.4 -port 42
PARAMETERS

-Address
The IPAddress or HostName of the Mqtt broker system to remove.

Type: String
Parameter Sets: (All)
Aliases:

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Destination
The destination address, where to Remove the specified Mqtt route.

This can be the NetId, the HostName or the IPAddress

Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Port
The AmsNetID for the ADS route to add.

Type: Int32
Parameter Sets: (All)
Aliases:

Required: True
Position: 1
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False

-Confirm
Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-WhatIf
Shows what would happen if the cmdlet runs.

The cmdlet is not run.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

CommonParameters
This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkID=113216).

5.23 Send-TcReadWrite

SYNOPSIS
Sends a Read/Write access to ADS Server / TwinCAT Devices.

SYNTAX
NetIdPortIndexed (Default)
Send-TcReadWrite -IndexGroup <UInt32> [-IndexOffset <UInt32>] [-WriteValue <Object>] [-WriteLength <Int32>] [-ReadLength <Int32>] [-ReadType <Type>] -NetId <AmsNetId[]> -Port <Int32> [-Encoding <Encoding>] [-Async] [-Force] [-WhatIf] [-Confirm] [<CommonParameters>]

AddressIndexed

RouteIndexed
Send-TcReadWrite -IndexGroup <UInt32> [-IndexOffset <UInt32>] [-WriteValue <Object>] [-WriteLength <Int32>] [-ReadLength <Int32>] [-ReadType <Type>] -Route <RouteTargetCollection> -Port <Int32> [-Encoding <Encoding>] [-Async] [-Force] [-WhatIf] [-Confirm] [<CommonParameters>]

SessionIndexed

SessionIdIndexed
Send-TcReadWrite -IndexGroup <UInt32> [-IndexOffset <UInt32>] [-WriteValue <Object>] [-WriteLength <Int32>] [-ReadLength <Int32>] [-ReadType <Type>] -SessionId <Int32[]> [-Encoding <Encoding>] [-Async] [-Force] [-WhatIf] [-Confirm] [<CommonParameters>]

DESCRIPTION
This Cmdlet Read/Writes values from/to TwinCAT Devices and works with different ValueProviders.
Because this is a low level data access, only IndexGroup/IndexOffset addressing is available.

IMPORTANT: Sending Read/Write commands should be done with highest care because it could destabilize the TwinCAT System when the write operation is not addressed properly.

To enhance secure operation, the user is enforced to use Length parameters in conjunction with the in/out values which will be checked by the Cmdlet.

The highest attention should also be taken with the IndexGroup/IndexOffset because that represents the Address in the Process Image and cannot be checked by principle.

To prevent that process image overwrites important data by accident please use the -WhatIf and -Confirm parameters whenever it is appropriate and inform about the $ConfirmPreference settings (PS> get-help about_Preference_Variables) before usage of the Send-TcReadWrite Cmdlet.

EXAMPLES

EXAMPLE 1

> Send-TcReadWrite -NetId 1.2.3.4.5.6 -Port 851 -IndexGroup SymbolValueByName -IndexOffset 0 -WriteValue "TwinCAT_SystemInfoVarList._AppInfo.ProjectName" -ReadType string -ReadLength 1024

ReadWrite access of process image on target '1.2.3.4.5.6:851':
Start ReadWrite operation WriteData: 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'(IG:0xf004,IO:0x0000,Len:47),
Read: Type 'System.String' (Len:'1024) on target '1.2.3.4.5.6:851'?
[Y] Yes[A] Yes to All[N] No[L] No to All[S] Suspend[?] Help(default is "Y"): y
ADS_DynSymbols

Sends a Read/Write request with index group 0xf004 (SymbolValueByName) and offset 0.

The write data will be initialized with the project symbol path and an returned (read) string (Default encoded) returned.

EXAMPLE 2

> Send-TcReadWrite -NetId 1.2.3.4.5.6 -Port 851 -IndexGroup SymbolValueByName -WriteValue "TwinCAT_SystemInfoVarList._AppInfo.ProjectName" -ReadLength 64 | format-hex

ReadWrite access of process image on target '1.2.3.4.5.6:851':
Start ReadWrite operation WriteData: 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'(IG:0xf004,IO:0x0000,Len:47),
Read: Type 'System.Byte[]' (Len:'64) on target '1.2.3.4.5.6:851'?
[Y] Yes[A] Yes to All[N] No[L] No to All[S] Suspend[?] Help(default is "Y"): y

<table>
<thead>
<tr>
<th>00</th>
<th>01</th>
<th>02</th>
<th>03</th>
<th>04</th>
<th>05</th>
<th>06</th>
<th>07</th>
<th>08</th>
<th>09</th>
<th>0A</th>
<th>0B</th>
<th>0C</th>
<th>0D</th>
<th>0E</th>
<th>0F</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>01</td>
<td>02</td>
<td>03</td>
<td>04</td>
<td>05</td>
<td>06</td>
<td>07</td>
<td>08</td>
<td>09</td>
<td>0A</td>
<td>0B</td>
<td>0C</td>
<td>0D</td>
<td>0E</td>
<td>0F</td>
</tr>
<tr>
<td>41</td>
<td>44</td>
<td>53</td>
<td>5F</td>
<td>44</td>
<td>79</td>
<td>6E</td>
<td>53</td>
<td>79</td>
<td>6D</td>
<td>62</td>
<td>6F</td>
<td>6C</td>
<td>73</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
</tbody>
</table>

Sends a Read/Write request with index group 0xf004 (SymbolValueByName) and offset 0.

The write data will be initialized with the project symbol path and the returned (read) data is by default a byte array of 64 bytes.

The result value will be formatted as hex code.

EXAMPLE 3

PS> $route = Get-AdsRoute -Name 'CX-123456'
PS> $session = $route | New-TcSession -Port 851
PS> $handle = $session | Send-TcReadWrite -IndexGroup SymbolHandleByName -WriteValue "GVL.vgInt" -ReadType Int32 -force
PS> $session | Read-TcValue -IndexGroup SymbolValueByHandle -IndexOffset $handle -ValueType Int16
42
Create a session to the PLC (Port 851) of a target system, determine the SymbolHandle by InstancePath and use this handle to read its 'Int16' Value (INT on PLC System).

PARAMETERS

- **-Address**
  The target address where to read/write the Value.
  The Address can consist of RouteName, NetId, HostName or IPAddress.
  Wildcards are permitted.

  **Type:** String[]
  **Parameter Sets:** AddressIndexed
  **Aliases:**
  **Required:** True
  **Position:** Named
  **Default value:** None
  **Accept pipeline input:** False
  **Accept wildcard characters:** True

- **-Async**
  Starts the write on different threads.
  Only for internal use and test purposes.

  **Type:** SwitchParameter
  **Parameter Sets:** (All)
  **Aliases:**
  **Required:** False
  **Position:** Named
  **Default value:** False
  **Accept pipeline input:** False
  **Accept wildcard characters:** False

- **-Encoding**
  Specifies the Encoding for strings.
  The Default is Encoding.Default (ANSI with actual code page)

  **Type:** Encoding
  **Parameter Sets:** (All)
  **Aliases:**
  **Required:** False
  **Position:** Named
  **Default value:** System.Text.SBSCCodePageEncoding
  **Accept pipeline input:** False
  **Accept wildcard characters:** False

- **-Force**
  Suppress the 'ShouldProcess' message and forces the write.

  **Type:** SwitchParameter
  **Parameter Sets:** (All)
  **Aliases:**
  **Required:** False
  **Position:** Named
  **Default value:** False
  **Accept pipeline input:** False
  **Accept wildcard characters:** False
### -IndexGroup

IndexGroup of the Value to ReadWrite, only for IndexGroup/IndexOffset access.

**IMPORTANT:** Please be aware, that writing data via IndexGroup/IndexOffset can overwrite data in the ProcessImage and possibly destabilizes the system.

No validity check is done for the symbol alignment and therefore this should be done with highest care!

<table>
<thead>
<tr>
<th>Type:</th>
<th>UInt32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases:</td>
<td>IG</td>
</tr>
<tr>
<td>Required:</td>
<td>True</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>0</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

### -IndexOffset

IndexOffset of the Value to write, only for IndexGroup/IndexOffset access.

**IMPORTANT:** Please be aware, that writing data via IndexGroup/IndexOffset simply overwrites data in the ProcessImage and can destabilize the system.

No validity check is done for the symbol alignment and therefore this should be done with highest care!

If applicable writing data via symbolic information should be preferred!

<table>
<thead>
<tr>
<th>Type:</th>
<th>UInt32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases:</td>
<td>IO</td>
</tr>
<tr>
<td>Required:</td>
<td>False</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>0</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

### -NetId

The ADS target NetID(s) of the system(s) where to read/write the Value.

More than one target will be supported.

When not specified, this argument defaults to AmsNetId.Local.

<table>
<thead>
<tr>
<th>Type:</th>
<th>AmsNetId[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>NetIdPortIndexed</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required:</td>
<td>False</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>172.17.60.167.1.1</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

### -Port

The Port, where to read/write the value.

This Parameter is used in combination with the NetId, Route or Address input parameter.

<table>
<thead>
<tr>
<th>Type:</th>
<th>Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>NetIdPortIndexed, AddressIndexed, RouteIndexed</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required:</td>
<td>True</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>10000</td>
</tr>
</tbody>
</table>
**-ReadLength**

The Length of the data that will be read from the process image.

By default, when not specifying this parameter the marshalling size of the -ReadType parameter will be taken.

This -ReadLength parameter is only helpful when the marshalling size cannot be determined from the read type (e.g. `byte[]`)

**Type:** Int32  
**Parameter Sets:** (All)  
**Aliases:**

Required: False  
Position: Named  
Default value: -1  
Accept pipeline input: False  
Accept wildcard characters: False

**-ReadType**

Use the ReadType parameter to specify the Read/Return type of the data.

If not used, this cmdlet returns the raw `byte[]` as result.

**Type:** Type  
**Parameter Sets:** (All)  
**Aliases:**

Required: False  
Position: Named  
Default value: System.Byte[]  
Accept pipeline input: False  
Accept wildcard characters: False

**-Route**

The target system (as Route) where to read/write the value.

**Type:** RouteTargetCollection  
**Parameter Sets:** RouteIndexed  
**Aliases:** Destination

Required: True  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

**-Session**

The session object represents the target session where to read/write the value.

**Type:** ISession[]  
**Parameter Sets:** SessionIndexed  
**Aliases:**

Required: True  
Position: Named  
Default value: None  
Accept pipeline input: True (ByPropertyName, ByVal)  
Accept wildcard characters: False
-SessionId

The session ID represents the target session where to read/write the value.

Type: Int32[]
Parameter Sets: SessionIdIndexed
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-WriteLength

The Length of the data that will be overwritten within the process image.

By default the marshal size of the object used in the -WriteValue parameter is taken.

This parameter is used to override the marshal size and can be helpful to secure the write operation - to not overwrite more data then expected.

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

-WriteValue

The value to write.

If no additional Length parameter is set, the Write-TcValue Cmdlet marshals this value to its appropriate size.

To not overwrite data of other symbols within the process image, special attention must be taken (see the Confirm and Whatif parameters).

Type: Object
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Confirm

Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-WhatIf

Shows what would happen if the cmdlet runs.
The cmdlet is not run.

**Type:** SwitchParameter  
**Parameter Sets:** (All)  
**Aliases:** wi

**Required:** False  
**Position:** Named  
**Default value:** None  
**Accept pipeline input:** False  
**Accept wildcard characters:** False

### CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkID=113216).

### INPUTS

**TwinCAT.ISession[]**

The session object represents the target session where to read/write the value.

### 5.24 Set-AdsState

#### SYNPLOSIS

Set the ADS State of a TwinCAT Target.

#### SYNTAX

**NetIdPort (Default)**

[-Timeout <Int32>] [<CommonParameters>]

**Route**

[-Timeout <Int32>] [<CommonParameters>]

**AddressStr**

[<CommonParameters>]

#### DESCRIPTION

This Cmdlet sets the ADS State of the specified TwinCAT Targets (E.g. Start / Stop / Config / Reconfig)
EXAMPLES

EXAMPLE 1
PS > Set-AdsState Config -Reinitialize

EXAMPLE 2
PS > Set-AdsState Run 1.2.3.4,CX_0130C7

EXAMPLE 3
PS > get-AdsRoute | set-AdsState Config

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
<th>OK</th>
<th>Time (ms)</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>work-nb2</td>
<td>Config</td>
<td>True</td>
<td>0</td>
<td>172.17.242.54.1.1</td>
</tr>
<tr>
<td>CX_0130C7</td>
<td>Config</td>
<td>True</td>
<td>0</td>
<td>5.1.48.199.1.1</td>
</tr>
</tbody>
</table>

PARAMETERS

- **Address**
  The address of the system where to set the state.
  This can be the RouteName, NetId, the HostName or the IPAddress.
  Wildcards are permitted.

  Type: String[]
  Parameter Sets: AddressStr
  Aliases:
  Required: True
  Position: 1
  Default value: None
  Accept pipeline input: False
  Accept wildcard characters: True

- **Force**
  Forces the command (no questions asked, ResetFailFastHandler)

  Type: SwitchParameter
  Parameter Sets: (All)
  Aliases:
  Required: False
  Position: Named
  Default value: False
  Accept pipeline input: False
  Accept wildcard characters: False

- **InputObject**
  Target route(s), where to set the state.

  Type: RouteTargetCollection
  Parameter Sets: Route
  Aliases: Destination
  Required: True
  Position: 1
  Default value: None
  Accept pipeline input: True (ByValue)
  Accept wildcard characters: False
-NetId
The NetId address where to set the state (Local system by default)

Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: 1
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

-Port
The AmsPort where to set the state (Port 1000, SystemService by default)

Type: Int32
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: 2
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False

-Quiet
Sets the Quiet mode of the command.
The Cmdlet then returns a $true or $false but not the actual states of the targets.
The return value will be $true if at least one SetState operations succeed and it will be $false if have failed.

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Reinitialize
Reinitializes the target system before it is set to the target state (goes over STOP)

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-State
The state/value to set.
Possible values: Invalid, Idle, Reset, Init, Start, Run, Stop, SaveConfig, LoadConfig, PowerFailure, PowerGood, Error, Shutdown, Suspend, Resume, Config, Reconfig, Stopping, Incompatible, Exception, Maxstates

Type: AdsState
Parameter Sets: (All)
Aliases:
Accepted values: Invalid, Idle, Reset, Init, Start, Run, Stop, SaveConfig, LoadConfig, PowerFailure, PowerGood, Error, Shutdown, Suspend, Resume, Config, Reconfig, Stopping, Incompatible, Exception, Maxstates

Required: True
Position: 0
Default value: Invalid
Accept pipeline input: False
Accept wildcard characters: False

- **Timeout**

Communication timeout in ms.

<table>
<thead>
<tr>
<th>Type</th>
<th>Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>(All)</td>
</tr>
</tbody>
</table>

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkId=113216).

**INPUTS**

**TwinCAT.RouteTargetCollection**

Target route(s), where to set the state.

**5.25 Test-AdsRoute**

**SYNOPSIS**

Test the specified route connection.

**SYNTAX**

**AddressStr (Default)**


NetId


**Route**

DESCRIPTION
This Cmdlet establishes a connection to the specified target system and tests if the connection is working.
A Port scan can be executed.

EXAMPLES

EXAMPLE 1
PS > Test-AdsRoute -Port 851

PARAMETERS

-DefaultPorts
Tests all default ports.
The following ports will be tested: 10000, 300, 301, 302, 303, 304, 305, 801, 811, 821, 831, 850, 851, 852, 853, 854, 855, 19200 The 'DefaultPorts' switch overrides the 'Port' parameter.

  Type: SwitchParameter
  Parameter Sets: (All)
  Aliases:
  
  Required: False
  Position: Named
  Default value: False
  Accept pipeline input: False
  Accept wildcard characters: False

-InputObject
The routes targets to test with this Cmdlet.

  Type: RouteTargetCollection
  Parameter Sets: Route
  Aliases: Destination
  
  Required: True
  Position: 1
  Default value: None
  Accept pipeline input: True (ByValue)
  Accept wildcard characters: False

-Mode
The Ping Strategy (PingStrategy.Ads by default)

  Possible values: IP, HostName, IPOrHostName, Ads

  Type: PingStrategy
  Parameter Sets: (All)
  Aliases:
  Accepted values: IP, HostName, IPOrHostName, Ads
  
  Required: False
  Position: Named
  Default value: Ads
  Accept pipeline input: False
  Accept wildcard characters: False

-Name
The name(s) or address(es) of the systems to test.
These can consist of RouteName, NetID, HostName or IPAddress.
Wildcards are permitted.
Type: String[]
Parameter Sets: AddressStr
Aliases: Address

Required: False
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-NetId
The NetId(s) of the target system to test (AmsNetId.Local by default)
Type: AmsNetId[]
Parameter Sets: NetId
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-OnlinePorts
Determines all active/online ports from the target and tests them.
The 'OnlinePorts' switch overrides the 'DefaultPorts' and 'Port' parameters.
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Port
The Port(s) of the target system to test.
Type: Int32[]
Parameter Sets: (All)
Aliases:

Required: False
Position: 2
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Quiet
The Quiet mode.
Returns a boolean only ($true, if one ping succeeded and $false if all failed)
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
- **TimeToLive**

   The TimeToLive value.

   The is the timeout how long the Cmdlet waits for the answer.

<table>
<thead>
<tr>
<th>Type: Int32</th>
<th>Parameter Sets: (All)</th>
<th>Aliases: TTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required: False</td>
<td>Position: Named</td>
<td>Default value: 2</td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
<td>Accept wildcard characters: False</td>
<td></td>
</tr>
</tbody>
</table>

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkID=113216).

**INPUTS**

**System.String[]**

The name(s) or address(es) of the systems to test.

These can consist of RouteName, NetID, HostName or IPAddress.

Wildcards are permitted.

**TwinCAT.Ads.AmsNetId[]**

The NetId(s) of the target system to test (AmsNetId.Local by default)

**TwinCAT.RouteTargetCollection**

The routes targets to test with this Cmdlet.

### 5.26 Unregister-AdsHandle

**SYNOPSIS**

Unregisters a symbol handle.

**SYNTAX**

**NetIdPortHandle (Default)**

Unregister-AdsHandle [-NetId <AmsNetId>] -Port <Int32> [-Handle] <UInt32[]> [CommonParameters]

**NetIdPortInfo**

Unregister-AdsHandle [-NetId <AmsNetId>] -Port <Int32> [-InputObject] <AdsHandleInfo[]> [CommonParameters]

**RouteHandle**

Unregister-AdsHandle -Route <IRoute> -Port <Int32> [-Handle] <UInt32[]> [CommonParameters]
**DESCRIPTION**

This Cmdlet unregisters an already registered symbol handle from the target system.

The Cmdlet supports raw `\[uint\]` handles or AdsHandleInfo objects.

**EXAMPLES**

**EXAMPLE 1**

```
PS> $session = New-TcSession -NetId '1.2.3.4.5.6' -Port 851
PS> $handleInfo = $session | get-AdsHandle -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'
PS> $handleInfo = register-AdsHandle -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName' -Session $s
PS> $handleInfo
<table>
<thead>
<tr>
<th>InstancePath</th>
<th>Result</th>
<th>Handle</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT_SystemInfoVarList._AppInfo.ProjectName</td>
<td>NoError</td>
<td>0x428000FC (1115685116)</td>
</tr>
</tbody>
</table>
PS> Read-TcValue -Session $session -IndexGroup SymbolValueByHandle -IndexOffset $handleInfo.Handle -Type String
PS> MyProject
PS> $handle | Unregister-AdsHandle -Session $session
PS> $session | Close-tcsession
```

Opens a new device session, registers a Symbol Handle to the ProjectName of the running PLC Project, Reads the value by handle unregisters the handle and closes the session again.
PARAMETERS

-Address

The target address of the system.

The Address can consist of RouteName, NetId, IPAddress or HostName.

Wildcards are permitted and ArgumentCompleter is supported.

Type: String
Parameter Sets: AddressHandle, AddressInfo
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-Handle

The instance path of the symbol to read (symbolic access).

This parameter supports wildcards.

Type: UInt32[]
Parameter Sets: NetIdPortHandle, RouteHandle, AddressHandle, SessionHandle, SessionIdHandle
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-InputObject

The AdsHandleInfo object (produced by Register-AdsHandle Cmdlet)

Type: AdsHandleInfo[]
Parameter Sets: NetIdPortInfo, RouteInfo, AddressInfo, SessionInfo, SessionIdInfo
Aliases: HandleInfo

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-NetId

The NetId part of the device target address.

Type: AmsNetId
Parameter Sets: NetIdPortHandle, NetIdPortInfo
Aliases:

Required: False
Position: Named
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

-Port

The address Port to use.

ClearText names for the Port and ArgumentCompleter are supported.
Type: Int32  
Required: True  
Position: Named  
Default value: 10000  
Accept pipeline input: False  
Accept wildcard characters: False

**-Route**

Specifies the target system.

Type: IRoute  
Required: True  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

**-Session**

The Session object (instead of specifying the target system address).

Type: ISession  
Required: True  
Position: Named  
Default value: None  
Accept pipeline input: True (ByPropertyName, ByValue)  
Accept wildcard characters: False

- **SessionId**

Specifies the Session (with unique ID) to use (instead of specifying the address).

Type: Int32  
Required: True  
Position: Named  
Default value: -1  
Accept pipeline input: False  
Accept wildcard characters: False

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkID=113216).

**INPUTS**

**TwinCAT.ISession**

The Session object (instead of specifying the target system address).

**TwinCAT.Management.Automation.AdsHandleInfo[]**

The AdsHandleInfo object (produced by Register-AdsHandle Cmdlet)
5.27 Write-TcValue

SYNOPSIS

Write values to TwinCAT devices.

SYNTAX

NetIdPortSymbol (Default)

NetIdPortIndexed

RouteIndexed

RouteSymbol

AddressIndexed

AddressSymbol

SessionIndexed

SessionSymbol

SessionIdIndexed
**SessionIdSymbol**

```powershell
```

**InputObject**

```powershell
```

**DESCRIPTION**

This Cmdlet writes values to TwinCAT Devices.

The devices can be accessed via different ValueProviders.

All sorts of ADS-addressing will be supported by this Cmdlet: Addressing by IndexGroup / IndexOffset (see IndexGroup, IndexOffset parameters) Addressing by Instance Path (see path parameter) Addressing by Symbol (see InputObject / Symbol parameter)

**IMPORTANT:** Writing values should be done with highest care because it could destabilize the TwinCAT System when the write operation is not addressed properly.

While writing with available symbol information is not critical and should be preferred the size and position of symbol data is known within the process image), the access via Instance path is less secure.

The size of the overwritten data is not known and therefore not checked by the Cmdlet.

The highest attention should be taken with write IndexGroup / IndexOffset write operations because beneath the unknown data size even the position of the data is not checked.

The data is written directly into the process image.

**EXAMPLES**

**EXAMPLE 1**

```powershell
PS> $session = New-TcSession -NetId 1.2.3.4.5.6 -Port 851
PS> $projectNameSymbol = $session | Get-TcSymbol -path "*ProjectName"
PS> $projectNameSymbol

<table>
<thead>
<tr>
<th>InstanceName</th>
<th>DataType</th>
<th>Size</th>
<th>InstancePath</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProjectName</td>
<td>STRING(63)</td>
<td>64</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ProjectName</td>
</tr>
</tbody>
</table>

PS> $projectNameSymbol | Read-TcValue

OldProjectName

PS> $projectNameSymbol | Write-TcValue -Value "NewProjectName" -force
PS> $projectNameSymbol | Read-TcValue

NewProjectName
```

This example shows how to create a session, determining the Symbol 'ProjectName within the _AppInfo Struct on a running PLC project and reading its value.

After that, the Value will be overwritten with 'NewProjectName'.

**EXAMPLE 2**

```powershell
Write-TcValue -session $session -IndexGroup 0x4040 -IndexOffset 0x1247A8 -Value "NewProjectName"
```

Writes a string typed Value to the specified IndexGroup/IndexOffset Address.
### PARAMETERS

- **Address**

  The target address where to write the Value.

  The Address can consist of RouteName, NetId, HostName or IPAddress.

  **Wildcards are permitted.**

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter Sets</th>
<th>Aliases</th>
</tr>
</thead>
<tbody>
<tr>
<td>String[]</td>
<td>AddressIndexed, AddressSymbol</td>
<td>Required: True, Position: Named, Default value: None, Accept pipeline input: False, Accept wildcard characters: True</td>
</tr>
</tbody>
</table>

- **Async**

  Starts the write on different threads.

  Only for internal use and test purposes.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter Sets</th>
<th>Aliases</th>
</tr>
</thead>
</table>

- **Encoding**

  Specifies the Encoding for strings.

  The Default is Encoding.Default (ANSI with actual code page)

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter Sets</th>
<th>Aliases</th>
</tr>
</thead>
</table>

- **Force**

  Suppress the 'ShouldProcess' message and forces the write.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter Sets</th>
<th>Aliases</th>
</tr>
</thead>
</table>

- **IndexGroup**

  IndexGroup of the Value to write, only for IndexGroup/IndexOffset access.
IMPORTANT: Please be aware, that writing data via IndexGroup/IndexOffset simply overwrites data in the ProcessImage and can destabilize the system.

No validity check is done for the symbol alignment and therefore this should be done with highest care!

If applicable writing data via symbolic information should be preferred!

-IndexOffset

IndexOffset of the Value to write, only for IndexGroup/IndexOffset access.

IMPORTANT: Please be aware, that writing data via IndexGroup/IndexOffset simply overwrites data in the ProcessImage and can destabilize the system.

No validity check is done for the symbol alignment and therefore this should be done with highest care!

If applicable writing data via symbolic information should be preferred!

-InputObject

The symbol object on which to write the value.

Type: ISymbol
Parameter Sets: InputObject
Aliases: Symbol

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-NetId

The ADS target NetID(s) of the system(s) where to write the Value.

More than one target will be supported.

Type: AmsNetId[]
Parameter Sets: NetIdPortSymbol, NetIdPortIndexed
Aliases:

Required: False
Position: Named
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False
-Path

The instance path to the symbol to write (Symbolic access).

Wildcards are permitted.

Type: String
Parameter Sets: NetIdPortSymbol, RouteSymbol, AddressSymbol, SessionSymbol, SessionIdSymbol
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-Port

The Port, where to write the value.

This Parameter is used in combination with the NetId, Route or Address input parameter.

Type: Int32
Parameter Sets: NetIdPortSymbol, NetIdPortIndexed, RouteIndexed, RouteSymbol, AddressIndexed, AddressSymbol
Aliases: Destination

Required: True
Position: Named
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False

-Route

The target system (as Route) where to write the value.

Type: RouteTargetCollection
Parameter Sets: RouteIndexed, RouteSymbol
Aliases: Destination

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Session

The session object represents the target session where to write the value.

Type: ISession[]
Parameter Sets: SessionIndexed, SessionSymbol
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName, ByValue)
Accept wildcard characters: False

-SessionId

The session ID represents the target session where to write the value.

Type: Int32[]
Parameter Sets: SessionIdIndexed, SessionIdSymbol
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Size

The Length of the data that will be overwritten within the process image.

IMPORTANT: Please be aware, that writing data via IndexGroup/IndexOffset simply overwrites data in the ProcessImage and could destabilize the system.

No further validity check is done for the symbol alignment and therefore this should be done with highest care (best with use of the -Confirm and -Whatif Cmdlet arguments).

If applicable writing data via symbolic information should be preferred!

Type: Int32
Parameter Sets: NetIdPortIndexed, RouteIndexed, AddressIndexed, SessionIndexed, SessionIdIndexed
Aliases: Length, WriteSize

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

-Value

The value to write.

If no additional Length parameter is set, the Write-TcValue Cmdlet marshalls this value to its appropriate size.

To not overwrite data of other symbols within the process image, special attention must be taken (see the Confirm and Whatif parameters).

Type: Object
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Confirm

Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about_CommonParameters](http://go.microsoft.com/fwlink/?LinkID=113216).

**INPUTS**

**TwinCAT.ISession[]**

The session object represents the target session where to write the value.

**TwinCAT.TypeSystem.ISymbol**

The symbol object on which to write the value.
6 TcXaeMgmt Version 5.X

This is the Platform independent version of the ‘TcXaeMgmt’ Module. This can run on all Platforms that are supported by ‘Microsoft Powershell’ and ‘Windows Powershell’ >= Version 5.

Differences between Microsoft Powershell and Windows Powershell are documented [here](#).

Supported TwinCAT Versions are TwinCAT 3.1.4024 and newer. If an older Version of TwinCAT is installed locally, please use the Version 3.X series [13] of the 'TcXaeMgmt' module.

6.1 About TcXaeMgmt

PowerShell TwinCAT XAE Management Console (TcXaeMgmt)

SHORT DESCRIPTION
Cmdlets for managing and accessing ADS Routes, Reading/Writing Values and managing Remote targets.

LONG DESCRIPTION
The Powershell TwinCAT Management Console is a PowerShell module that provides a number of useful cmdlets for TwinCAT System Management and for communicating with ADS devices over the ADS protocol.

This includes the following tasks/features:

- Establishing/Removing Route Connections (**Add-AdsRoute**, **Remove-AdsRoute**)
- Browsing Routes locally and within the network (**Broadcast Search**, **Get-AdsRoute**)
- Getting remote device states and information (**Get-AdsState**, **Get-TcTargetInfo**, **Get-TcVersionInfo**)
- Establishing and Closing Remote communication sessions (**New-TcSession**, **Get-TcSession**, **Close-TcSession**)
- Browsing Symbol Information (**Get-TcSymbol**, **Get-TcDataType**)
- Reading/Writing raw and symbolic values (**Read-TcValue**, **Write-TcValue**, **Send-TcReadWrite**)
- Uploading/Downloading files to/from remote devices (**Copy-AdsFile**)
- Browsing License information (**Get-TcLicense**)

This Module is usable under all Powershell Version >= 5.1 including 'Windows Powershell' and 'Powershell Core' Versions.

As Prerequisite the TcXaeMgmt Module needs a local TwinCAT installation larger equals than TwinCAT 4024.10. There are no limitations to access other/older TwinCAT Versions remotely.

PREREQUISITES

>= TwinCAT 3.1.4024.10 (XAR Runtime or Full) (local installation)

POWERSHELL COMPATIBILITY

>= Windows Powershell 5.1
>= Powershell (Core) 6.0
CMDLETS

To see what cmdlets are provided by the TcXaeMgmt Module, execute the command:

```
PS> Get-Command -Module TcXaeMgmt -CommandType Cmdlet
```

The actual TcXaeMgmt cmdlets are listed below:

**Add-AdsRoute [115]**
Cmdlet for adding TwinCAT Routes.

**Add-MqttRoute [124]**
Adds an MQTT route to the destination system.

**Close-TcSession [128]**
Closes the specified session object.

**Copy-AdsFile [129]**
Uploads / Downloads files from/to TwinCAT target.

**Get-AdsRoute [133]**
List routes on a TwinCAT System / Broadcast search.

**Get-AdsState [136]**
Gets the Ads State of a TwinCAT Target.

**Get-AmsRouterEndpoint [140]**
Get the actual AmsConfiguration / RouterEndpoint of the process.

**Get-EcBoxes [141]**
Gets the EtherCAT Boxes actually loaded to the target system.

**Get-EcFrameStatistics [142]**
Gets the EtherCAT Frame statistics from an ETHERCAT master.

**Get-EcMaster [144]**
Gets the Ads State of a TwinCAT Target.

**Get-IoDevice [146]**
Gets actually loaded IO Devices of the target system.

**Get-IoFreeRun [149]**
Gets the IO FreeRun State of the specified target.

**Get-MqttRoute [151]**
Remove a MQTT Route.
Get-RTimeCpuSettings [152]
Getting the Cpu Settings of the TwinCAT System

Get-RTimeLatency [153]
Get the latency of TwinCAT Realtime Cores of the specified TwinCAT target system.

Get-RTimePerformance [156]
Gets the Realtime Performance of the specified system.

Get-TcDataType [159]
Get the DataTypes from a TwinCAT target system / Device.

Get-TcEvent [162]
Gets TwinCAT events from event logs on local and remote computers.

Get-TcLicense [166]
Get TwinCAT License information.

Get-TcRouterInfo [171]
Gets the router status information of the specified target system.

Get-TcSession [173]
List the currently established Sessions.

Get-TcSymbol [174]
Get the symbols from a TwinCAT target system / Device.

Get-TcTargetInfo [179]
Get TwinCAT Device Target information.

Get-TcVersion [182]
Get the TwinCAT Version of a target system.

New-TcSession [185]
Create a new session to a TwinCAT Target.

Read-TcValue [187]
Reads values from TwinCAT devices.

Register-AdsHandle [194]
Registers and returns a symbol handle.

Register-AdsNatRoute [197]
Changes an standard Route to an AmsNAT route on the target system (obsolete).
Remove-AdsRoute [200]
Remove an ADS Route.

Remove-MqttRoute [203]
Remove a MQTT Route.

Reset-IoFreeRun [205]
Resets the IO FreeRun state on the specified target.

Restart-AdsComputer [208]
Restarts ("reboots") the operating system on local and remote TwinCAT computers.

Send-TcReadWrite [212]
Sends a Read/Write access to ADS Server / TwinCAT Devices.

Set-AdsState [218]
Set the ADS State of a TwinCAT Target.

Set-AmsRouterEndpoint [222]
Sets the AmsConfiguration (Loopback address and port, RouterEndpoint).

Set-IoFreeRun [224]
Sets the IO FreeRun state of the target.

Stop-AdsComputer [227]
Stops (shuts down) local and remote TwinCAT computers.

Test-AdsRoute [231]
Test the specified route connection.

Unregister-AdsHandle [234]
Unregisters a symbol handle.

Write-TcValue [238]
Write values to TwinCAT devices.

EXAMPLES

**Getting Route**

PS> $route = get-adsroute TC3TEST*  
PS> $route

<table>
<thead>
<tr>
<th>Name</th>
<th>NetId</th>
<th>Address</th>
<th>Sub Version</th>
<th>RTSystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC3TESTA1-CP67X</td>
<td>172.17.62.105</td>
<td>172.17.62.105</td>
<td>0.0</td>
<td>Unknown</td>
</tr>
</tbody>
</table>
Create Session

PS> $session = New-TcSession -Route $route -Port 851
PS> $session

ID Address IsConnected EstablishedAt
-- ------- ----------- -------------
1 172.17.62.105.1.1:851 True 12/12/2016 12:22:02 PM

Read Ads Value (Struct)

PS> $v1 = Read-TcValue -SessionId 1 -Path "GVL.vgStruct"
PS> $v1

vBool : True
vByte : 123
vWord : 12345
vDWord : 12345678
vSInt : -121
vUSInt : 212
vInt : -12121
vUInt : 21212
vDInt : -1212121
vUDInt : 2121212
vReal : 123,456
vLReal : 1234567890,12346
vString : QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;:_
vTime : 01:02:03.0040000
vTod : 23:45:06.7890000
vDate : 17.11.2005 00:00:00
vDT : 17.11.2005 12:34:56
vAlias : 8
vEnum : 8
vRange : 7
PSValue : ...

Read Ads Value (Boolean)

PS> $v2 = Read-TcValue -SessionId 1 -Path "Main.bChange"
PS> $v2

False

Read Ads Value (Array of Strings)

PS> $v3 = Read-TcValue -SessionId 1 -Path "GVL.vgaString"

Dimensions Elements
---------- --------
{Twincat.TypeSystem.Dimension} {QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;:_;_:;MNRV...}
...

Read Array Of Structs

PS> $v4 = Read-TcValue -SessionId 1 -Path "GVL.vgastruct"

Dimensions Elements
---------- --------
{Twincat.TypeSystem.Dimension} @{vBool=True; vByte=123; vWord=12345; vDWord=12345678; vSInt=-121; vUSInt=212; vInt=-12121; vUInt=21212; vDInt=-1212121; vUD...}

Dump Array Elements

PS> $v4.Dimensions.ElementCount
2
PS> $v4.Elements
vBool : True
vByte : 123
vWord : 12345
vDWord : 12345678
vSInt : -121
vUSInt : 212
vInt : -12121
vUInt : 21212
vReal : 123.456
vLReal : 1234567890.12346
vString : QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;_:;

vTime : 01:02:03.0040000
vTod : 23:45:06.7890000
vDate : 17.11.2005 00:00:00
vDT : 17.11.2005 12:34:56
vAlias : 8
vEnum : 8
vRange : 8
PSValue : ...

vBool : False
vByte : 234
vWord : 23456
vDWord : 23456789
vSInt : 121
vUSInt : 131
vInt : 12121
vUInt : 13131
vDInt : 1212121
vUDInt : 1313131
vReal : 456,321
vLReal : 987654321,123457
vString : _:;MNBVCXYÄÖLKJHGFDSAÜPOIUZTREWQ

## Browse Data Types (Query by Category)

```powershell
PS> $session | Get-TcDataType | where Category -eq "Array"
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Category</th>
<th>Comment</th>
<th>ElementType</th>
<th>Dimensions</th>
<th>Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARRAY [-1..1] OF INT</td>
<td>6</td>
<td>Array</td>
<td>INT</td>
<td>TwinCAT.Type...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARRAY [-10..-8] OF BOOL</td>
<td>3</td>
<td>Array</td>
<td>BOOL</td>
<td>TwinCAT.Type...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARRAY [0..1] OF A_Alias</td>
<td>4</td>
<td>Array</td>
<td>A_Alias</td>
<td>TwinCAT.Type...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Browse DataTypes by name

```powershell
PS> $session | Get-TcDataType -name "Array"
```

Browse all Symbols recursively

```powershell
PS> $session | Get-TcSymbol -recurse
```

... returns all symbols

**Browse Symbols recursively by Symbol Path (Here specific array index ’TaskInfo[1]’)**

```powershell
PS> $session | Get-TcSymbol -recurse -path "\"TaskInfo[1]\"", ".ProjectName"
```

<table>
<thead>
<tr>
<th>InstanceName</th>
<th>Comment</th>
<th>DataType</th>
<th>Size</th>
<th>InstancePath</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProjectName</td>
<td></td>
<td>STRING(63)</td>
<td>64</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ProjectName</td>
</tr>
<tr>
<td>ObjId</td>
<td></td>
<td>UTCID</td>
<td>4</td>
<td>TwinCAT_SystemInfoVarList._TaskInfo[1].ObjId</td>
</tr>
<tr>
<td>CycleTime</td>
<td></td>
<td>UDINT</td>
<td>4</td>
<td>TwinCAT_SystemInfoVarList._TaskInfo[1].CycleTime</td>
</tr>
</tbody>
</table>

112
Version: 1.2
TE1000
Browse only Symbols ending with path *.ProjectName

```powershell
PS> $project = Get-TcSymbol -Session $session -recurse -path "*.ProjectName"
```

<table>
<thead>
<tr>
<th>InstanceName</th>
<th>DataType</th>
<th>Size</th>
<th>InstancePath</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProjectName</td>
<td>STRING(63)</td>
<td>64</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ProjectName</td>
<td></td>
</tr>
</tbody>
</table>

**Ads Read ProjectName**

```powershell
PS> $project | Read-TcValue -Session $session
```

**Ads Write ProjectName**

```powershell
PS> $project | Write-TcValue -Session $session -Value "NewProjectName"
PS> $project | Read-TcValue -Session $session NewProjectName
```

**ReadWrite by Symbol Path**

```powershell
PS> Read-TcValue -SessionId 1 -Path "Main.bChange" false
PS> Write-TcValue -SessionId 1 -Symbol "Main.bChange" -Value True
PS> Read-TcValue -SessionId 1 -Path "GVL.vgBool"
PS> Write-TcValue -SessionId 1 -Path "GVL.vgBool" -value $true
```

**ReadWrite by Piping**

```powershell
PS> $projectNameSymbol = $session | Get-TcSymbol -Recurse -path "*ProjectName"
PS> $projectNameSymbol | Read-TcValue -SessionId 1
PS> $projectNameSymbol | Write-TcValue -SessionId 1 -Value "NewProjectName"
PS> $projectNameSymbol | Read-TcValue -SessionId 1
```

**Get Target Information**

```powershell
PS> get-adsroute | Get-TcTargetInfo
```

```powershell
Target | Version | Level OS | Image | Device | CPUArch
-------- | ------- | --------- | ------ | ------- | -------
TC3TESTA1-CP67X | 3.1.4021.131 CP | Win7 | IntelX86
```

```powershell
PS> get-adsroute | Get-TcVersion
```

```powershell
Major Minor Build Revision
---- ---- ---- ------
3    1    4021 131
```

**PROVIDERS**

The TcXaeMgmt module includes the AdsSymbolProvider and the AdsFileProvider
AdsSymbolProvider

Binds the target device symbolic information to a PSDrive. To register a symbol server as

PSDrive type (here the Target Route 'CX_01234' with AmsPort: 851)

PS> New-PSDrive -Name CX_01234_Symbols -PSProvider AdsSymbolProvider -Address CX_01234 -Port 851 -Root ''
PS> cd CX_01234_Symbols:
PS CX_01234_Symbols:> dir

AdsFileProvider

PS> New-PSDrive -name CX_01234 -PSProvider AdsFileProvider -Address CX_01234 -Root ''
PS> cd CX_01234:
PS> dir

FEEDBACK

Please submit any feedback, including defects and enhancement requests, to

support@beckhoff.com

We are also interested in suggestions you may have for cmdlets. Over time, we hope to be able to add some more features.

NOTE

To see what functions are provided by TcXaeMgmt, execute the command:

PS> Get-Command -Module TcXaeMgmt -CommandType Function

For more information, most of the cmdlets have help associated with them e.g.:

PS> Get-Help Add-AdsRoute -full

The definitive information on a cmdlet's parameters can be obtained by executing:

PS> Get-Command Add-AdsRoute -syntax

or more tersely:
SEE ALSO

Documentation TcXaeMgmt Module

About the TcXaeMgmt Module

Beckhoff Homepage

PS> get-help about_providers

KEYWORDS

• ADS
• TwinCAT
• ManagementConsole
• Routes

6.2 Add-AdsRoute

SYNOPSIS

Cmdlet for adding TwinCAT Routes.

SYNTAX

Routes (Default)
[-Confirm] [<CommonParameters>]

Address
[-Confirm] [<CommonParameters>]

AddressPSK
[<CommonParameters>]

AddressSCA
[-Unidirectional] [-Quiet] [-Nat <AmsNetId>] [-WhatIf] [-Confirm] [<CommonParameters>]

AddressSSC
DESCRIPTION

Adds a Route to the destination target System (Temporary or static).

EXAMPLES

EXAMPLE 1
> Get-AdsRoute -All -name "Tc3*"

Name    NetId    Address    Sub TcVersion RTSystem
Search for Systems that start with the name "TC3*", then asks the user for Credentials and adds the Route as 'temporary' (with TC2 compatible security, clear text password).

Afterwards, the connection is checked.

**EXAMPLE 2**

```bash
> Get-AdsRoute -All -name "Tc3*"

Name          NetId          Address        Sub TcVersion RTSystem
--           ------          -------         --- --------- --------
TC3TestA1-CP67x 172.17.62.105.1.1 172.17.62.105 3.1.4021 Win7
TC3Test13-C6650 172.17.60.239.1.1 172.17.62.156 2.11.2246 Win7

PS> $cred = Get-Credential -Message "Get Credentials" -UserName "UserName"
PS> Add-AdsRoute -Credential $cred -name "TC3TestA1-CP67x" -selfSigned

Name          NetId          Address        Sub TcVersion RTSystem
--           ------          -------         --- --------- --------
TC3TestA1-CP67x 172.17.62.105.1.1 172.17.62.105 3.1.4021 Win7

PS> Get-AdsRoute -name "TC3TestA1-CP67x" | Test-AdsRoute
```

Search for Systems that start with the name "TC3*", then asks the user for Credentials and adds the Route with 'SelfSigned' AdsSecure settings.

Afterwards, the connection is checked.

**EXAMPLE 3**

```bash
PS> Add-AdsRoute -Address 172.17.62.105 -sca

Name          NetId          Address        Sub TcVersion RTSystem
--           ------          -------         --- --------- --------
TC3TestA1-CP67x 172.17.62.105.1.1 172.17.62.105 3.1.4021 Win7

PS> $cred = Get-Credential -Message "Get Credentials" -UserName "UserName"
PS> Add-AdsRoute -Credential $cred -NetId 172.17.62.105 -Nat 1.2.3.4.1.1

Name          NetId          Address        Sub TcVersion RTSystem
--           ------          -------         --- --------- --------
TC3TestA1-CP67x 1.2.3.4.1.1 172.17.62.105 3.1.4024 Win10 (2004)
```

Add a route with a local network address translation (NAT AmsNetId) to project a remote AmsNetId (RemoteNetId) locally to a different address.
EXAMPLE 5

Add-AdsRoute -name "TestRoute" -NetId 1.2.3.4.1.1 -IPOrHostName 1.2.3.4 -Temporary -RemotePersistance None

Adding a route 'TestRoute' single sided and temporary only to the local system.

The remote device doesn't need to be online.

PARAMETERS

- **Name**

The name of the route(s) to add.

If the Routes address is ambiguous and more than one route will be found online for adding then the route names will be numbered to be distinct.

Without setting this parameter, the default route name will be its Computename / Hostname.

Type: String
Parameter Sets: Address, AddressPSK, AddressSCA, AddressSSC, NetId, NetIdPSK, NetIdSCA, NetIdSSC
Aliases: 

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

- **Address**

The address for the ADS route.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

Type: String[]
Parameter Sets: Address, AddressPSK, AddressSCA, AddressSSC
Aliases: TargetAddress

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

- **NetId**

The AmsNetID for the ADS route to add.

If no further IPAddress or HostName is specified, a broadcast search is triggered to find an online device.

If a single sided route should be added, specify the IPAddress or HostName Parameter in combination with -RemotePersistance:None.

Type: AmsNetId
Parameter Sets: NetId, NetIdPSK, NetIdSCA, NetIdSSC
Aliases: TargetNetId

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
- **IPOrHostName**

The HostName Address of the target route of the IPAddress.

Because this parameter is mandatory, the Add-AdsRoute Cmdlet tries to detect the HostName/IPAddress via a Broadcast search in the Network when it is not specified.

That means it must be available and reachable within the network.

If not the Add-AdsRoute Cmdlet will fail.

If the IP Or HostName is specified, the target availability is necessary and Add-Route will register the Route whatever is specified as address.

Type: String
Parameter Sets: NetId, NetIdPSK, NetIdSCA, NetIdSSC
Aliases:
- Required: False
- Position: Named
- Default value: None
- Accept pipeline input: False
- Accept wildcard characters: False

- **RemotePersistance**

The persistence type of the remote route.

None/Server means no remote route will be created.

Other valid values are 'Static' or 'Temporary'

Possible values: None, Server, Temporary, Static

Type: RoutePersistanceType
Parameter Sets: Address, NetId
Aliases:
- Accepted values: None, Server, Temporary, Static
- Required: False
- Position: Named
- Accept pipeline input: False
- Accept wildcard characters: False

- **Timeout**

(Broadcast) Search Timeout for searching the unregistered target (Default 5000).

Type: Int32
Parameter Sets: Address, AddressPSK, AddressSCA, AddressSSC
Aliases:
- Required: False
- Position: Named
- Default value: 5000
- Accept pipeline input: False
- Accept wildcard characters: False

- **InputObject**

The input Ads Routes.

Type: IRoute[]
Parameter Sets: Routes, RoutesPSK, RoutesSCA, RoutesSSC
Aliases: Route, TargetRoute

- Required: True
- Position: Named
- Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-Destination

The Destination Address, where the route is added.

Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-DestinationCredential

The credentials of the destination system, where to add the route.

Local system by default.

Type: PSCredential
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Credential

Credentials of the route to be added to the destination system.

This parameter is only necessary, when a bidirectional route will be added.

When specifying

IMPORTANT: Please be aware, that in the current version, the password is transferred as clear text through the network.

Use this only in safe subnetworks.

Type: PSCredential
Parameter Sets: Routes, AddressPSK, AddressSSC, NetIdPSK, NetIdSSC, RoutesPSK, RoutesSSC
Aliases: TargetCredential

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

Type: PSCredential
Parameter Sets: Address, NetId
Aliases: TargetCredential

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-BinaryKey

The BinaryKey can be used instead of the credential Password on SecureSettings.PreSharedKeys (PSK).
There is no function for this parameter on other security settings.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: Byte[]</td>
<td>Parameter Sets: AddressPSK, NetIdPSK, RoutesPSK</td>
</tr>
<tr>
<td>Required: False</td>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: None</td>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
<td></td>
</tr>
</tbody>
</table>

**-HostName**

If set, the route will be registered as HostName

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: SwitchParameter</td>
<td>Parameter Sets: (All)</td>
</tr>
<tr>
<td>Required: False</td>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: False</td>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
<td></td>
</tr>
</tbody>
</table>

**-Temporary**

If set, the Route will be registered as temporary route.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: SwitchParameter</td>
<td>Parameter Sets: (All)</td>
</tr>
<tr>
<td>Required: False</td>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: False</td>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
<td></td>
</tr>
</tbody>
</table>

**-SelfSigned**

Gets or sets the SelfSigned (SSC) mode for adding the route.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: SwitchParameter</td>
<td>Parameter Sets: AddressSSC, NetIdSSC, RoutesSSC</td>
</tr>
<tr>
<td>Alias: SSC</td>
<td>Required: True</td>
</tr>
<tr>
<td>Position: Named</td>
<td>Default value: False</td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
<td></td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
<td></td>
</tr>
</tbody>
</table>

**-FingerPrint**

The Fingerprint used for adding the route.

This parameter can be used when the parameter `-SelfSigned` is set.

If specified, the found OnlineTarget will be checked against this fingerprint.

If not specified, the `Add-AdsRoute` Cmdlet doesn't check the fingerprint, always adding the route.

Using the fingerprint means that only single routes can be added, multi-adding routes with fingerprint is not supported.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: String</td>
<td>Parameter Sets: AddressSSC, NetIdSSC, RoutesSSC</td>
</tr>
<tr>
<td>Alias:</td>
<td></td>
</tr>
</tbody>
</table>
-PreSharedKey

Gets or sets the PreSharedKey (PSK) mode for adding the route.

Because a valid certificate is expected at the target, it is not necessary to enter credentials.

Type: SwitchParameter
Parameter Sets: AddressPSK, NetIdPSK, RoutesPSK
Aliases: PSK

Required: True
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-SharedCertAuth

Gets or sets the SharedCertificateAuthority (SCA) mode for adding the route.

Because a valid certificate is expected at the target, it is not necessary to enter credentials.

Type: SwitchParameter
Parameter Sets: AddressSCA, NetIdSCA, RoutesSCA
Aliases: SCA

Required: True
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-IgnoreCN

Gets or sets the 'Ignore Common Name' mode for SharedCertificateAuthority (SCA) while adding the route.

The "CommonName" of the certificate must correspond to the name used when establishing the connection in the certificate.

This behavior can be deactivated by this option.

Type: SwitchParameter
Parameter Sets: AddressSCA, NetIdSCA, RoutesSCA
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Unidirectional

Gets or sets the unidirectional setting.

The Unidirectional setting registers the ADS Route as 'one-way' channel.

That means that the engineering/source system (that's where the route request is initiated) can send requests to the remote target, but not in the opposite direction.

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
-Quiet

The Quiet parameter suppresses the 'ShouldProcess' message and the ADSRoute will be added without further question.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Nat

The 'Nat' parameter sets the local representation of the routes AmsNetId.

All (local) addressing to this netld will be translated to the remote/network AmsAddress of the route.

If using the '-Nat' parameter, the 'Add-AdsRoute' Cmdlet is limited to single route additions.

Multi-adding is not supported.

This Parameter can be used with TwinCAT Versions >= 3.1.4024.11.

Type: AmsNetId
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Confirm

Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute[]
The input Ads Routes.

OUTPUTS

NOTES

6.3 Add-MqttRoute

SYNOPSIS

Adds an MQTT route to the destination system.

SYNTAX

Default (Default)

Identity

Psk

SCA

DESCRIPTION

This Cmdlet adds an MQTT route to the destination system.

To add the route, the Address of a MQTT route must be specified.
EXAMPLES

EXAMPLE 1
> Add-MqttRoute -Address 1.2.3.4 -port 42

Adds the MQTT route to an MQTT Broker system with the IPAddress '1.2.3.4' and Port '42' on the local system.

EXAMPLE 2
> Add-MqttRoute -Address MqttSystem -port 42 -Destination CX_1234

Adds the MQTT route on the destination System 'CX_1234' to the MQTT Broker with Address '1.2.3.4' and Port '42'.

PARAMETERS

-Address
The Address of the MQTT Broker to add.
This can be the HostName or the IPAddress.

Type: String
Parameter Sets: (All)
Aliases:

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Port
The TCP/IP Port of the MQTT Broker to add.

Type: Int32
Parameter Sets: (All)
Aliases:

Required: True
Position: 1
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False

-Topic
The MQTT Topic string under which this MQTT Consumer sends/receives data.

Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: 2
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Destination
The Destination Address, where the MQTT route is added remotely.
**-Credential**

Credentials of the Preshared Key Identity.

**IMPORTANT:** Please be aware, that in the current version, the password is transferred as clear text through the network.

**Use this only in safe subnetworks.**

Type: PSCredential
Parameter Sets: Identity
Aliases: TargetCredential

**-IdentityCaseSensitive**

Key will be generated Sha256(Identity+Pwd), Identity in upper case if 'IdentityCaseSensitive' = false - UTF8

Type: SwitchParameter
Parameter Sets: Identity

**-Identity**

The identity name used to talk to the MQTT message broker (Preshared Key method).

Type: String
Parameter Sets: Psk

**-PreSharedKey**

The Preshared key used together with the identity for MQTT message broker communication.

Type: String
Parameter Sets: Psk
-CA
Path to the Certificate Authority file.
Certificates of MQTT broker, signed by this CA will be accepted for connection.
The file must be already located on the target system.
Type: String
Parameter Sets: SCA
Aliases:
Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Cert
Path to the public key Certificate (X.509).
The file must be already located on the target system.
Type: String
Parameter Sets: SCA
Aliases:
Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Key
Path of the private Key file of the X.509 Certificate.
The file must be already located on the target system.
Type: String
Parameter Sets: SCA
Aliases:
Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Quiet
The Quiet parameter suppresses the 'ShouldProcess' message and the routes will be removed without further question.
Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent
Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Confirm
Prompts you for confirmation before running the cmdlet.
-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi
Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

OUTPUTS

NOTES

6.4 Close-TcSession

SYNOPSIS

Closes the specified session object.

SYNTAX

Default (Default)

Close-TcSession -Id <Int32> [CommonParameters]

Session

Close-TcSession -InputObject <ISession> [CommonParameters]

DESCRIPTION

This Cmdlet closes the specified Point-To-Point Connection to the TwinCAT Target that is represented by the returned session object.

All registered SessionProvider types of Sessions can be used here (e.g. ADS, MQTT, OPC).
EXAMPLES

EXAMPLE 1
PS > Close-AdsSession $session

PARAMETERS

-Id
The session object to close is specified by this session ID.
Type: Int32
Parameter Sets: Default
Aliases:
Required: True
Position: Named
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False

-InputObject
The Session object to close.
Type: ISession
Parameter Sets: Session
Aliases: Session
Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

CommonParameters
This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession
The Session object to close.

OUTPUTS

NOTES

6.5 Copy-AdsFile

SYNOPSIS
Uploads / Downloads files from/to TwinCAT target.
SYNTAX

NetId (Default)
[-NetId <AmsNetId>] [<CommonParameters>]

Route
-[InputObject <IRoute>] [<CommonParameters>]

AddressStr
[-Address <String>] [<CommonParameters>]

SessionId
-[SessionId <Int32>] [<CommonParameters>]

DESCRIPTION

This Cmdlet implements ADS file transfer operations with TwinCAT Systems.

EXAMPLES

EXAMPLE 1
PS > Copy-AdsFile -address CX_00001 -path CurrentConfig.xml -Destination c:\tmp\Config1.xml -Directory BootDir

Downloads the the CurrentConfig.xml from the BootDir of the target system to 'c:\tmp\Config1.xml'

EXAMPLE 2
PS > Copy-AdsFile -address CX_00001 -upload -path c:\tmp\Config1.xml -Destination CurrentConfig.xml -Directory BootDir

Uploads the file "c:\tmp\Config1.xml" on local system to the Target BootFolder of system CX_00001

EXAMPLE 3
PS > Copy-AdsFile -address CX_0001 -path c:\ReadMe.txt -destination d:\tmp\ReadMe.txt

Downloads the File "C:\ReadMe.txt" form System CX_0001 to the local system and store it under d:\tmp\ReadMe.txt

PARAMETERS

-Path
The source path specifier, where the file is taken from.
If this Cmdlet is in Download mode, this is the specifier or FullPath of the (remote) file, dependant of the StandardFolder Parameter.
In case of 'Uploading' this is the FullPath of the file to be transferred.
-Destination

The Destination path/specifier, where the file is stored.

If the Cmdlet is in Download mode, this has to be the FullPath of the target location.

In case of 'Uploading' this can be the FileName or a FullPath dependent of the StandardDirectory Parameter.

-Directory

The Directory specifier on the remote system.

The Default is "Generic".

Possible values: Generic, BootDir, TargetDir, ConfigDir, InstallDir, RepositoryDir, UserPath1, UserPath2, UserPath3, UserPath4, UserPath5, UserPath6, UserPath7, UserPath8, UserPath9

-Upload

Switches the Cmdlet to Upload mode.

If not set, the Cmdlet is in 'Download' mode.

-Force

Forces to create the Directory on the target side (and overwrites any preexisting file).
### -NetId

The address (AmsNetId) of the system where the file is Downloaded from / Uploaded to (Default: Local)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: AmsNetId</td>
<td>Parameter Sets: NetId</td>
</tr>
<tr>
<td>Required: False</td>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: 172.17.60.167.1.1</td>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
<td></td>
</tr>
</tbody>
</table>

### -InputObject

The address (Route) of the system where the file is Downloaded from / Uploaded to (Default: Local)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: IRoute</td>
<td>Parameter Sets: Route</td>
</tr>
<tr>
<td>Required: True</td>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: None</td>
<td>Accept pipeline input: True (ByValue)</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
<td></td>
</tr>
</tbody>
</table>

### -Address

The address of the system where the file is Downloaded from / Uploaded to (Default: Local) This can be the RouteName, NetId, the HostName or the IPAddress.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: String</td>
<td>Parameter Sets: AddressStr</td>
</tr>
<tr>
<td>Required: True</td>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: None</td>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: True</td>
<td></td>
</tr>
</tbody>
</table>

### -SessionId

The target system address is derived from the Session Information where the file is Downloaded from / Uploaded to.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: Int32</td>
<td>Parameter Sets: SessionId</td>
</tr>
<tr>
<td>Required: True</td>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: -1</td>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
<td></td>
</tr>
</tbody>
</table>
CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute

The address (Route) of the system where the file is Downloaded from / Uploaded to (Default: Local)

OUTPUTS

NOTES

6.6 Get-AdsRoute

SYNOPSIS

List routes on a TwinCAT System / Broadcast search.

SYNTAX

GetRoutes (Default)


LocalSystem

Get-AdsRoute [-InputObject <String>] [-Local] [<CommonParameters>]

DESCRIPTION

This Cmdlet can list the routes configured on a TwinCAT local/remote system, or start determining all TwinCAT Systems within the current subnet.

EXAMPLES

EXAMPLE 1

PS> Get-AdsRoute
Name NetId Address Sub TcVersion RTSystem
---- ----- ------- ------- -------- ---------
CP-15ECA0 172.17.62.128.1.1 172.17.62.178 [UNKNOWN] [UNKNOWN]
CP-15ECA1 172.17.62.105.1.1 172.17.62.105 [UNKNOWN] [UNKNOWN]

Lists all registered local routes.

Because only the local port 10000 is addressed, the TcVersion and RTSystem is unknown (the Cmdlet doesn't contact the targets and doesn't produce additional roundtrips.

EXAMPLE 2

PS> get-AdsRoute -All
Name NetId Address Sub Version RTSystem
---- ----- ------- ------ -------- ---------
CX-1CEEDA 5.16.136.222.1.1 172.17.62.139 3.1.4020 Win7
Start a Broadcast search from the local system and lists the devices within the connected network.

EXAMPLE 3

```
PS> Get-AdsRoute -Name "Tc3*"
Name            NetId          Address       Sub Version RTSystem
----            -----          -------       ---       --------
TC3TESTA1-CP67X 172.17.62.105.1.1 172.17.62.105 0.0          Unknown
```

Get the (actual) route assigned to the local system that has the name pattern "Tc3***"

EXAMPLE 4

```
PS> Get-AdsRoute -All | where TcVersion -lt "3.1.0.0"
Name            NetId          Address       Sub Version RTSystem
----            -----          -------       ---       --------
TC3Test17-C6930 172.17.62.98.1.1 172.17.62.98 2.11.2234 Win7
CX2030-B401B    172.17.60.157.1.1 172.17.60.157 2.11.2256 Win7
CX-10A87B       5.16.168.123.1.1 172.17.62.140 2.11.2254 CE7.0
TC3Test13-C6650 172.17.60.239.1.1 172.17.62.156 2.11.2246 Win7
ECATTest01      172.17.61.6.1.1 172.17.61.31 2.11.2239 Win7
CX-12BCE5       172.17.60.165.1.1 172.17.62.191 2.11.2237 CE7.0
CX-1D82AA       172.17.62.180.1.1 172.17.62.180 3.1.4021 Win8
CX_0AB4F0       5.10.180.240.1.1 172.17.60.195 2.11.2243 XP
CP_1DFA0A       172.17.62.118.1.1 172.17.62.118 3.1.4021 Win7
CX-AF0001       172.17.62.75.1.1 172.17.62.70  3.1.4020 Win10
```

PARAMETERS

-All

Broadcast switch.

If activated a broadcast search is triggered within the local network.

The search can be constrained additionally by the -Address/-Name parameter.

Searching by Address (direct access of targets if no wildcards, otherwise using Broadcast search): - HostName: Searching the target by dns resolution and then via IP (fallback broadcast search filtering DeviceName/Hostname, not working over subnets!) - IPAddress: Directly accessing via IP (works also over subnets) - AmsNetId: Working via Broadcast search (not working over subnet segments!) Searching by Name: Works always via Broadcast search, wildcards permitted

Type: SwitchParameter
Parameter Sets: GetRoutes
Aliases: Broadcast

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Timeout

(Broadcast) Search Timeout (Default 5000 ms)
-**Address**

The Name / Address of the route to get.

The address of the route can be coded as NetId, the HostName or the IPAddress in string representation.

Wildcards are permitted.

-**InputObject**

The Destination address specifies the target, where the the routes are determined.

Use this to get the registered routes of a remote system.

The Destination system can be specified by RouteName (route name on local system), AmsNetId, IPAddress or HostName

-**StaticRoutes**

Indicates that only static routes will be returned.

By default this Cmdlet returns the actual registered routes.

-**Local**

If set, the local system route will be returned.

By default a list of the actual registered routes will be returned.
Type: SwitchParameter  
Parameter Sets: LocalSystem  
Aliases: Self  
Required: True  
Position: Named  
Default value: False  
Accept pipeline input: False  
Accept wildcard characters: False

**-Force**

If set, the broadcast search won't use cached routes.  
The Route will be determined by broadcast always.  
Only available with the -All parameter.

Type: SwitchParameter  
Parameter Sets: GetRoutes  
Aliases:  
Required: False  
Position: Named  
Default value: False  
Accept pipeline input: False  
Accept wildcard characters: False

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

**INPUTS**

**OUTPUTS**

**NOTES**

### 6.7 Get-AdsState

**SYNOPSIS**

Gets the Ads State of a TwinCAT Target.

**SYNTAX**

**NetIdPort (Default)**

[<CommonParameters>]

**AddressStr**

[<CommonParameters>]
Session
[<CommonParameters>]

SessionId
[<CommonParameters>]

Route
[<CommonParameters>]

DESCRIPTION
This command lets get the ADS state of a TwinCAT target.

EXAMPLES

EXAMPLE 1
PS > Get-AdsState 1.2.3.4.5.6

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
<th>OK</th>
<th>Time (ms)</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORK01</td>
<td>Config</td>
<td>True</td>
<td>0</td>
<td>1.2.3.4.5.6</td>
</tr>
</tbody>
</table>

EXAMPLE 2
PS > Get-AdsState

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
<th>OK</th>
<th>Time (ms)</th>
<th>Address</th>
</tr>
</thead>
</table>
| WORK01| Config| True| 0         | 1.2.3.4.5.6| ///

EXAMPLE 3
PS > Get-AdsState 1.2.3.4,CX_0130C7

EXAMPLE 4
PS > get-route | get-adsState

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
<th>OK</th>
<th>Time (ms)</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORK01</td>
<td>Config</td>
<td>True</td>
<td>0</td>
<td>1.2.3.4.5.6</td>
</tr>
</tbody>
</table>
| CX_0130C7  | Config  | True| 0         | 5.1.48.199.1.1| ///

EXAMPLE 5

EXAMPLE 6
PARAMETERS

-NetId

Enter the target system address(es).

Type: AmsNetId[]
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: 1
Default value: 172.17.60.167:1.1
Accept pipeline input: False
Accept wildcard characters: False

-Port

The AmsPort of the target system.

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: 2
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False

-Address

The address(es) where to get the State.

This can be the RouteName, NetId, the HostName or the IPAddress.
Wildcards are permitted.

Type: String[]
Parameter Sets: AddressStr
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-Session

The Session to use for the Cmdlet.

Type: ISession[]
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-SessionId

Specifies the Session (with unique ID) to use for the Cmdlet.

Type: Int32[]
Parameter Sets: SessionId
Aliases:

Required: True
### -InputObject

The target systems, where to get the AdsState from.

<table>
<thead>
<tr>
<th>Position: Named</th>
<th>Default value: None</th>
<th>Accept pipeline input: False</th>
<th>Accept wildcard characters: False</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: IRoute[]</td>
<td>Parameter Sets: Route</td>
<td>Aliases: Destination, Route</td>
<td></td>
</tr>
<tr>
<td>Required: True</td>
<td>Position: 1</td>
<td>Default value: None</td>
<td></td>
</tr>
<tr>
<td>Accept pipeline input: True (ByPropertyName, ByValue)</td>
<td>Accept wildcard characters: False</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### -Quiet

The quiet mode

<table>
<thead>
<tr>
<th>Position: Named</th>
<th>Default value: False</th>
<th>Accept pipeline input: False</th>
<th>Accept wildcard characters: False</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: SwitchParameter</td>
<td>Parameter Sets: (All)</td>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required: False</td>
<td>Position: Named</td>
<td>Default value: False</td>
<td></td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
<td>Accept wildcard characters: False</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### -StateOnly

The StateOnly mode

<table>
<thead>
<tr>
<th>Position: Named</th>
<th>Default value: False</th>
<th>Accept pipeline input: False</th>
<th>Accept wildcard characters: False</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: SwitchParameter</td>
<td>Parameter Sets: (All)</td>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required: False</td>
<td>Position: Named</td>
<td>Default value: False</td>
<td></td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
<td>Accept wildcard characters: False</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### -Force

Forced Mode

<table>
<thead>
<tr>
<th>Position: Named</th>
<th>Default value: False</th>
<th>Accept pipeline input: False</th>
<th>Accept wildcard characters: False</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: SwitchParameter</td>
<td>Parameter Sets: (All)</td>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required: False</td>
<td>Position: Named</td>
<td>Default value: False</td>
<td></td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
<td>Accept wildcard characters: False</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### -Timeout

Communication timeout in ms.

<table>
<thead>
<tr>
<th>Position: Named</th>
<th>Default value: False</th>
<th>Accept pipeline input: False</th>
<th>Accept wildcard characters: False</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: Int32</td>
<td>Parameter Sets: (All)</td>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required: False</td>
<td>Position: Named</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
<td>Accept wildcard characters: False</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession[]
The Session to use for the Cmdlet.

TwinCAT.IRoute[]
The target systems, where to get the AdsState from.

OUTPUTS

NOTES

6.8 Get-AmsRouterEndpoint

SYNOPSIS

Get the actual AmsConfiguration / RouterEndpoint of the process.

SYNTAX

Get-AmsRouterEndpoint [<CommonParameters>]

DESCRIPTION

This Cmdlet returns actual process-wide settings for the AmsConfiguration.

For more information please have a look at the SetAmsRouterEndpointCmdlet Cmdlet help.

EXAMPLES

EXAMPLE 1

PS> Get-AmsRouterEndpoint

<table>
<thead>
<tr>
<th>AddressFamily</th>
<th>Address</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>InterNetwork</td>
<td>127.0.0.1</td>
<td>48898</td>
</tr>
</tbody>
</table>

PARAMETERS

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.
Get-EcBoxes

SYNOPSIS

Gets the EtherCAT Boxes actually loaded on the target system.

SYNTAX

Get-EcBoxes [-InputObject] <EcMaster> [-Configured] [-Timeout <Int32>] [CommonParameters]

DESCRIPTION

This command lists the EtherCAT Boxes actually loaded on the target system.

EXAMPLES

EXAMPLE 1

PS> $m = Get-EcMaster -NetId 5.62.192.46.1.1
PS> $m | Get-EcBoxes

<table>
<thead>
<tr>
<th>Pos</th>
<th>Name</th>
<th>Type</th>
<th>Port</th>
<th>State</th>
<th>PortCrcError</th>
<th>FW</th>
<th>HW</th>
<th>ProductionDate</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Term 2 (EL1808)</td>
<td>EL1808-0000-0018</td>
<td>1001</td>
<td>PreOp A:0,B:0</td>
<td>7</td>
<td>0</td>
<td>2021-5-29</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Term 3 (EL2088)</td>
<td>EL2088-0000-0018</td>
<td>1002</td>
<td>PreOp A:0,B:0</td>
<td>9</td>
<td>0</td>
<td>2021-6-04</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Term 4 (EL2624)</td>
<td>EL2624-0000-0018</td>
<td>1003</td>
<td>PreOp A:0,B:0</td>
<td>12</td>
<td>1</td>
<td>2021-5-25</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Term 5 (EL3064)</td>
<td>EL3064-0000-0020</td>
<td>1004</td>
<td>PreOp A:0,B:0</td>
<td>15</td>
<td>9</td>
<td>2021-6-01</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Term 6 (EL4004)</td>
<td>EL4004-0000-0020</td>
<td>1005</td>
<td>PreOp A:0,B:0</td>
<td>19</td>
<td>4</td>
<td>2021-5-31</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Term 7 (EL6021)</td>
<td>EL6021-0000-0021</td>
<td>1006</td>
<td>PreOp A:0,B:0</td>
<td>13</td>
<td>9</td>
<td>2021-6-01</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Term 8 (EL9110)</td>
<td>EL9110-0000-0018</td>
<td>1007</td>
<td>PreOp A:0,B:0</td>
<td>14</td>
<td>0</td>
<td>2021-5-17</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Term 9 (EL1004)</td>
<td>EL1004-0000-0016</td>
<td>1008</td>
<td>PreOp A:0,B:0</td>
<td>0</td>
<td>0</td>
<td>2000-1-02</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Term 10 (EL2008)</td>
<td>EL2008-0000-0016</td>
<td>1009</td>
<td>PreOp A:0,B:0</td>
<td>0</td>
<td>0</td>
<td>2000-1-02</td>
<td></td>
</tr>
</tbody>
</table>

EXAMPLE 2

PS> $m = Get-EcMaster -NetId 5.62.192.46.1.1
PS> $m | Get-EcBoxes -configured

<table>
<thead>
<tr>
<th>Pos</th>
<th>Name</th>
<th>Type</th>
<th>Port</th>
<th>State</th>
<th>PortCrcError</th>
<th>FW</th>
<th>HW</th>
<th>ProductionDate</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Term 2 (EL1808)</td>
<td>EL1808-0000-0018</td>
<td>1001</td>
<td>PreOp A:0,B:0</td>
<td>7</td>
<td>0</td>
<td>2021-5-29</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Term 3 (EL2088)</td>
<td>EL2088-0000-0018</td>
<td>1002</td>
<td>PreOp A:0,B:0</td>
<td>9</td>
<td>0</td>
<td>2021-6-04</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Term 4 (EL2624)</td>
<td>EL2624-0000-0018</td>
<td>1003</td>
<td>PreOp A:0,B:0</td>
<td>12</td>
<td>1</td>
<td>2021-5-25</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Term 5 (EL3064)</td>
<td>EL3064-0000-0020</td>
<td>1004</td>
<td>PreOp A:0,B:0</td>
<td>15</td>
<td>9</td>
<td>2021-6-01</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Term 6 (EL4004)</td>
<td>EL4004-0000-0020</td>
<td>1005</td>
<td>PreOp A:0,B:0</td>
<td>19</td>
<td>4</td>
<td>2021-5-31</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Term 7 (EL6021)</td>
<td>EL6021-0000-0021</td>
<td>1006</td>
<td>PreOp A:0,B:0</td>
<td>13</td>
<td>9</td>
<td>2021-6-01</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Term 8 (EL9110)</td>
<td>EL9110-0000-0018</td>
<td>1007</td>
<td>PreOp A:0,B:0</td>
<td>14</td>
<td>0</td>
<td>2021-5-17</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Term 9 (EL1004)</td>
<td>EL1004-0000-0016</td>
<td>1008</td>
<td>PreOp A:0,B:0</td>
<td>0</td>
<td>0</td>
<td>2000-1-02</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Term 10 (EL2008)</td>
<td>EL2008-0000-0016</td>
<td>1009</td>
<td>PreOp A:0,B:0</td>
<td>0</td>
<td>0</td>
<td>2000-1-02</td>
<td></td>
</tr>
</tbody>
</table>

PARAMETERS

-InputObject

The EtherCAT master.

Type: EcMaster

Parameter Sets: (All)

Aliases:
**-Configured**

The List of Configured Boxes will be returned if specified.

If not specified, this Cmdlet will return the Online Boxes.

<table>
<thead>
<tr>
<th>Type: SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: (All)</td>
</tr>
<tr>
<td>Aliases:</td>
</tr>
</tbody>
</table>

| Required: False |
| Position: Named |
| Default value: False |
| Accept pipeline input: False |
| Accept wildcard characters: False |

**-Timeout**

Ads Communication timeout in ms (5000 by default)

| Type: Int32 |
| Parameter Sets: (All) |
| Aliases: |

| Required: False |
| Position: Named |
| Default value: 5000 |
| Accept pipeline input: False |
| Accept wildcard characters: False |

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

**INPUTS**

**EtherCAT.EcMaster**

The EtherCAT master.

**OUTPUTS**

**NOTES**

**6.10** Get-EcFrameStatistics

**SYNOPSIS**

Gets the EtherCAT Frame statistics from an ETHERCAT master.
SYNTAX

NetIdPortList (Default)
Get-EcFrameStatistics [-Timeout <Int32>] [-Count <Int32>] [-Delay <Int32>] [CommonParameters]

Default
[CommonParameters]

DESCRIPTION

Gets the EtherCAT Frame statistics from an ETherCAT master.

EXAMPLES

EXAMPLE 1
PS> $m = Get-EcMaster -NetId 5.62.192.46.1.1
PS> $m | Get-EcFrameStatistics -count 5 -delay 0

PARAMETERS

-InputObject

The EtherCAT master.

Type: EcMaster
Parameter Sets: Default
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-Timeout

ADS Communication timeout

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 5000
Accept pipeline input: False
Accept wildcard characters: False

-Count

Specifies the number of statistic requests (Default is 1)

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 1
Accept pipeline input: False
Accept wildcard characters: False
-Delay
Delay in Seconds between requests in Seconds (Default is 1s)

<table>
<thead>
<tr>
<th>Type</th>
<th>Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required:</td>
<td>False</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>1</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

EtherCAT.EcMaster
The EtherCAT master.

OUTPUTS

NOTES

6.11 Get-EcMaster

SYNOPSIS

Gets the Ads State of a TwinCAT Target.

SYNTAX

NetIdPortList (Default)
Get-EcMaster [-Id <Int32>] [-Timeout <Int32>] [<CommonParameters>]

NetIdPort
Get-EcMaster [-NetId <AmsNetId>] [-Id <Int32>] [-Timeout <Int32>] [<CommonParameters>]

AddressStr
Get-EcMaster [-Address <String>] [-Id <Int32>] [-Timeout <Int32>] [<CommonParameters>]

Session
Get-EcMaster -Session <ISession> [-Id <Int32>] [-Timeout <Int32>] [<CommonParameters>]

DESCRIPTION

This command let gets the ADS state of a TwinCAT target.
EXAMPLES

EXAMPLE 1
PS> Get-IODevice -NetId 5.62.192.46.1.1

<table>
<thead>
<tr>
<th>ID</th>
<th>DeviceName</th>
<th>DeviceType</th>
<th>DeviceNetId</th>
<th>BoxesCount</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Device 1 (EtherCAT)</td>
<td>EtherCAT_DirectModeV210</td>
<td>172.16.1.3.1.1</td>
<td>4</td>
</tr>
</tbody>
</table>

EXAMPLE 2
PS> (Get-IODevice -Address CX_01234 -Id 1).Boxes

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>BoxType</th>
<th>Port</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Box 1 (IFC2422)</td>
<td>EtherCAT_EXXXXX</td>
<td>1001</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Box 2 (IFC2421m)</td>
<td>EtherCAT_EXXXXX</td>
<td>1002</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Box 3 (IFC2421m1)</td>
<td>EtherCAT_EXXXXX</td>
<td>1003</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Box 4 (IFC2421m2)</td>
<td>EtherCAT_EXXXXX</td>
<td>1004</td>
<td></td>
</tr>
</tbody>
</table>

PARAMETERS

-NetId
Enter the target system address(es).

Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:
Required: False
Position: Named
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

-Address
The address(es) where to get Devices.
This can be the RouteName, NetId, the HostName or the IPAddress.
Wildcards are permitted.

Type: String
Parameter Sets: AddressStr
Aliases:
Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-Session
The Session to use for the Cmdlet, must be connected to port 300, R0_IO

Type: ISession
Parameter Sets: Session
Aliases:
Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
-Id
Specifies the DeviceId to scan by the Cmdlet.
If not specified, all Devices will be scanned.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Int32</td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
</tr>
<tr>
<td>Position</td>
<td>Named</td>
</tr>
<tr>
<td>Default</td>
<td>-1</td>
</tr>
<tr>
<td>Accept</td>
<td>pipeline input: False</td>
</tr>
<tr>
<td>Accept</td>
<td>wildcard characters: False</td>
</tr>
</tbody>
</table>

-Timeout
Ads Communication timeout in ms.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Int32</td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
</tr>
<tr>
<td>Position</td>
<td>Named</td>
</tr>
<tr>
<td>Default</td>
<td>-1</td>
</tr>
<tr>
<td>Accept</td>
<td>pipeline input: False</td>
</tr>
<tr>
<td>Accept</td>
<td>wildcard characters: False</td>
</tr>
</tbody>
</table>

CommonParameters
This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS
TwinCAT.ISession
The Session to use for the Cmdlet, must be connected to port 300, R0_IO

OUTPUTS

NOTES

6.12 Get-IODevice

SYNOPSIS
Gets actually loaded IO Devices of the target system.

SYNTAX

NetIdPortList (Default)
Get-IODevice [-Id <Int32>] [-Timeout <Int32>] [<CommonParameters>]

NetIdPort
Get-IODevice [-NetId <AmsNetId>] [-Id <Int32>] [-Timeout <Int32>] [<CommonParameters>]
AddressStr

Get-IODevice [~Address] <String> [-Id <Int32>] [-Timeout <Int32>] [CommonParameters]

Session

Get-IODevice -Session <ISession> [-Id <Int32>] [-Timeout <Int32>] [CommonParameters]

DESCRIPTION

This command lists the actually loaded IO Devices of the target system.

The list can be filtered by specific Device IDs.

EXAMPLES

EXAMPLE 1

PS> Get-IODevice -NetId 5.62.192.46.1.1

<table>
<thead>
<tr>
<th>ID</th>
<th>DeviceName</th>
<th>DeviceType</th>
<th>DeviceNetId</th>
<th>BoxesCount</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Device 1</td>
<td>(EtherCAT)</td>
<td>172.16.1.3.2.1</td>
<td>4</td>
</tr>
</tbody>
</table>

EXAMPLE 2

PS> (Get-IODevice -Address CX_01234 -Id 1).Boxes

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>BoxType</th>
<th>Port</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Box 1</td>
<td>IFC2422</td>
<td>1001</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Box 2</td>
<td>IFC2421m</td>
<td>1002</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Box 3</td>
<td>IFC2421m1</td>
<td>1003</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Box 4</td>
<td>IFC2421m2</td>
<td>1004</td>
<td></td>
</tr>
</tbody>
</table>

PARAMETERS

-**-NetId**

Enter the target system address(es).

Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: Named
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

-**-Address**

The address(es) where to get Devices.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

Type: String
Parameter Sets: AddressStr
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

**-Session**

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

Type: ISession
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

**-Id**

Specifies the DeviceId to scan by the Cmdlet.

If not specified, all Devices will be scanned.

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False

**-Timeout**

ADS Communication timeout in ms.

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

**INPUTS**

**TwinCAT.ISession**

The Session to use for the Cmdlet, must be connected to port 300, R0_IO
6.13 Get-IoFreeRun

SYNOPSIS
Gets the IO FreeRun State of the specified target.

SYNTAX
NetIdPortList (Default)
Get-IoFreeRun [-Timeout <Int32>] [<CommonParameters>]

NetIdPort
Get-IoFreeRun [-NetId <AmsNetId>] [-Timeout <Int32>] [<CommonParameters>]

AddressStr
Get-IoFreeRun [-Address] <String> [-Timeout <Int32>] [<CommonParameters>]

Session
Get-IoFreeRun -Session <ISession> [-Timeout <Int32>] [<CommonParameters>]

DESCRIPTION
This command gets the IO FreeRun state of specified target when the target is in config mode.
If the target system is not in config mode, a warning is produced.

EXAMPLES

EXAMPLE 1
PS> Get-IoFreeRun -NetId 5.62.192.46.1.1
$true

PARAMETERS

-NetId
Enter the target system address(es).
Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:
Required: False
Position: Named
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False
- **Address**

The address(es) where to get Devices.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

Type: String
Parameter Sets: AddressStr
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

- **Session**

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

Type: ISession
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

- **Timeout**

ADS Communication timeout in ms.

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

**INPUTS**

**TwinCAT.ISession**

The Session to use for the Cmdlet, must be connected to port 300, R0_IO
6.14 Get-MqttRoute

SYNOPSIS
Remove a MQTT Route.

SYNTAX
Get-MqttRoute [-Destination <String>] [CommonParameters]

DESCRIPTION
Removes a MQTT Route of the specified system.

EXAMPLES
EXAMPLE 1
PS> Get-MqttRoute -destination CX_1234
Address   TcpPort Topic  Qos Security
-------   ------- ----- --- --------
192.168.2.1 44124 Topic1   TLS
192.200.2.2 44124 Topic2   PSK
192.200.3.3 44124 Topic3   None

Gets the MQTT Routes registered on the destination System 'CX_1234'.

PARAMETERS
-Destination
The destination address, where to Remove the specified Mqtt route.

This can be the NetId, the HostName or the IPAddress

Type: String
Parameter Sets: (All)
Aliases:
Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

CommonParameters
This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.
6.15 Get-RTimeCpuSettings

SYNOPSIS
Getting the Cpu Settings of the TwinCAT System

SYNTAX

NetIdPort (Default)
Get-RTimeCpuSettings [-NetId <AmsNetId>] [-Timeout <Int32>] [CommonParameters]

AddressStr
Get-RTimeCpuSettings [-Address] <String> [-Timeout <Int32>] [CommonParameters]

Session
Get-RTimeCpuSettings -Session <ISession> [-Timeout <Int32>] [CommonParameters]

DESCRIPTION
This command lists the actually configured Relatime, Windows and Realtime cores of the TwinCAT System.

EXAMPLES

EXAMPLE 1
PS> Get-RTimeCpuSettings

<table>
<thead>
<tr>
<th>NetId</th>
<th>Windows Cores</th>
<th>NonWin Cores</th>
<th>RealTime Cores</th>
<th>Cpu Type</th>
<th>Cpu Family</th>
<th>CpuFrequency (GHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.17.60.167.1.1</td>
<td>22</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>3793</td>
</tr>
</tbody>
</table>

PARAMETERS

-NetId
The AmsNetId of the Target system.

Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: Named
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

-Address
The target address(es) where to get the CPU Settings.
This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

Type: String
Parameter Sets: AddressStr
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-Session

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

Type: ISession
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-Timeout

Ads Timeout in Seconds for the ADS command roundtrip.

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

OUTPUTS

NOTES

6.16 Get-RTimeLatency

SYNOPSIS

Get the latency of TwinCAT Realtime Cores of the specified TwinCAT target system.
SYNTAX

NetIdPort (Default)

AddressStr

Session
Get-RTimeLatency -Session <ISession> [-Core <Int32>] [-TimeoutSeconds <Int32>] [-Count <Int32>] [-Delay <Int32>] [<CommonParameters>]

DESCRIPTION

This command lists the Realtime Cores of specified TwinCAT target systems.

The values can be repeated by count parameter and a repeat delay can be set.

EXAMPLES

EXAMPLE 1
PS> Get-RTimeLatency

<table>
<thead>
<tr>
<th>NetId</th>
<th>CoreId</th>
<th>Latency (us)</th>
<th>MaxLatency (us)</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.91.172.198.1.1</td>
<td>1</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

EXAMPLE 2
PS> Get-RTimeLatency -NetId 5.91.172.198.1.1 -core 1 -count 5 -Delay 0

<table>
<thead>
<tr>
<th>NetId</th>
<th>CoreId</th>
<th>Latency (us)</th>
<th>MaxLatency (us)</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.91.172.198.1.1</td>
<td>1</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>5.91.172.198.1.1</td>
<td>1</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>5.91.172.198.1.1</td>
<td>1</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>5.91.172.198.1.1</td>
<td>1</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>5.91.172.198.1.1</td>
<td>1</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

PARAMETERS

-NetId

The AmsNetId of the target system.

The Default is the local system (if left out).

Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: Named
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False
**-Address**

The address(es) where to get the Realtime latency.

This can be the RouteName, NetId, the HostName or the IPAddress.

Multiple Addresses and Wildcards are permitted.

Type: String  
Parameter Sets: AddressStr  
Aliases: 
  Required: True  
  Position: 1  
  Default value: None  
  Accept pipeline input: False  
  Accept wildcard characters: True

**-Session**

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

Type: ISession  
Parameter Sets: Session  
Aliases: 
  Required: True  
  Position: Named  
  Default value: None  
  Accept pipeline input: True (ByValue)  
  Accept wildcard characters: False

**-Core**

Specifies the CoreID of the Realtime Core.

If not specified, this Cmdlet returns all Realtime Cores.

Type: Int32  
Parameter Sets: (All)  
Aliases: 
  Required: False  
  Position: Named  
  Default value: -1  
  Accept pipeline input: False  
  Accept wildcard characters: False

**-TimeoutSeconds**

Scanning timeout in ms.

This is the timeout for each single ADS roundtrip used by this Cmdlet.

Type: Int32  
Parameter Sets: (All)  
Aliases: 
  Required: False  
  Position: Named  
  Default value: -1  
  Accept pipeline input: False  
  Accept wildcard characters: False

**-Count**

Specifies the number request repeatations (Default is 1).

Type: Int32  
Parameter Sets: (All)  
Aliases: 
-Delay

Delay in Seconds between requests in Seconds (Default is 1s)

<table>
<thead>
<tr>
<th>Type: Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: (All)</td>
</tr>
</tbody>
</table>

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see aboutCommonParameters.

INPUTS

TwinCAT.ISession

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

OUTPUTS

NOTES

6.17 Get-RTimePerformance

SYNOPSIS

Gets the Realtime Performance of the specified system.

SYNTAX

**NetIdPort (Default)**


**AddressStr**

Session

DESCRIPTION
Gets the Realtime Performance of the specified TwinCAT Target systems.
The output can be filtered for specified Realtime CPUs and contains the actual CPU Latency and CPU Load.
This Cmdlet is preliminary and subject of change.
Performance Data is not supported before TwinCAT Build 4025.5.

EXAMPLES
EXAMPLE 1
PS> Get-RTimePerformance

PARAMETERS

-NetId
The AmsNetId of the target system.
Uses the Local system if empty.
Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:
Required: False
Position: Named
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

-Address
The address(es) where to get the performance data.
This can be the RouteName, NetId, the HostName or the IPAddress.
Multiple addresses and wildcards are permitted.
Type: String
Parameter Sets: AddressStr
Aliases:
Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-Session
The Session to use for the Cmdlet, must be connected to port 300, R0_IO
Type: ISession
Parameter Sets: Session
Aliases:
Required: True
Position: Named
**-Core**

Specifies the ID of the Core where to determine the performance data.

If not specified, all Realtime CPUs will be returned.

- **Type**: Int32
- **Parameter Sets**: (All)
- **Aliases**: 
- **Required**: False
- **Position**: Named
- **Default value**: -1
- **Accept pipeline input**: False
- **Accept wildcard characters**: False

**-Timeout**

Ads Timeout in Seconds for a single Performance request.

- **Type**: Int32
- **Parameter Sets**: (All)
- **Aliases**: 
- **Required**: False
- **Position**: Named
- **Default value**: -1
- **Accept pipeline input**: False
- **Accept wildcard characters**: False

**-Count**

Specifies the number of performance requests (Default is 1) per target and CPU.

- **Type**: Int32
- **Parameter Sets**: (All)
- **Aliases**: 
- **Required**: False
- **Position**: Named
- **Default value**: 1
- **Accept pipeline input**: False
- **Accept wildcard characters**: False

**-Delay**

Delay in Seconds between performance requests in Seconds (Default is 1s)

- **Type**: Int32
- **Parameter Sets**: (All)
- **Aliases**: 
- **Required**: False
- **Position**: Named
- **Default value**: 1
- **Accept pipeline input**: False
- **Accept wildcard characters**: False

**-Reset**

Reset the max delay of the PerformanceData before getting new data.

- **Type**: SwitchParameter
- **Parameter Sets**: (All)
- **Aliases**: 
Get-TcDataType

SYNOPSIS
Get the DataTypes from a TwinCAT target system / Device.

SYNTAX
NetIdPort (Default)
Get-TcDataType [[-Name] <String[]>] [-NetId <AmsNetId>] -Port <Int32> [<CommonParameters>]

Route
Get-TcDataType [[-Name] <String[]>] -Route <IRoute> -Port <Int32> [<CommonParameters>]

AddressStr
Get-TcDataType [[-Name] <String[]>] -Address <String> -Port <Int32> [<CommonParameters>]

Session
Get-TcDataType [[-Name] <String[]>] -InputObject <ISession> [<CommonParameters>]

SessionId
Get-TcDataType [[-Name] <String[]>] -SessionId <Int32> [<CommonParameters>]

DESCRIPTION
This Cmdlet get the DataTypes from a target system if symbolic information is provided by the device (Symbol Server running).

The DataTypes can be determined via different Providers (e.g. ADS, MQTT, OPC, see the 'Provider' parameter.)
EXAMPLES

EXAMPLE 1
PS> Get-TcDataType -port 851
Name          Size  Category     BaseType
----          ----  --------     --------
BYTE          1     Primitive    
WORD          2     Primitive    
DINT          4     Primitive    
UDINT         4     Primitive    
DWORD         4     Primitive    
E_ByteEnum    1     Enum        BYTE
FB_Test       12424 Struct
PLC.PlcAppSystemInfo 256 Struct
PLC.PlcTaskSystemInfo 128 Struct
POINTER TO BYTE 4     Pointer     BYTE
R_Range       2     Alias       INT (-6..12)
REFERENCE TO BOOL 4     Reference    BOOL
ST_SimpleStruct 166 Struct
STRING(80)    81     String
...

Get the data types from the local system (Port 851):

EXAMPLE 2
PS> $types = Get-TcDataType -Name 'ST_*' -NetId 1.2.3.4.5.6 -Port 851

Gets the DataTypes with name pattern 'ST_*' from the NetId / Port address symbol server.

EXAMPLE 3
PS> $session = New-TcSession -Name 'CX_123456' -port 851
PS> Get-TcDataType -Session $session | where ByteSize -gt 1KB

Gets an ADS-Session/Connection to the target system CX_123456 on port 851, downloads the datatype information and returns all the DataTypes that are larger than 1KB of Size.

PARAMETERS

-Name
The data type name(s) to get.

Wildcards are permitted.

Type: String[
Parameter Sets: (All)
Aliases:

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-NetId
The NetID address of the target system where to load the datatypes (Local by default).

Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: Named
Default value: 172.17.60.167.1.1
Accept pipeline input: False  
Accept wildcard characters: False

- **Route**

The Route object where to load the datatypes from (RouteTarget.Local by default).

<table>
<thead>
<tr>
<th>Type:</th>
<th>IRoute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>Route</td>
</tr>
<tr>
<td>Aliases:</td>
<td>Destination</td>
</tr>
<tr>
<td>Required:</td>
<td>True</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

- **Address**

The address where to load the datatype descriptions.

This can be the RouteName, NetId, the HostName or the IPAddress. Wildcards are permitted.

<table>
<thead>
<tr>
<th>Type:</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>AddressStr</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required:</td>
<td>True</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>True</td>
</tr>
</tbody>
</table>

- **Port**

The Port where to load the datatype descriptions.

<table>
<thead>
<tr>
<th>Type:</th>
<th>Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>NetIdPort, Route, AddressStr</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required:</td>
<td>True</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>10000</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

- **InputObject**

The session object to use for datatype upload.

<table>
<thead>
<tr>
<th>Type:</th>
<th>ISession</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>Session</td>
</tr>
<tr>
<td>Aliases:</td>
<td>Session</td>
</tr>
<tr>
<td>Required:</td>
<td>True</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>True (ByValue)</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

- **SessionId**

The unique session Identifier that represents the session to use for the datatype upload.

<table>
<thead>
<tr>
<th>Type:</th>
<th>Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>SessionId</td>
</tr>
<tr>
<td>Aliases:</td>
<td>Id</td>
</tr>
<tr>
<td>Required:</td>
<td>True</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>
### CommonParameters

This cmdlet supports the common parameters: `-Debug`, `-ErrorAction`, `-ErrorVariable`, `-InformationAction`, `-InformationVariable`, `-OutVariable`, `-OutBuffer`, `-PipelineVariable`, `-Verbose`, `-WarningAction`, and `-WarningVariable`. For more information, see `about_CommonParameters`.

### INPUTS

**TwinCAT.ISession**

The session object to use for datatype upload.

### OUTPUTS

### NOTES

#### 6.19 Get-TcEvent

**SYNOPSIS**

Gets TwinCAT events from event logs on local and remote computers.

**SYNTAX**

```
```

**DESCRIPTION**

TwinCAT creates LogEntries in the Application log.

These most important entries for system diagnostics source from the TcSysUI Application and the TwinCAT System Service (TcSysSrv) containing the logentries from the TwinCAT Drivers.

This Get-TcEvent Cmdlet gets these events from the local system and as well from other reachable systems within the corporate network.

If you're not running PowerShell as an Administrator, you might see error messages that you cannot retrieve information about a log.

**EXAMPLES**

**EXAMPLE 1**

```
PS> get-tcevent -MaxEvents 30
  ProviderName: TcSysUi

  TimeCreated       Id   LevelDisplayName             Message
  --------------    --   -----------------------------          -------
  17.08.2021 15:56:44 3   Information                Process startup apps was already triggered.
  17.08.2021 15:56:44 1   Information                Process startup apps after reaching RUN state.

  ProviderName: TcSysSrv
```
<table>
<thead>
<tr>
<th>TimeCreated</th>
<th>Id</th>
<th>Level</th>
<th>DisplayName</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.08.2021 15:56:44</td>
<td>66</td>
<td>Info</td>
<td>TcEventLogger</td>
<td>Starting COM Server TcEventLogger !</td>
</tr>
<tr>
<td>17.08.2021 15:56:44</td>
<td>20000</td>
<td>Info</td>
<td>TcEventLogger</td>
<td>TwinCAT System Message: Source: License Server; Timestamp: 8/17/2021 3:56:44 PM 345 ms Message: license validation status is Valid(3)</td>
</tr>
<tr>
<td>17.08.2021 15:56:44</td>
<td>15</td>
<td>Info</td>
<td>TcRTime</td>
<td>Server started: TcRTime.</td>
</tr>
<tr>
<td>17.08.2021 15:56:44</td>
<td>15</td>
<td>Info</td>
<td>TcIo</td>
<td>Server started: TcIo.</td>
</tr>
<tr>
<td>17.08.2021 15:56:44</td>
<td>69</td>
<td>Info</td>
<td>TcEventLogger</td>
<td>Initializing COM Server TcEventLogger !</td>
</tr>
<tr>
<td>17.08.2021 15:56:44</td>
<td>71</td>
<td>Info</td>
<td>TcEventLogger</td>
<td>Loading configuration of COM server TcEventLogger !</td>
</tr>
</tbody>
</table>

**ProviderName: TcSysUi**

<table>
<thead>
<tr>
<th>TimeCreated</th>
<th>Id</th>
<th>Level</th>
<th>DisplayName</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.08.2021 15:56:44</td>
<td>2</td>
<td>Info</td>
<td></td>
<td>Process startup apps skipped after reaching state '4'</td>
</tr>
<tr>
<td>17.08.2021 15:56:44</td>
<td>2</td>
<td>Info</td>
<td></td>
<td>Process startup apps skipped after reaching state '6'</td>
</tr>
</tbody>
</table>

**ProviderName: TcSysSrv**

<table>
<thead>
<tr>
<th>TimeCreated</th>
<th>Id</th>
<th>Level</th>
<th>DisplayName</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.08.2021 15:56:43</td>
<td>67</td>
<td>Info</td>
<td>TcEventLogger</td>
<td>Stopping COM Server TcEventLogger !</td>
</tr>
<tr>
<td>17.08.2021 15:56:43</td>
<td>28</td>
<td>Info</td>
<td>TCIDRIVERS</td>
<td>Server stopped.</td>
</tr>
<tr>
<td>17.08.2021 15:56:43</td>
<td>28</td>
<td>Info</td>
<td>TCIDECAT</td>
<td>Server stopped.</td>
</tr>
<tr>
<td>17.08.2021 15:56:43</td>
<td>28</td>
<td>Info</td>
<td>TCIOETH</td>
<td>Server stopped.</td>
</tr>
<tr>
<td>17.08.2021 15:56:43</td>
<td>28</td>
<td>Info</td>
<td>TCRTSOBJECTS</td>
<td>Server stopped.</td>
</tr>
<tr>
<td>17.08.2021 15:56:43</td>
<td>28</td>
<td>Info</td>
<td>TCRTIME</td>
<td>Server stopped.</td>
</tr>
<tr>
<td>17.08.2021 15:56:43</td>
<td>28</td>
<td>Info</td>
<td>TCIO</td>
<td>Server stopped.</td>
</tr>
<tr>
<td>17.08.2021 15:56:43</td>
<td>68</td>
<td>Info</td>
<td>TcEventLogger</td>
<td>Shutting down COM Server TcEventLogger !</td>
</tr>
<tr>
<td>17.08.2021 15:56:42</td>
<td>70</td>
<td>Info</td>
<td>TcEventLogger</td>
<td>Saving configuration of COM server TcEventLogger !</td>
</tr>
</tbody>
</table>

**ProviderName: TcSysUi**

<table>
<thead>
<tr>
<th>TimeCreated</th>
<th>Id</th>
<th>Level</th>
<th>DisplayName</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.08.2021 15:56:42</td>
<td>2</td>
<td>Info</td>
<td></td>
<td>Process startup apps skipped after reaching state '17'</td>
</tr>
</tbody>
</table>

**ProviderName: TcSysSrv**

<table>
<thead>
<tr>
<th>TimeCreated</th>
<th>Id</th>
<th>Level</th>
<th>DisplayName</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.08.2021 15:56:42</td>
<td>33</td>
<td>Info</td>
<td></td>
<td>TwinCAT System Restart initiated from AmsNetId: 172.1 7.60.197.1.1 port 32894.</td>
</tr>
</tbody>
</table>

**ProviderName: TcSysUi**

<table>
<thead>
<tr>
<th>TimeCreated</th>
<th>Id</th>
<th>Level</th>
<th>DisplayName</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.08.2021 15:50:56</td>
<td>2</td>
<td>Info</td>
<td></td>
<td>Process startup apps skipped after reaching state '15'</td>
</tr>
</tbody>
</table>

**ProviderName: TcSysSrv**

<table>
<thead>
<tr>
<th>TimeCreated</th>
<th>Id</th>
<th>Level</th>
<th>DisplayName</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.08.2021 15:50:56</td>
<td>66</td>
<td>Info</td>
<td>TcEventLogger</td>
<td>Starting COM Server TcEventLogger !</td>
</tr>
<tr>
<td>17.08.2021 15:50:55</td>
<td>15</td>
<td>Info</td>
<td>TCIDRIVERS</td>
<td>Server started: TCIDRIVERS.</td>
</tr>
</tbody>
</table>

Gets last 30 events (all events) on the local system.

**EXAMPLE 2**

PS> get-tcevent -computerName CX_1234 -Level Critical, Error, Warning -source TcSysSrv -StartTime([DateTime]::Now -[TimeSpan]::FromDays(1))
Getting the logged errors/warnings of system 'CX_1234' of the last 24 hours and filter the events for the Event provider 'TcSysSrv' (the TwinCAT System Service)

EXAMPLE 3

PS> get-tcevent --StartTime '2021-08-17 15:50:55' -EndTime '2021-08-17 15:55:56' -source TcSysSrv

ProviderName: TcSysSrv

<table>
<thead>
<tr>
<th>TimeCreated</th>
<th>Id</th>
<th>Level</th>
<th>DisplayName</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.08.2021 15:50:56</td>
<td>66</td>
<td>Information</td>
<td>Starting COM Server TcEventLogger !</td>
<td></td>
</tr>
<tr>
<td>17.08.2021 15:50:55</td>
<td>15</td>
<td>Information</td>
<td>TCIODRIVERS Server started: TCIODRIVERS.</td>
<td></td>
</tr>
<tr>
<td>17.08.2021 15:50:55</td>
<td>15</td>
<td>Information</td>
<td>TCIOECAT Server started: TCIOECAT.</td>
<td></td>
</tr>
<tr>
<td>17.08.2021 15:50:55</td>
<td>15</td>
<td>Information</td>
<td>TCIOETH Server started: TCIOETH.</td>
<td></td>
</tr>
<tr>
<td>17.08.2021 15:50:55</td>
<td>15</td>
<td>Information</td>
<td>TCRTSOBJECTS Server started: TCRTSOBJECTS.</td>
<td></td>
</tr>
<tr>
<td>17.08.2021 15:50:55</td>
<td>15</td>
<td>Information</td>
<td>TCRTIME Server started: TCRTIME.</td>
<td></td>
</tr>
<tr>
<td>17.08.2021 15:50:55</td>
<td>15</td>
<td>Information</td>
<td>TCIO Server started: TCIO.</td>
<td></td>
</tr>
<tr>
<td>17.08.2021 15:50:55</td>
<td>69</td>
<td>Information</td>
<td>Initializing COM Server TcEventLogger !</td>
<td></td>
</tr>
<tr>
<td>17.08.2021 15:50:55</td>
<td>71</td>
<td>Information</td>
<td>Loading configuration of COM server TcEventLogger !</td>
<td></td>
</tr>
<tr>
<td>17.08.2021 15:50:55</td>
<td>67</td>
<td>Information</td>
<td>Stopping COM Server TcEventLogger !</td>
<td></td>
</tr>
</tbody>
</table>

Show EventLog of TwinCAT SystemService between two points in time.

PARAMETERS

- **-MaxEvents**

  Specifies the maximum number of events that are returned.

  Enter an integer such as 100.

  The default is to return all TwinCAT events.

  Type: Int32
  Parameter Sets: (All)
  Aliases:

  Required: False
  Position: Named
  Default value: None
  Accept pipeline input: False
  Accept wildcard characters: False

- **-ComputerName**

  Specifies the name of the computer that this cmdlet gets events from the event logs.

  Type the NetBIOS name, an IP address, or the fully qualified domain name (FQDN) of the computer.

  The default value is the local computer, localhost.

  Type: String
  Parameter Sets: (All)
  Aliases:

  Required: False
  Position: Named
  Default value: None
  Accept pipeline input: False
  Accept wildcard characters: False
-**Credential**

Specifies a user account that has permission to perform this action.

The default value is the current user.

Type a user name, such as User01 or Domain01\User01. Or, enter a PSCredential object, such as one generated by the Get-Credential cmdlet.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter Sets</th>
<th>Aliases</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSCredential</td>
<td>(All)</td>
<td></td>
</tr>
</tbody>
</table>

Required: False  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

-**Level**

The log level, that is determined.

By default, all EventLog content will be returned.

Allowed values are: - All - Critical - Error - Warning - Informational - Verbose

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter Sets</th>
<th>Aliases</th>
</tr>
</thead>
<tbody>
<tr>
<td>String[]</td>
<td>(All)</td>
<td></td>
</tr>
</tbody>
</table>

Required: False  
Position: Named  
Default value: All  
Accept pipeline input: False  
Accept wildcard characters: False

-**Source**

The log level, that is determined.

By default, all TwinCAT EventLog content will be returned.

Actually allowed values are: - All - TcSysUI - TcSysSrv - TcEventLogger - Tc3ScopeServer - 'TF3300 TwinCAT 3 Scope Server' - 'TwinCAT3 Scope Server'

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter Sets</th>
<th>Aliases</th>
</tr>
</thead>
<tbody>
<tr>
<td>String[]</td>
<td>(All)</td>
<td></td>
</tr>
</tbody>
</table>

Required: False  
Position: Named  
Default value: All  
Accept pipeline input: False  
Accept wildcard characters: False

-**StartTime**

Filters the EventLog to return only entries that were produced after the specified start time.

By default the start time is not set, returning Events from the beginning of the log.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter Sets</th>
<th>Aliases</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTimeOffset</td>
<td>(All)</td>
<td></td>
</tr>
</tbody>
</table>

Required: False  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False
**-EndTime**

Filters the EventLog to return only entries that were produced up to the specified end time.

By default the end time is not set, returning the newest entries.

<table>
<thead>
<tr>
<th>Type: DateTimeOffset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: (All)</td>
</tr>
<tr>
<td>Aliases:</td>
</tr>
<tr>
<td>Required: False</td>
</tr>
<tr>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: None</td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
</tr>
</tbody>
</table>

**-ID**

Filters the EventLog to the specified EventIds.

<table>
<thead>
<tr>
<th>Type: Int32[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: (All)</td>
</tr>
<tr>
<td>Aliases:</td>
</tr>
<tr>
<td>Required: False</td>
</tr>
<tr>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: None</td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
</tr>
</tbody>
</table>

**-Timeout**

Timeout of the Read event log operation (default: no timeout).

<table>
<thead>
<tr>
<th>Type: Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: (All)</td>
</tr>
<tr>
<td>Aliases:</td>
</tr>
<tr>
<td>Required: False</td>
</tr>
<tr>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: None</td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
</tr>
</tbody>
</table>

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbosity, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

**INPUTS**

**OUTPUTS**

**NOTES**

**6.20 Get-TcLicense**

**SYNOPSIS**

Get TwinCAT License information.
SYNTAX

NetIdPort (Default)
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] [-NetId <AmsNetId>] [-Status <LicenseStatus>] [<CommonParameters>]

Route
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] -Route <IRoute> [-Status <LicenseStatus>] [<CommonParameters>]

AddressStr
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] -Address <String> [-Status <LicenseStatus>] [<CommonParameters>]

Session
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] -InputObject <ISession> [-Status <LicenseStatus>] [<CommonParameters>]

SessionId
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] -SessionId <Int32> [-Status <LicenseStatus>] [<CommonParameters>]

DESCRIPTION

This Cmdlet gets information about TwinCAT licenses from the target system.

To contact the target system, it must be available as actual route or the local system.

EXAMPLES

EXAMPLE 1
PS> Get-TcLicense

Get the the valid licenses from the local system.

EXAMPLE 2
PS> $session = New-TcSession -Route TC3TESTA1-CP67X -Port 30
PS> $session | Get-TcLicense -Status All -name *scope*

<table>
<thead>
<tr>
<th>Name</th>
<th>Valid</th>
<th>ValidityCode</th>
<th>ExpireTime</th>
<th>Available</th>
<th>Used</th>
<th>VolumeNo</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC3 Scope Server</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TC3 Scope View Professional</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>00</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Create a session to the License Server on target 'TC3TESTA1-CP67X' and return all valid and invalid licenses that contain 'scope' in their name.

EXAMPLE 3
PS> Get-TcLicense -Route TC3TESTA1-CP67X -Status Valid

<table>
<thead>
<tr>
<th>Name</th>
<th>Valid</th>
<th>ValidityCode</th>
<th>ExpireTime</th>
<th>Available</th>
<th>Used</th>
<th>VolumeNo</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC3 C++ / MatSim</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TC3 CNC</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TC3 Target For Matlab Simulink</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TC3 CNC Axis</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>00</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Connect to the License Server on target 'TC3TESTa1-CP67X' and return all valid licenses.

**EXAMPLE 4**

```
> Get-TcLicense -NetId 172.17.60.153.1.1 -Status Invalid | format-list
```

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>ExpireTime</th>
<th>ValidityCode</th>
<th>Valid</th>
<th>AvailableLicenses</th>
<th>UsedLicenses</th>
<th>VolumeNo</th>
</tr>
</thead>
<tbody>
<tr>
<td>4c256767-e6e6-4af5-bd68-9f7abad0c200</td>
<td>TC3 ADS</td>
<td>8/17/2017 12:00:00 AM</td>
<td>Expired</td>
<td>False</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>ExpireTime</th>
<th>ValidityCode</th>
<th>Valid</th>
<th>AvailableLicenses</th>
<th>UsedLicenses</th>
<th>VolumeNo</th>
</tr>
</thead>
<tbody>
<tr>
<td>66689887-ccbd-452c-ac9a-039d997c6e66</td>
<td>TC3 PLC</td>
<td>8/17/2017 12:00:00 AM</td>
<td>Expired</td>
<td>False</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>ExpireTime</th>
<th>ValidityCode</th>
<th>Valid</th>
<th>AvailableLicenses</th>
<th>UsedLicenses</th>
<th>VolumeNo</th>
</tr>
</thead>
<tbody>
<tr>
<td>3ff18e97-7754-401b-93fb-70544de28a13</td>
<td>TC3 IO</td>
<td>8/17/2017 12:00:00 AM</td>
<td>Expired</td>
<td>False</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Connect to NetId 172.17.60.153.1.1, determine all invalid licenses and format the result into a list.

**EXAMPLE 5**

```
> Get-TcLicense -OrderId TE* | format-list
```

<table>
<thead>
<tr>
<th>OrderID</th>
<th>Name</th>
<th>Valid</th>
<th>ValidityCode</th>
<th>ExpireTime</th>
<th>Available</th>
<th>Used</th>
<th>VolumeNo</th>
</tr>
</thead>
<tbody>
<tr>
<td>TE1400</td>
<td>TC3 Target For Matlab Simulink</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TE1500</td>
<td>TC3 Valve-Diagram-Editor</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TE1120</td>
<td>TC3 XCAD Interface</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TE1510</td>
<td>TC3 Cam-Design-Tool</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TE1110</td>
<td>TC3 Simulation Manager</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TE1410</td>
<td>TC3 Interface For Matlab Simulink</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TE1300</td>
<td>TC3 Scope View Professional</td>
<td>X</td>
<td>Valid</td>
<td>CPU License</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Get the valid licenses from local system and filter them for OrderIds starting with TE*.

**PARAMETERS**

- **-Name**
  The name of the license to get.
  Wildcards are permitted.
**Type**: `String[]`  
**Parameter Sets**: (All)  
**Aliases:**

- **Required**: False  
- **Position**: Named  
- **Default value**: None  
- **Accept pipeline input**: False  
- **Accept wildcard characters**: True

---

**-OrderId**

The OrderID of the license.

Wildcards are permitted.

**Type**: `String[]`  
**Parameter Sets**: (All)  
**Aliases:**

- **Required**: False  
- **Position**: Named  
- **Default value**: None  
- **Accept pipeline input**: False  
- **Accept wildcard characters**: True

---

**-NetId**

The NetID address of the target system where to load the licenses (Local by default).

**Type**: `AmsNetId`  
**Parameter Sets**: NetIdPort  
**Aliases:**

- **Required**: False  
- **Position**: Named  
- **Default value**: `172.17.60.167.1.1`  
- **Accept pipeline input**: False  
- **Accept wildcard characters**: False

---

**-Route**

The Route object where to load the licenses from (RouteTarget.Local by default).

**Type**: `IRoute`  
**Parameter Sets**: Route  
**Aliases**: Destination

- **Required**: True  
- **Position**: Named  
- **Default value**: None  
- **Accept pipeline input**: True (ByValue)  
- **Accept wildcard characters**: False

---

**-Address**

The address where to load the licenses.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

**Type**: `String`  
**Parameter Sets**: AddressStr  
**Aliases:**

- **Required**: True  
- **Position**: Named  
- **Default value**: None  
- **Accept pipeline input**: False  
- **Accept wildcard characters**: True
-**InputObject**

The session object to use for license upload.

This must target port 30 (AmsPort.R0_LicenseServer).

<table>
<thead>
<tr>
<th>Type: ISession</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: Session</td>
</tr>
<tr>
<td>Aliases: Session</td>
</tr>
<tr>
<td>Required: True</td>
</tr>
<tr>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: None</td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
</tr>
</tbody>
</table>

-**SessionId**

The unique session Identifier that represents the session to use for the license upload.

<table>
<thead>
<tr>
<th>Type: Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: SessionId</td>
</tr>
<tr>
<td>Aliases: Id</td>
</tr>
<tr>
<td>Required: True</td>
</tr>
<tr>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: -1</td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
</tr>
</tbody>
</table>

-**Status**

The Status parameter selects the Licenses to return.

Available is 'Valid' (the valid licenses), 'Invalid' (the invalid licenses) and 'All' ('Valid' + 'Invalid') licenses.

The Default Value is 'All'

Possible values: None, Valid, Invalid, All

<table>
<thead>
<tr>
<th>Type: LicenseStatus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: (All)</td>
</tr>
<tr>
<td>Aliases:</td>
</tr>
<tr>
<td>Accepted values: None, Valid, Invalid, All</td>
</tr>
<tr>
<td>Required: False</td>
</tr>
<tr>
<td>Position: Named</td>
</tr>
<tr>
<td>Default value: All</td>
</tr>
<tr>
<td>Accept pipeline input: False</td>
</tr>
<tr>
<td>Accept wildcard characters: False</td>
</tr>
</tbody>
</table>

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

**INPUTS**

**TwinCAT.IRoute**

The Route object where to load the licenses from (RouteTarget.Local by default).
6.21 Get-TcRouterInfo

SYNOPSIS
Gets the router status information of the specified target system.

SYNTAX

NetIdPort (Default)
Get-TcRouterInfo [-NetId] <AmsNetId[]> [-Timeout <Int32>] [CommonParameters]

Route
Get-TcRouterInfo [-InputObject] <IRoute[]> [-Timeout <Int32>] [CommonParameters]

AddressStr
Get-TcRouterInfo [-Address] <String[]> [-Timeout <Int32>] [CommonParameters]

Session
Get-TcRouterInfo -Session <I3Session[]> [-Timeout <Int32>] [CommonParameters]

SessionId
Get-TcRouterInfo -SessionId <Int32[]> [-Timeout <Int32>] [CommonParameters]

DESCRIPTION
This Cmdlet gets status information from the specified target system.

To contact the target system, it must be available as actual route or must be the local system.

The status information contains the amount of overall router memory and the used memory.

Furthermore the number of active connections and the size of the actual router mailbox will be shown.

EXAMPLES

EXAMPLE 1
PS> Get-TcRouterInfo
Target Result TotalMem(kb) AvailMem(kb) Ports Drivers Transports Mailbox Size(kb) Mailbox Queue
-------- ------ --------------- ------------ --- ----- ------- ---------- ----------- ------ ----------------
CX_1234 Ok 32768 32759 31 4 11 0 0

Get router information from the local system.
PARAMETERS

- **NetId**

NetId(s) of the target system.

<table>
<thead>
<tr>
<th>Type:</th>
<th>AmsNetId[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>NetIdPort</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
</tbody>
</table>

Required: False
Position: 1
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

- **InputObject**

The route object where to get the Target information from..

<table>
<thead>
<tr>
<th>Type:</th>
<th>IRoute[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>Route</td>
</tr>
<tr>
<td>Aliases:</td>
<td>Destination, Route</td>
</tr>
</tbody>
</table>

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

- **Address**

Target names/addresses.

These can consist of RouteName, NetID, HostName or IPAddress.

Wildcards are permitted.

<table>
<thead>
<tr>
<th>Type:</th>
<th>String[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>AddressStr</td>
</tr>
<tr>
<td>Aliases:</td>
<td>Name</td>
</tr>
</tbody>
</table>

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

- **Session**

The Session to use for the value read.

<table>
<thead>
<tr>
<th>Type:</th>
<th>ISession[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>Session</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
</tbody>
</table>

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName)
Accept wildcard characters: False

- **SessionId**

Specifies the Session (with unique ID) to use for the value read.

<table>
<thead>
<tr>
<th>Type:</th>
<th>Int32[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>SessionId</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
</tbody>
</table>

Required: True
- **Timeout**

  Timeout of the separate ADS Read operations

  Type: Int32

  Parameter Sets: (All)

  Required: False

  Position: Named

  Default value: 2500

  Accept pipeline input: False

  Accept wildcard characters: False

---

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

**INPUTS**

**TwinCAT.IRoute[]**

The route object where to get the Target information from.

**TwinCAT.ISession[]**

The Session to use for the value read.

**OUTPUTS**

**NOTES**

---

### 6.22 Get-TcSession

**SYNOPSIS**

List the currently established Sessions.

**SYNTAX**

**Default (Default)**

Get-TcSession [-Force] [CommonParameters]

<table>
<thead>
<tr>
<th>Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get-TcSession -Id &lt;Int32&gt; [-Force] [CommonParameters]</td>
</tr>
</tbody>
</table>

**DESCRIPTION**

This Cmdlet lists all actually Point-To-Point connections to TwinCAT Targets in form of their session representation.
Different types of Sessions can be accessed via the registered types of SessionProviders (e.g. ADS, MQTT, OPC).

EXAMPLES

EXAMPLE 1

PS > Get-TcSession

PARAMETERS

-Id

Specifies the ID of the session to get.

Type: Int32
Parameter Sets: Id
Aliases: SessionID

Required: True
Position: Named
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False

-Force

Forces the Cmdlet to determine also the internal used sessions.

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

OUTPUTS

NOTES

6.23 Get-TcSymbol

SYNOPSIS

Get the symbols from a TwinCAT target system / Device.
SYNTAX

NetIdPort (Default)
Get-TcSymbol [[-Path] <String[]>] [-NetId <AmsNetId>] -Port <Int32> [-Recurse] [-Force] [<CommonParameters>]

AddressStr
Get-TcSymbol [[-Path] <String[]>] -Address <String> -Port <Int32> [-Recurse] [-Force] [<CommonParameters>]

Route
Get-TcSymbol [[-Path] <String[]>] -Route <IRoute> -Port <Int32> [-Recurse] [-Force] [<CommonParameters>]

Session
Get-TcSymbol [[-Path] <String[]>] -InputObject <ISession> [-Recurse] [-Force] [<CommonParameters>]

SessionId
Get-TcSymbol [[-Path] <String[]>] -SessionId <Int32> [-Recurse] [-Force] [<CommonParameters>]

NetIdPortLiteral
Get-TcSymbol -LiteralPath <String[]> [-NetId <AmsNetId>] -Port <Int32> [-Recurse] [-Force] [<CommonParameters>]

AddressStrLiteral
Get-TcSymbol -LiteralPath <String[]> -Address <String> -Port <Int32> [-Recurse] [-Force] [<CommonParameters>]

RouteLiteral
Get-TcSymbol -LiteralPath <String[]> -Route <IRoute> -Port <Int32> [-Recurse] [-Force] [<CommonParameters>]

SessionLiteral
Get-TcSymbol -LiteralPath <String[]> -InputObject <ISession> [-Recurse] [-Force] [<CommonParameters>]

SessionIdLiteral
Get-TcSymbol -LiteralPath <String[]> -SessionId <Int32> [-Recurse] [-Force] [<CommonParameters>]

DESCRIPTION

This Cmdlet get the symbolic information from a target system if symbols are provided.

The information can be determined via different Providers (e.g. ADS, MQTT, OPC).
EXAMPLES

EXAMPLE 1
PS> Get-TcSymbol -port 851

<table>
<thead>
<tr>
<th>InstanceName</th>
<th>DataType</th>
<th>Size</th>
<th>InstancePath</th>
</tr>
</thead>
<tbody>
<tr>
<td>tc2vBool</td>
<td>BOOL</td>
<td>1</td>
<td>tc2vBool</td>
</tr>
<tr>
<td>tc2vInt</td>
<td>INT</td>
<td>2</td>
<td>tc2vInt</td>
</tr>
<tr>
<td>Constants</td>
<td></td>
<td>0</td>
<td>Constants</td>
</tr>
<tr>
<td>GVL</td>
<td></td>
<td>0</td>
<td>GVL</td>
</tr>
<tr>
<td>MAIN</td>
<td></td>
<td>0</td>
<td>MAIN</td>
</tr>
<tr>
<td>Slow</td>
<td></td>
<td>0</td>
<td>Slow</td>
</tr>
<tr>
<td>TwinCAT_SystemInfoVarList</td>
<td></td>
<td>0</td>
<td>TwinCAT_SystemInfoVarList</td>
</tr>
</tbody>
</table>

Get the root symbolic information from the local system (Port 851):

EXAMPLE 2
PS>$session = New-TcSession -NetId 1.2.3.4.5.6 -Port 851
PS>$session | Get-TcSymbol "TwinCAT_SystemInfoVarList._AppInfo" -recurse

<table>
<thead>
<tr>
<th>InstanceName</th>
<th>DataType</th>
<th>Size</th>
<th>InstancePath</th>
</tr>
</thead>
<tbody>
<tr>
<td>_AppInfo</td>
<td>PLC.PlcAppSystemInfo</td>
<td>256</td>
<td>TwinCAT_SystemInfoVarList._AppInfo</td>
</tr>
<tr>
<td>ObjId</td>
<td>OTCID</td>
<td>4</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ObjId</td>
</tr>
<tr>
<td>TaskCnt</td>
<td>UDINT</td>
<td>4</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.TaskCnt</td>
</tr>
<tr>
<td>OnlineChangeCnt</td>
<td>UDINT</td>
<td>4</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.OnlineChangeCnt</td>
</tr>
<tr>
<td>Flags</td>
<td>DWORD</td>
<td>4</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.Flags</td>
</tr>
<tr>
<td>AdsPort</td>
<td>UINT</td>
<td>2</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.AdsPort</td>
</tr>
<tr>
<td>BootDataLoaded</td>
<td>BOOL</td>
<td>1</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.BootDataLoaded</td>
</tr>
<tr>
<td>OldBootData</td>
<td>BOOL</td>
<td>1</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.OldBootData</td>
</tr>
<tr>
<td>AppTimestamp</td>
<td>DT</td>
<td>4</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.AppTimestamp</td>
</tr>
<tr>
<td>KeepOutputsOnBP</td>
<td>BOOL</td>
<td>1</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.KeepOutputsOnBP</td>
</tr>
<tr>
<td>ShutdownInProgress</td>
<td>BOOL</td>
<td>1</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ShutdownInProgress</td>
</tr>
<tr>
<td>LicensesPending</td>
<td>BOOL</td>
<td>1</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.LicensesPending</td>
</tr>
<tr>
<td>BSDOOccupied</td>
<td>BOOL</td>
<td>1</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.BSDOOccupied</td>
</tr>
<tr>
<td>TComSrvPtr</td>
<td>ITComObjectServer</td>
<td>4</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.TComSrvPtr</td>
</tr>
<tr>
<td>AppName</td>
<td>STRING(63)</td>
<td>64</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.AppName</td>
</tr>
<tr>
<td>ProjectName</td>
<td>STRING(63)</td>
<td>64</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ProjectName</td>
</tr>
</tbody>
</table>

Create a session to the target system '1.2.3.4.5.6' Port: 851 and get the symbol 'TwinCAT_SystemInfoVarList._AppInfo' and its subsymbols recursively.

EXAMPLE 3
PS> $session = New-TcSession -Name 'CX_123456' -port 851
PS> $session | Get-TcSymbol -recurse | where InstanceName -like 'Project*'  

<table>
<thead>
<tr>
<th>InstanceName</th>
<th>DataType</th>
<th>Size</th>
<th>InstancePath</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProjectName</td>
<td>STRING(63)</td>
<td>64</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ProjectName</td>
</tr>
</tbody>
</table>

Gets an ADS-Session/Connection to the target system CX_123456 on port 851, downloads the symbol information recursively and returns all Instances where the instance name is like the pattern 'Project*'.

EXAMPLE 4
PS> $s = New-TcSession -port 851
PS> $s | Get-TcSymbol -path '.tc2vStructArray[0]', '.tc2vStructArray[1]'  

<table>
<thead>
<tr>
<th>InstancePath</th>
<th>Category</th>
<th>DataType</th>
<th>Size</th>
<th>Static</th>
<th>Persistant</th>
<th>IG</th>
<th>IO</th>
</tr>
</thead>
<tbody>
<tr>
<td>.tc2vStructArray[0]</td>
<td>Struct</td>
<td>ST_SimpleStruct</td>
<td>165</td>
<td>False</td>
<td>False</td>
<td>4040</td>
<td>117942</td>
</tr>
<tr>
<td>.tc2vStructArray[1]</td>
<td>Struct</td>
<td>ST_SimpleStruct</td>
<td>165</td>
<td>False</td>
<td>False</td>
<td>4040</td>
<td>117987</td>
</tr>
</tbody>
</table>

Gets an ADS-Session/connection to the local system PLC (Port 851) and get two Array Elements.

Because the -path parameter uses the wildcard parameter "[" and "]" these characters must be escaped with backtick '\"'.

EXAMPLE 5

```
PS> $s = New-TcSession -port 851
PS> $s | Get-TcSymbol -literalPath '.tc2vStructArray[0]', '.tc2vStructArray[1]'
```

<table>
<thead>
<tr>
<th>InstancePath</th>
<th>Category</th>
<th>DataType</th>
<th>Size</th>
<th>Static</th>
<th>Persistent</th>
<th>IG</th>
<th>IO</th>
</tr>
</thead>
<tbody>
<tr>
<td>.tc2vStructArray[0]</td>
<td>Struct</td>
<td>ST_SimpleStruct</td>
<td>165</td>
<td>False</td>
<td>False</td>
<td>4040</td>
<td>117942</td>
</tr>
<tr>
<td>.tc2vStructArray[1]</td>
<td>Struct</td>
<td>ST_SimpleStruct</td>
<td>165</td>
<td>False</td>
<td>False</td>
<td>4040</td>
<td>1179E7</td>
</tr>
</tbody>
</table>

Gets an ADS-Session/connection to the local system PLC (Port 851) and gets two Array Elements with their literal path.

PARAMETERS

- **Path**

  The instance path of the Symbol(s).

  Because wildcards are permitted with path, the wildcard parameters '\[' and '\]' must be escaped with a backtick.

  Type: String[]([])  
  Parameter Sets: NetIdPort, AddressStr, Route, Session, SessionId  
  Aliases:

  Required: False  
  Position: 0  
  Default value: None  
  Accept pipeline input: False  
  Accept wildcard characters: True

- **LiteralPath**

  The literal instance path of the symbol.

  Type: String[]([])  
  Parameter Sets: NetIdPortLiteral, AddressStrLiteral, RouteLiteral, SessionLiteral, SessionIdLiteral  
  Aliases:

  Required: True  
  Position: Named  
  Default value: None  
  Accept pipeline input: False  
  Accept wildcard characters: False

- **NetId**

  The target system NetId.

  Type: AmsNetId  
  Parameter Sets: NetIdPort, NetIdPortLiteral  
  Aliases:

  Required: False  
  Position: Named  
  Default value: 172.17.60.167.1.1  
  Accept pipeline input: False  
  Accept wildcard characters: False

- **Route**

  The target system route.

  Type: IRoute  
  Parameter Sets: Route, RouteLiteral  
  Aliases:

  Required: True  
  Position: Named  
  Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Address

The address for the target system where to get the symbol. This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

Type: String
Parameter Sets: AddressStr, AddressStrLiteral
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-Port

The target system port.

Type: Int32
Parameter Sets: NetIdPort, AddressStr, Route, NetIdPortLiteral, AddressStrLiteral, RouteLiteral
Aliases:

Required: True
Position: Named
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False

-InputObject

The session object that is used to get the symbols.

Type: ISession
Parameter Sets: Session, SessionLiteral
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-SessionId

The unique id of the session object that is used to get the symbols.

Type: Int32
Parameter Sets: SessionId, SessionIdLiteral
Aliases: Id

Required: True
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

-Recurse

Gets the symbol recursively. Often used in conjunction with Wildcards in -Path
Type: SwitchParameter  
Parameter Sets: (All)  
Aliases:

Required: False  
Position: Named  
Default value: False  
Accept pipeline input: False  
Accept wildcard characters: False

-Force

Active only in recursive mode - ignored otherwise.

This parameter forces the Cmdlet to output all symbols - even Array Elements.

Please take care because the output can be very lengthy dependent on the Size of the Array.

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession  
The session object that is used to get the symbols.

OUTPUTS

NOTES

6.24 Get-TcTargetInfo

SYNOPSIS

Get TwinCAT Device Target information.

SYNTAX

NetIdPort (Default)

Get-TcTargetInfo [[-NetId] <AmsNetId[]>] [-Timeout <Int32>] [CommonParameters]

Route

Get-TcTargetInfo [-InputObject] <IRoute[]> [-Timeout <Int32>] [CommonParameters]
**DESCRIPTION**

This Cmdlet gets information from the specified target system.

To contact the target system, it must be available as actual route.

The information contains the TargetName, TwinCAT Version, Running Operating system, CPU Architecture and Image Information.

**EXAMPLES**

**EXAMPLE 1**

```powershell
PS > Get-TcTargetInfo
```

<table>
<thead>
<tr>
<th>Target</th>
<th>Version</th>
<th>Level</th>
<th>OS</th>
<th>Image</th>
<th>Device</th>
<th>CPUArch</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC3TESTA1-CP67X</td>
<td>3.1.4021.54</td>
<td>CP</td>
<td>Win7</td>
<td></td>
<td></td>
<td>IntelX86</td>
</tr>
</tbody>
</table>

Get the target information of th localSystem system.

**EXAMPLE 2**

```powershell
PS > get-adsRoute | Get-TcTargetInfo
```

<table>
<thead>
<tr>
<th>Target</th>
<th>Version</th>
<th>Level</th>
<th>OS</th>
<th>Image</th>
<th>Device</th>
<th>CPUArch</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP-15ECA0</td>
<td>3.1.4021.50</td>
<td>CP</td>
<td>Win7</td>
<td></td>
<td></td>
<td>IntelX86</td>
</tr>
<tr>
<td>TC3TESTA1-CP67X</td>
<td>3.1.4021.54</td>
<td>CP</td>
<td>Win7</td>
<td></td>
<td></td>
<td>IntelX86</td>
</tr>
</tbody>
</table>

Get the target information of the actual connected routes.

**PARAMETERS**

- **-NetId**

NetId(s) of the target system.

Type: AmsNetId[
Parameter Sets: NetIdPort
Aliases:

- Required: False
- Position: 1
- Default value: 172.17.60.167.1.1
- Accept pipeline input: False
- Accept wildcard characters: False

- **-InputObject**

The route object where to get the Target information from..
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Parameter Sets</th>
<th>Aliases</th>
<th>Required</th>
<th>Position</th>
<th>Default value</th>
<th>Accept pipeline input</th>
<th>Accept wildcard characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Address</td>
<td>String[]</td>
<td>AddressStr</td>
<td>Name</td>
<td>True</td>
<td>1</td>
<td>None</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>-Session</td>
<td>ISession[]</td>
<td>Session</td>
<td></td>
<td>True</td>
<td>Named</td>
<td>None</td>
<td>True (ByValue)</td>
<td>False</td>
</tr>
<tr>
<td>-SessionId</td>
<td>Int32[]</td>
<td>SessionId</td>
<td></td>
<td>True</td>
<td>Named</td>
<td>None</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>-Timeout</td>
<td>Int32</td>
<td>(All)</td>
<td></td>
<td>False</td>
<td>Named</td>
<td>-1</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>
CommonParameters
This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS
TwinCAT.IRoute[]
The route object where to get the Target information from..

TwinCAT.ISession[]
The Session to use for the value read.

OUTPUTS

NOTES
6.25 Get-TcVersion

SYNOPSIS
Get the TwinCAT Version of a target system.

SYNTAX
NetIdPort (Default)
Get-TcVersion [[-NetId] <AmsNetId[]>] [-Timeout <Int32>] [CommonParameters]

Route
Get-TcVersion [-Timeout <Int32>] [-InputObject] <IRoute[]> [CommonParameters]

AddressStr
Get-TcVersion [-Timeout <Int32>] [-Address] <String[]> [CommonParameters]

Session
Get-TcVersion [-Timeout <Int32>] -Session <ISession[]> [CommonParameters]

SessionId
Get-TcVersion [-Timeout <Int32>] -SessionId <Int32[]> [CommonParameters]

DESCRIPTION
This Cmdlet gets the TwinCAT version of the specified target version and returns the version object.
EXAMPLES

EXAMPLE 1
PS > Get-TcVersion

<table>
<thead>
<tr>
<th>Major</th>
<th>Minor</th>
<th>Build</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>4021</td>
<td>50</td>
</tr>
</tbody>
</table>

Get the TwinCAT version of the local system.

EXAMPLE 2
PS > Get-AdsRoute | Get-TcVersion

<table>
<thead>
<tr>
<th>Major</th>
<th>Minor</th>
<th>Build</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>4021</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4021</td>
<td>54</td>
</tr>
</tbody>
</table>

Get the TwinCAT version actual routes.

PARAMETERS

- **-NetId**
  The target address.
  Type: AmsNetId[]
  Parameter Sets: NetIdPort
  Aliases:
  Required: False
  Position: 1
  Default value: 172.17.60.167.1.1
  Accept pipeline input: False
  Accept wildcard characters: False

- **-Timeout**
  The Ads timeout in milliseconds.
  Type: Int32
  Parameter Sets: (All)
  Aliases:
  Required: False
  Position: Named
  Default value: -1
  Accept pipeline input: False
  Accept wildcard characters: False

- **-InputObject**
  The target routes where to determine the Version information.
  Type: IRoute[]
  Parameter Sets: Route
  Aliases: Destination, Route
  Required: True
  Position: 1
  Default value: None
  Accept pipeline input: True (ByValue)
  Accept wildcard characters: False
**-Address**
Addresses where to determine the Version information.
The Addresses can consist of NetId, IPAddress or HostName.
Wildcards are permitted.

| Type: String[] |
| Parameter Sets: AddressStr |
| Aliases: Name |
| Required: True |
| Position: 1 |
| Default value: None |
| Accept pipeline input: False |
| Accept wildcard characters: True |

**-Session**
The Session to use for the Cmdlet.

| Type: ISession[] |
| Parameter Sets: Session |
| Aliases: |
| Required: True |
| Position: Named |
| Default value: None |
| Accept pipeline input: True (ByValue) |
| Accept wildcard characters: False |

**-SessionId**
Specifies the Session (with unique ID) to use for the Cmdlet.

| Type: Int32[] |
| Parameter Sets: SessionId |
| Aliases: |
| Required: True |
| Position: Named |
| Default value: None |
| Accept pipeline input: False |
| Accept wildcard characters: False |

**CommonParameters**
This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

**INPUTS**

**TwinCAT.IRoute[]**
The target routes where to determine the Version information.

**TwinCAT.ISession[]**
The Session to use for the Cmdlet.
6.26 New-TcSession

SYNOPSIS
Create a new session to a TwinCAT Target.

SYNTAX
NetIdPort (Default)
New-TcSession [[-NetId] <AmsNetId>] [-Port] <Int32> [-Force] [<CommonParameters>]

Route
New-TcSession -InputObject <IRoute> [-Port] <Int32> [-Force] [<CommonParameters>]

AddressStr
New-TcSession [-Provider <String>] [-Address] <String> [[-Port] <Int32>] [-Force] [<CommonParameters>]

DESCRIPTION
Creates a new Point-To-Point Connection to a TwinCAT Target that is represented by the returned session object.

Different types of Sessions can be accessed by the registered types of SessionProviders (e.g. ADS, MQTT, OPC).

EXAMPLES
EXAMPLE 1
PS> $route = Get-AdsRoute -Name "Tc3*"
PS> $session = New-TcSession -Route $route -Port 851
PS> $session

ID Address IsConnected EstablishedAt
-- ------- ----------- -------------
5 172.17.62.105.1.1:851 True 12/12/2016 12:22:02 PM

Establishes a new ADS Session/Connection to the specified route destination that has the name pattern "tc3*" via port 851 (PLC1)

EXAMPLE 2
PS> New-TcSession -NetId '172.17.62.105.1.1' -port 851

ID Address IsConnected EstablishedAt
-- ------- ----------- -------------
5 172.17.62.105.1.1:851 True 12/12/2016 12:22:02 PM

Establishes a new Ads Session/Connection to the specified NetId/Port address.
EXAMPLE 3

PS> New-TcSession -Name 'CX_123456' -port 851

<table>
<thead>
<tr>
<th>ID</th>
<th>Address</th>
<th>IsConnected</th>
<th>EstablishedAt</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>172.17.62.105.1.1:851</td>
<td>True</td>
<td>12/12/2016 12:22:02 PM</td>
</tr>
</tbody>
</table>

Establishes a new Ads Session/Connection to the target system with the Name/HostName 'CX_123456' (Port 851).

PARAMETERS

-NetId

The NetID Address

Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: 1
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

-InputObject

The route target object.

Type: IRoute
Parameter Sets: Route
Aliases: Destination, Route

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-Provider

Selects the session provider registered on the System (ADS by default)

Type: String
Parameter Sets: AddressStr
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Address

The target address of the new session.

This can be the NetId, the HostName or the IPAddress.

Wildcards are permitted.

Type: String
Parameter Sets: AddressStr
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-Port
The AmsPort Address of the new session.
Type: Int32
Parameter Sets: NetIdPort, Route
Aliases:
Required: True
Position: 2
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False

Type: Int32
Parameter Sets: AddressStr
Aliases:
Required: False
Position: 2
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False

-Force
Forces to create the session independant of ReachableRoutes
Type: SwitchParameter
Parameter Sets: (All)
Aliases:
Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

CommonParameters
This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS
TwinCAT.IRoute
The route target object.

OUTPUTS

NOTES

6.27 Read-TcValue

SYNOPSIS
Reads values from TwinCAT devices.
SYNTAX

NetIdPortSymbol (Default)
Read-TcValue [-NetId <AmsNetId[]>] -Port <Int32> [-Path <String>] [-Extended] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]

NetIdPortIndexed
Read-TcValue [-NetId <AmsNetId[]>] -Port <Int32> [-IndexGroup] <UInt32> [-IndexOffset] <UInt32> [-Size] <Int32> [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]

NetIdPortIndexedTyped
Read-TcValue [-NetId <AmsNetId[]>] -Port <Int32> [-IndexGroup] <UInt32> [-IndexOffset] <UInt32> [-ValueType] <Type> [-Size] <Int32> [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]

RouteIndexed
Read-TcValue -Route <IRoute[]> -Port <Int32> [-IndexGroup] <UInt32> [-IndexOffset] <UInt32> [-Size] <Int32> [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]

RouteIndexedTyped
Read-TcValue -Route <IRoute[]> -Port <Int32> [-IndexGroup] <UInt32> [-IndexOffset] <UInt32> [-ValueType] <Type> [-Size] <Int32> [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]

RouteSymbol
Read-TcValue -Route <IRoute[]> -Port <Int32> [-Path <String>] [-Extended] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]

AddressIndexed
Read-TcValue -Address <String[]> -Port <Int32> [-IndexGroup] <UInt32> [-IndexOffset] <UInt32> [-Size] <Int32> [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]

AddressIndexedTyped
Read-TcValue -Address <String[]> -Port <Int32> [-IndexGroup] <UInt32> [-IndexOffset] <UInt32> [-ValueType] <Type> [-Size] <Int32> [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]

AddressSymbol
Read-TcValue -Address <String[]> -Port <Int32> [-Path <String>] [-Extended] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]

SessionIndexed
Read-TcValue -Session <ISession[]> [-IndexGroup] <UInt32> [-IndexOffset] <UInt32> [-Size] <Int32> [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]
**SessionIndexedTyped**

```
```

**SessionSymbol**

```
Read-TcValue -Session <ISession[]> [-Path] <String> [-Extended] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]
```

**SessionIdIndexed**

```
Read-TcValue -SessionId <Int32[]> [-IndexGroup] <UInt32> [[-IndexOffset] <UInt32>] [-Size] <Int32> [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]
```

**SessionIdIndexedTyped**

```
```

**SessionIdSymbol**

```
Read-TcValue -SessionId <Int32[]> [-Path] <String> [-Extended] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]
```

**InputObject**

```
Read-TcValue [-InputObject] <ISymbol> [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [CommonParameters]
```

**DESCRIPTION**

This Cmdlet read values from TwinCAT Devices.

The devices can be accessed via different ValueProviders.

**EXAMPLES**

**EXAMPLE 1**

```
PS> $session = New-TcSession -NetId '1.2.3.4.5.6' -Port 851
PS> $symbol = $session | Get-TcSymbol -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'
PS> $symbol | Read-TcValue

ADS_DynSymbols
```

Create an ADS Session/Connection, determine the 'ProjectName' Symbol from the running PLC Project, read the current value of the symbol and print it to the console.

**EXAMPLE 2**

```
PS> Read-TcValue -IndexGroup 0x4040 -IndexOffset 0x1247a8 -NetId 172.17.62.105.1.1 -port 851 -size 0xff | format-hex
```

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 41 44 53 5F 44 79 6E 53 79 6D 62 6F 6C 73 00 00 ADS_DynSymbols..
EXAMPLE 3

PS> Read-TcValue -session $session -IndexGroup 0x4040 -IndexOffset 0x1247A8 -ValueType String ADS_DynSymbols

EXAMPLE 4

PS> $route = Get-AdsRoute -Name 'CX-123456'
PS> $session = $route | New-TcSession -Port 851
PS> $handle = $session | Send-TcReadWrite -IndexGroup SymbolHandleByName -WriteValue "GVL.vgInt" -ReadType Int32 -force
PS> $session | Read-TcValue -IndexGroup SymbolValueByHandle -IndexOffset $handle -ValueType Int16
42

PARAMETERS

-NetId

The NetId part of the AmsAddress for the value read.

-Route

Specifies the target system(s) to read value from.

-Address

The Address(es) of the system(s) where to read the value.
The Address can consist of NetId, IPAddress or HostName.

Wildcards are permitted.

Type: String[]
Parameter Sets: AddressIndexed, AddressIndexedTyped, AddressSymbol
Aliases:

- Required: True
- Position: Named
- Default value: None
- Accept pipeline input: False
- Accept wildcard characters: True

-Session

The Session to use for the value read.

Type: ISession[]
Parameter Sets: SessionIndexed, SessionIndexedTyped, SessionSymbol
Aliases:

- Required: True
- Position: Named
- Default value: None
- Accept pipeline input: True (ByPropertyName, ByValue)
- Accept wildcard characters: False

-SessionId

Specifies the Session (with unique ID) to use for the value read.

Type: Int32[]
Parameter Sets: SessionIdIndexed, SessionIdIndexedTyped, SessionIdSymbol
Aliases:

- Required: True
- Position: Named
- Default value: None
- Accept pipeline input: False
- Accept wildcard characters: False

-Port

The address Port to use for the value read.

Type: Int32
Parameter Sets: NetIdPortSymbol, NetIdPortIndexed, NetIdPortIndexedTyped, RouteIndexed, RouteIndexedTyped, RouteSymbol, AddressIndexed, AddressIndexedTyped, AddressSymbol
Aliases:

- Required: True
- Position: Named
- Default value: 10000
- Accept pipeline input: False
- Accept wildcard characters: False

-IndexGroup

The IndexGroup of the Symbol to read from target system.

Only for IndexGroup/IndexOffset access.

Type: UInt32
Parameter Sets: NetIdPortIndexed, NetIdPortIndexedTyped, RouteIndexed, RouteIndexedTyped, AddressIndexed, AddressIndexedTyped, SessionIndexed, SessionIndexedTyped, SessionIdIndexed, SessionIdIndexedTyped
Aliases: IG

- Required: True
- Position: 1
- Default value: 0
Accept pipeline input: False
Accept wildcard characters: False

- **IndexOffset**

The IndexOffset of the Symbol to read from the target system.

Only for IndexGroup/IndexOffset access.

Type: UInt32
Parameter Sets: NetIdPortIndexed, NetIdPortIndexedTyped, RouteIndexed, RouteIndexedTyped, AddressIndexed, AddressIndexedTyped, SessionIndexed, SessionIndexedTyped, SessionIdIndexed, SessionIdIndexedTyped
Aliases: IO

Required: False
Position: 2
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False

- **ValueType**

The dataType of the Value for a 'ReadAny' access.

Only usable with IndexGroup/IndexOffset access.

Type: Type
Parameter Sets: NetIdPortIndexedTyped, RouteIndexedTyped, AddressIndexedTyped, SessionIndexedTyped, SessionIdIndexedTyped
Aliases: Type, ReadType

Required: True
Position: 3
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

- **Size**

The 'Size' of Value (in bytes) to read.

Type: Int32
Parameter Sets: NetIdPortIndexed, NetIdPortIndexedTyped, RouteIndexed, AddressIndexed, SessionIndexed, SessionIdIndexed
Aliases: ReadSize, Length

Required: True
Position: 3
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

Type: Int32
Parameter Sets: RouteIndexedTyped, AddressIndexedTyped, SessionIndexedTyped, SessionIdIndexedTyped
Aliases: ReadSize, Length

Required: False
Position: 3
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

- **Path**

The instance path of the symbol to read (symbolic access).

This parameter supports wildcards.
Type: String
Parameter Sets: NetIdPortSymbol, RouteSymbol, AddressSymbol, SessionSymbol, SessionIdSymbol
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-InputObject

The symbol object to read value from.

Type: ISymbol
Parameter Sets: InputObject
Aliases: Symbol

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-Extended

Switch on 'ExtendedMode', what means that primitive values are not resolved to their primitive managed (powershell) counterparts, but still contain rich metadata as DynamicValues.

Type: SwitchParameter
Parameter Sets: NetIdPortSymbol, RouteSymbol, AddressSymbol, SessionSymbol, SessionIdSymbol
Aliases: FullMetadata

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Force

Force reading value.

This flag bypasses the FailFastInterceptor to retry communication in every case.

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Timeout

Communication Timeout in milliseconds

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
- **Encoding**

Specifies the Encoding for strings.

The DefaultEncoding is ANSI with actual code page.

<table>
<thead>
<tr>
<th>Type: Encoding</th>
<th>Parameter Sets: (All)</th>
<th>Aliases:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept pipeline input: False</td>
<td>Accept wildcard characters: False</td>
<td></td>
</tr>
</tbody>
</table>

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

**INPUTS**

TwinCAT.ISession[]

The Session to use for the value read.

TwinCAT.TypeSystem.ISymbol

The symbol object to read value from.

**OUTPUTS**

NOTES

6.28 **Register-AdsHandle**

**SYNOPSIS**

Registers and returns a symbol handle.

**SYNTAX**

**NetIdPortSymbol (Default)**

Register-AdsHandle [-NetId <AmsNetId>] -Port <Int32> [-Path] <String[]> [<CommonParameters>]

**RouteSymbol**

Register-AdsHandle -Route <IRoute> -Port <Int32> [-Path] <String[]> [<CommonParameters>]

**AddressSymbol**

Register-AdsHandle -Address <String> -Port <Int32> [-Path] <String[]> [<CommonParameters>]

**SessionSymbol**

Register-AdsHandle -Session <ISession> [-Path] <String[]> [<CommonParameters>]
SessionIdSymbol
Register-AdsHandle -SessionId <Int32> [-Path] <String[]> [CommonParameters]

InputObject
Register-AdsHandle [-InputObject] <ISymbol[]> [CommonParameters]

DESCRIPTION
This Cmdlet registers a symbol handle at the connected system.

The handle is returned as AdsHandleInfo.

EXAMPLES
EXAMPLE 1
PS> $session = New-TcSession -NetId '1.2.3.4.5.6' -Port 851
PS> $handleInfo = $session | get-AdsHandle -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'
PS> $handleInfo = register-AdsHandle -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName' -Session $s
PS> $handleInfo
InstancePath Result Handle
------------ ------- ------
TwinCAT_SystemInfoVarList._AppInfo.ProjectName NoError 0x428000FC (1115685116)
PS> read-TcValue -Session $session -IndexGroup SymbolValueByHandle -IndexOffset $handleInfo.Handle -Type String
MyProject
PS> $handle | Unregister-AdsHandle -Session $session
PS> $session | Close-tcsession

Opens a new device session, registers a Symbol Handle to the ProjectName of the running PLC Project, Reads the value by handle unregisters the handle and closes the session again.

PARAMETERS
-NetId
The NetId address of the Target system
Type: AmsNetId
Parameter Sets: NetIdPortSymbol
Aliases:
Required: False
Position: Named
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

-Route
Specifies the route of the target system.
Type: IRoute
Parameter Sets: RouteSymbol
Aliases: Destination
Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
- **Address**

The Address of the target system where to register the symbol handle.

The Address can consist of RouteName, NetId, IPAddress or HostName.

Wildcards are permitted.

Type: String
Parameter Sets: AddressSymbol
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

- **Session**

The Session to use (instead of addressing the target system).

Type: ISession
Parameter Sets: SessionSymbol
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName, ByValue)
Accept wildcard characters: False

- **SessionId**

Specifies the Session (with unique ID) to use instead of specifying the target address.

Type: Int32
Parameter Sets: SessionIdSymbol
Aliases:

Required: True
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

- **Port**

The address Port to use (always in combination with the NetId).

ArgumentCompleter is supported.

Type: Int32
Parameter Sets: NetIdPortSymbol, RouteSymbol, AddressSymbol
Aliases:

Required: True
Position: Named
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False

- **Path**

The instance path to the symbol.

Type: String[]
Parameter Sets: NetIdPortSymbol, RouteSymbol, AddressSymbol, SessionSymbol, SessionIdSymbol
Aliases:

Required: True
Position: 1
**-InputObject**

The symbol object.

Type: ISymbol[]
Parameter Sets: InputObject
Aliases: Symbol

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

**INPUTS**

TwinCAT.ISession
The Session to use (instead of addressing the target system).

TwinCAT.TypeSystem.ISymbol[]
The symbol object.

**OUTPUTS**

**NOTES**

6.29 Register-AdsNatRoute

**SYNOPSIS**

Changes an standard Route to an AmsNAT route on the target system (obsolete).

**SYNTAX**

**Name (Default)**

Register-AdsNatRoute [-Name] <String> -NATNetId <AmsNetId> [-Destination <String>] [-Quiet] [-WhatIf] [-Confirm] [CommonParameters]

**NetId**

Register-AdsNatRoute [-NetId] <AmsNetId> -NATNetId <AmsNetId> [-Destination <String>] [-Quiet] [-WhatIf] [-Confirm] [CommonParameters]
DESCRIPTION

This Cmdlet Changes an standard Route to an AmsNAT route on the target system.

The route must be preexisting and the cmdlet adds the RemoteNetId/AmsNAT information to the StaticRoutes.xml of the destination system.

Afterwards the destination system needs a TwinCAT Restart.

For TwinCAT Versions >= 3.1.4024.11 (or newer), the Add-AdsRoute Cmdlet should be used with the -NAT Parameter as Replacement.

Therefore, this 'Register-AdsNatRoute' Cmdlet is classified as 'obsolete' and of limited use and could be removed in future.

EXAMPLES

EXAMPLE 1
PS> Register-AdsNatRoute -Name MyRoute -NATNetId 1.2.3.4.2.2

Adds an AmsNAT address translation to the existing route 'MyRoute' on the local system (e.g. from '1.2.3.4.1.1' to '1.2.3.4.2.2').

EXAMPLE 2
PS> Register-AdsNatRoute -NetId 1.2.3.4.1.1 -NATNetId 1.2.3.4.2.2 -Destination CX_1234

Adds an AmsNAT address translation to the existing route with NetId '1.2.3.4.1.1' to NATNetId '1.2.3.4.2.2' on System 'CX_1234'.

PARAMETERS

-Name

The Name of the Route where to add an AmsNAT entry.

Type: String
Parameter Sets: Name
Aliases:
Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-NetId

The NetId which specifies the existing route where to add an AmsNAT entry.

This NetId becomes the 'RemoteNetId' afterwards.

Type: AmsNetId
Parameter Sets: NetId
Aliases:
Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
**-NATNetId**

The NATNetId (the local representation of the remote system).

<table>
<thead>
<tr>
<th>Type</th>
<th>AmsNetId</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases</td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td>True</td>
</tr>
<tr>
<td>Position</td>
<td>Named</td>
</tr>
<tr>
<td>Default value</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard</td>
<td>False</td>
</tr>
</tbody>
</table>

**-Destination**

The Destination system, where the AmsNAT translation is added.

This Parameter allows RouteName, AmsNetId, IPAddress or HostName

<table>
<thead>
<tr>
<th>Type</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases</td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
</tr>
<tr>
<td>Position</td>
<td>Named</td>
</tr>
<tr>
<td>Default value</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard</td>
<td>False</td>
</tr>
</tbody>
</table>

**-Quiet**

The Quiet parameter suppresses the 'ShouldProcess' message and the routes will be removed without further question.

<table>
<thead>
<tr>
<th>Type</th>
<th>SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases</td>
<td>Silent</td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
</tr>
<tr>
<td>Position</td>
<td>Named</td>
</tr>
<tr>
<td>Default value</td>
<td>False</td>
</tr>
<tr>
<td>Accept pipeline</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard</td>
<td>False</td>
</tr>
</tbody>
</table>

**-Confirm**

Prompts you for confirmation before running the cmdlet.

<table>
<thead>
<tr>
<th>Type</th>
<th>SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases</td>
<td>cf</td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
</tr>
<tr>
<td>Position</td>
<td>Named</td>
</tr>
<tr>
<td>Default value</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard</td>
<td>False</td>
</tr>
</tbody>
</table>

**-WhatIf**

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

<table>
<thead>
<tr>
<th>Type</th>
<th>SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases</td>
<td>wi</td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
</tr>
<tr>
<td>Position</td>
<td>Named</td>
</tr>
</tbody>
</table>
**CommonParameters**

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

**INPUTS**

**OUTPUTS**

**NOTES**

### 6.30 Remove-AdsRoute

**SYNOPSIS**

Remove an ADS Route.

**SYNTAX**

**Address (Default)**

Remove-AdsRoute [-Destination <String>] [-Address] <String[]> [-Quiet] [-Credentials <PSCredential>] [-WhatIf] [-Confirm] [CommonParameters]

**NetId**

Remove-AdsRoute [-Destination <String>] [-NetId] <AmsNetId> [-Quiet] [-Credentials <PSCredential>] [-WhatIf] [-Confirm] [CommonParameters]

**Route**


**DESCRIPTION**

Removes static or temporary routes from the local system or from remote systems.

If access is available, the route is removed on both endpoints of the Route.

**EXAMPLES**

**EXAMPLE 1**

PS> Get-AdsRoute
Name           NetId          Address          Sub  TcVersion  RTSystem
----           -----          -------          ---  ---------  --------
CP-15ECA0      172.17.62.128 1.1  172.17.62.178  0.0  Unknown
TC3TESTA1-CP67X 172.17.62.105 1.1  172.17.62.105  0.0  Unknown

PS> Remove-AdsRoute -Name "CP-15ECA0","TC3TESTA1-CP67X"

Removes the Routes "CP-15ECA0" and "TC3TESTA1-CP67X" from the local system.
EXAMPLE 2

PS> Get-AdsRoute | Remove-AdsRoute -silent

Removes all registered routes from the local system.

PARAMETERS

-Destination

The destination address, where to Remove the specified route.

This can be the NetId, the HostName or the IPAddress

Type: String
Parameter Sets: (All)
Aliases:
Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Address

The address for the ADS route to remove.

This can be the NetId, the HostName or the IPAddress.

Wildcards are permitted.

Type: String[]
Parameter Sets: Address
Aliases: Name
Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-NetId

The NetID of the route to remove.

Type: AmsNetId
Parameter Sets: NetId
Aliases:
Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-InputObject

A collection of routes to remove (Pipeline support).

Type: ConfiguredRouteCollection
Parameter Sets: Route
Aliases:
Required: True
Position: 0
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
-Quiet

The Quiet parameter suppresses the 'ShouldProcess' message and the routes will be removed without further question.

<table>
<thead>
<tr>
<th>Type: SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: (All)</td>
</tr>
<tr>
<td>Aliases: Silent</td>
</tr>
</tbody>
</table>

Required: False  
Position: Named  
Default value: False  
Accept pipeline input: False  
Accept wildcard characters: False

-Credentials

Destination system route credentials (only if removing remotely).

<table>
<thead>
<tr>
<th>Type: PSCredential</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: (All)</td>
</tr>
<tr>
<td>Aliases:</td>
</tr>
</tbody>
</table>

Required: False  
Position: Named  
Accept pipeline input: False  
Accept wildcard characters: False

-Confirm

Prompts you for confirmation before running the cmdlet.

<table>
<thead>
<tr>
<th>Type: SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: (All)</td>
</tr>
<tr>
<td>Aliases: cf</td>
</tr>
</tbody>
</table>

Required: False  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

<table>
<thead>
<tr>
<th>Type: SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets: (All)</td>
</tr>
<tr>
<td>Aliases: wi</td>
</tr>
</tbody>
</table>

Required: False  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.
INPUTS

TwinCAT.ConfiguredRouteCollection
A collection of routes to remove (Pipeline support).

OUTPUTS

NOTES

6.31 Remove-MqttRoute

SYNOPSIS
Remove a MQQT Route.

SYNTAX

Address (Default)
[<CommonParameters>]

RouteInfo
Remove-MqttRoute -InputObject <MqttRoute[]> [-Destination <String>] [-Quiet] [-WhatIf] [-Confirm]
[<CommonParameters>]

DESCRIPTION
Removes a MQQT Route of the specified system.

EXAMPLES

EXAMPLE 1
PS> Remove-MqttRoute -address 1.2.3.4 -port 42 -Destination CX_1234

Removes the MQTT route on the destination System 'CX_1234' to MQTT Broker with Address '1.2.3.4' and Port '42'.

PARAMETERS

-Address
The IPAddress or HostName of the Mqtt broker system to remove.

Type: String
Parameter Sets: Address
Aliases:
Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
### -Port

The TCP/IP port specification.

| Type: Int32 |
| Parameter Sets: Address |
| Aliases: |
| Required: False |
| Position: 1 |
| Default value: -1 |
| Accept pipeline input: False |
| Accept wildcard characters: False |

### -InputObject

The Mqtt routes to remove.

| Type: MqttRoute[] |
| Parameter Sets: RouteInfo |
| Aliases: MqttRoute |
| Required: True |
| Position: Named |
| Default value: None |
| Accept pipeline input: True (ByValue) |
| Accept wildcard characters: False |

### -Destination

The destination address, where to Remove the specified Mqtt route.

This can be the NetId, the HostName or the IPAddress

| Type: String |
| Parameter Sets: (All) |
| Aliases: |
| Required: False |
| Position: Named |
| Default value: None |
| Accept pipeline input: False |
| Accept wildcard characters: False |

### -Quiet

The Quiet parameter suppresses the 'ShouldProcess' message and the routes will be removed without further question.

| Type: SwitchParameter |
| Parameter Sets: (All) |
| Aliases: Silent |
| Required: False |
| Position: Named |
| Default value: False |
| Accept pipeline input: False |
| Accept wildcard characters: False |

### -Confirm

Prompts you for confirmation before running the cmdlet.

| Type: SwitchParameter |
| Parameter Sets: (All) |
| Aliases: cf |
| Required: False |
| Position: Named |
| Default value: None |
Accept pipeline input: False
Accept wildcard characters: False

-WhatIf
Shows what would happen if the cmdlet runs.
The cmdlet is not run.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi
Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

CommonParameters
This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS
TwinCATSystemService.MqttRoute[]
The Mqtt routes to remove.

OUTPUTS

NOTES

6.32 Reset-IoFreeRun

SYNOPSIS
Resets the IO FreeRun state on the specified target.

SYNTAX

NetIdPortList (Default)
Reset-IoFreeRun [-Quiet] [-Timeout <Int32>] [-WhatIf] [-Confirm] [<CommonParameters>]

NetIdPort
Reset-IoFreeRun -NetId <AmsNetId> [-Quiet] [-Timeout <Int32>] [-WhatIf] [-Confirm] [<CommonParameters>]

AddressStr
Reset-IoFreeRun [-Address] <String> [-Quiet] [-Timeout <Int32>] [-WhatIf] [-Confirm] [<CommonParameters>]
Session
Reset-IoFreeRun -Session <ISession> [-Quiet] [-Timeout <Int32>] [-WhatIf] [-Confirm] [CommonParameters>

DESCRIPTION
Resets the IO FreeRun state on the specified target if its in Config state.
If its not in config state an error will be produced

EXAMPLES
EXAMPLE 1
PS> Reset-IOFreeRun -NetId 5.62.192.46.1.1

PARAMETERS
-NetId
Enter the target system address(es).
Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:
Required: True
Position: Named
Default value: 172.17.60.167.1.1
Accept pipeline input: False
Accept wildcard characters: False

-Address
The address(es) where to get Devices.
This can be the RouteName, NetId, the HostName or the IPAddress.
Wildcards are permitted.
Type: String
Parameter Sets: AddressStr
Aliases:
Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-Session
The Session to use for the Cmdlet, must be connected to port 300, R0_IO
Type: ISession
Parameter Sets: Session
Aliases:
Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
-Quiet

The Quiet parameter suppresses the 'ShouldProcess' message and the Cmdlet will proceed without further question.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Timeout

ADS communication timeout in ms

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

-Confirm

Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.
INPUTS

TwinCAT.ISession
The Session to use for the Cmdlet, must be connected to port 300, R0_IO

OUTPUTS

NOTES

6.33 Restart-AdsComputer

SYNOPSIS

Restarts ("reboots") the operating system on local and remote TwinCAT computers.

SYNTAX

NetIdPort (Default)
[-Force] [-WhatIf] [-Confirm] [<CommonParameters>]

Route
[-Timeout <Int32>] [-Force] [-WhatIf] [-Confirm] [<CommonParameters>]

AddressStr
[-Force] [-WhatIf] [-Confirm] [<CommonParameters>]

Session
[-Force] [-WhatIf] [-Confirm] [<CommonParameters>]

SessionId
[-Force] [-WhatIf] [-Confirm] [<CommonParameters>]

DESCRIPTION

The Restart-AdsComputer cmdlet restarts the operating system on the local and remote TwinCAT computers.

You can use the parameters of Restart-AdsComputer to specify available ADS target systems to restart.

The restart can be done delayed if Users are logged into the target system (existant Session UI) or forced immediately.

You can wait for the restart to complete before you run the next command and specify a waiting time-out.

This feature makes it practical to use Restart-AdsComputer in scripts and functions.
EXAMPLES

EXAMPLE 1
PS> Restart-AdsComputer CX_1111,CX_2222 -force

Restarts the computers CX_1111 and CX_2222 immediately without warning logged in users on the target system and returns immediately without waiting the finished reboot.

The Force parameter suppresses the ShouldProcess query.

EXAMPLE 2
PS> Restart-AdsComputer -netId '1.2.3.4.1.1', '1.2.3.5.1.1' -Delay 30 -Wait -WaitTimeout 240 -force

Restarts the TwinCAT targets ‘1.2.3.4.1.1’ and 1.2.3.5.1.1 without ShouldProcess query after a delay of 30 Seconds (if a user is logged in) and waits for the reboot before continuing.

The Wait timeout is set to 240 Seconds.

PARAMETERS

-Wait
Activates a wait until the rebooted system is available again.

The parameter -WaitTimeout specifies how long the script waits for the reboot.

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-WaitTimeout
The Wait time for the reboot of the target system (default 120 Seconds).

This parameter is used in conjunction with the -Wait parameter.

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 120
Accept pipeline input: False
Accept wildcard characters: False

-Delay
The delay time for the reboot/shutdown of the target system(s) in seconds.

The default is 120 Seconds.

If no user is logged in the target system the reboot/shutdown occurs always immediately without warning.

In case of a log in, a Warning message with countdown will be presented to the user.

Type: Int32
Parameter Sets: (All)
Aliases:
**-NetId**

NetId(s) of the target system.

<table>
<thead>
<tr>
<th>Type:</th>
<th>AmsNetId[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>NetIdPort</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required:</td>
<td>False</td>
</tr>
<tr>
<td>Position:</td>
<td>1</td>
</tr>
<tr>
<td>Default value:</td>
<td>172.17.60.167.1.1</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

**-InputObject**

The ADS routes to shutdown/reboot.

<table>
<thead>
<tr>
<th>Type:</th>
<th>IRoute[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>Route</td>
</tr>
<tr>
<td>Aliases:</td>
<td>Destination, Route</td>
</tr>
<tr>
<td>Required:</td>
<td>True</td>
</tr>
<tr>
<td>Position:</td>
<td>1</td>
</tr>
<tr>
<td>Default value:</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>True (ByValue)</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

**-Address**

Target names/addresses.

These can consist of RouteName, NetID, HostName or IPAddress.

This parameter supports wildcards.

<table>
<thead>
<tr>
<th>Type:</th>
<th>String[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>AddressStr</td>
</tr>
<tr>
<td>Aliases:</td>
<td>Name</td>
</tr>
<tr>
<td>Required:</td>
<td>True</td>
</tr>
<tr>
<td>Position:</td>
<td>1</td>
</tr>
<tr>
<td>Default value:</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>False</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>True</td>
</tr>
</tbody>
</table>

**-Session**

The Session(s) to use for addressing the target systems.

<table>
<thead>
<tr>
<th>Type:</th>
<th>ISession[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>Session</td>
</tr>
<tr>
<td>Aliases:</td>
<td></td>
</tr>
<tr>
<td>Required:</td>
<td>True</td>
</tr>
<tr>
<td>Position:</td>
<td>Named</td>
</tr>
<tr>
<td>Default value:</td>
<td>None</td>
</tr>
<tr>
<td>Accept pipeline input:</td>
<td>True (ByValue)</td>
</tr>
<tr>
<td>Accept wildcard characters:</td>
<td>False</td>
</tr>
</tbody>
</table>

**-SessionId**

Specifies the Sessions (with unique ID) to use for addressing the target systems.
**Timeout**

The ADS timeout in milliseconds for sending the Shutdown/Reboot request.

The Default is 5 Seconds.

**Force**

Suppress the 'ShouldProcess' message and forces Shutdown/Reboot.

**Confirm**

Prompts you for confirmation before running the cmdlet.

**WhatIf**

Shows what would happen if the cmdlet runs.

The cmdlet is not run.
CommonParameters
This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute[]
The ADS routes to shutdown/reboot.

TwinCAT.ISession[]
The Session(s) to use for addressing the target systems.

OUTPUTS

NOTES

6.34  Send-TcReadWrite

SYNOPSIS
Sends a Read/Write access to ADS Server / TwinCAT Devices.

SYNTAX

NetIdPortIndexed (Default)

AddressIndexed

RouteIndexed
Send-TcReadWrite -IndexGroup <UInt32> [-IndexOffset <UInt32>] [-WriteValue <Object>] [-WriteLength <Int32>] [-ReadLength <Int32>] [-ReadType <Type>] -Route <IRoute[]> -Port <Int32> [-Encoding <Encoding>] [-Async] [-Force] [-WhatIf] [-Confirm] [<CommonParameters>]

SessionIndexed
SessionIdIndexed
Send-TcReadWrite -IndexGroup <UInt32> [-IndexOffset <UInt32>] [-WriteValue <Object>] [-WriteLength <Int32>] [-ReadLength <Int32>] [-ReadType <Type>] -SessionId <Int32>[] [-Encoding <Encoding>] [-Async] [-Force] [-WhatIf] [-Confirm] [<CommonParameters>]

DESCRIPTION
This Cmdlet Read/Writes values from/to TwinCAT Devices and works with different ValueProviders.

Because this is a low level data access, only IndexGroup/IndexOffset addressing is available.

IMPORTANT: Sending Read/Write commands should be done with highest care because it could destabilize the TwinCAT System when the write operation is not addressed properly.

To enhance secure operation, the user is enforced to use Length parameters in conjunction with the in/out values which will be checked by the Cmdlet.

The highest attention should also be taken with the IndexGroup/IndexOffset because that represents the Address in the Process Image and cannot be checked by principle.

To prevent that process image overwrites important data by accident please use the -WhatIf and -Confirm parameters whenever it is appropriate and inform about the $ConfirmPreference settings (PS1&gt; get-help about_Preference_Variables) before usage of the Send-TcReadWrite Cmdlet.

EXAMPLES

EXAMPLE 1

> Send-TcReadWrite -NetId 1.2.3.4.5.6 -Port 851 -IndexGroup SymbolValueByName -IndexOffset 0 -WriteValue "TwinCAT_SystemInfoVarList._AppInfo.ProjectName" -ReadLength 1024

ReadWrite access of process image on target '1.2.3.4.5.6:851':
Start ReadWrite operation WriteData: 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'(IG:0xf004,IO:0 x0000,Len:47),
Read: Type 'System.String' (Len:'1024) on target '1.2.3.4.5.6:851'?
[Y] Yes[A] Yes to All[N] No[L] No to All[S] Suspend[?] Help(default is "Y")
y
ADS_DynSymbols

Sends a Read/Write request with index group 0xf004 (SymbolValueByName) and offset 0.
The write data will be initialized with the project symbol path and an returned (read) string (Default encoded) returned.

EXAMPLE 2

> Send-TcReadWrite -NetId 1.2.3.4.5.6 -Port 851 -IndexGroup SymbolValueByName -WriteValue "TwinCAT_SystemInfoVarList._AppInfo.ProjectName" -ReadLength 64 | format-hex

ReadWrite access of process image on target '1.2.3.4.5.6:851':
Start ReadWrite operation WriteData: 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'(IG:0xf004,IO:0 x0000,Len:47),
Read: Type 'System.Byte[]' (Len:'64) on target '1.2.3.4.5.6:851'?
[Y] Yes[A] Yes to All[N] No[L] No to All[S] Suspend[?] Help(default is "Y")
y
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 41 44 53 5F 44 79 6E 53 79 6D 62 6F 6C 73 00 00 ADS_DynSymbols..
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...............
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...............
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...............

Sends a Read/Write request with index group 0xf004 (SymbolValueByName) and offset 0.
The write data will be initialized with the project symbol path and the returned (read) data is by default a byte array of 64 bytes.
The result value will be formatted as hex code.

**EXAMPLE 3**

```powershell
PS> $route = Get-AdsRoute -Name 'CX-123456'
PS> $session = $route | New-TcSession -Port 851
PS> $handle = $session | Send-TcReadWrite -IndexGroup SymbolHandleByName -WriteValue "GVL.vgInt" -ReadType Int32 -force
PS> $session | Read-TcValue -IndexGroup SymbolValueByHandle -IndexOffset $handle -ValueType Int16
42
```

Create a session to the PLC (Port 851) of a target system, determine the SymbolHandle by InstancePath and use this handle to read its 'Int16' Value (INT on PLC System).

**PARAMETERS**

- **IndexGroup**

  IndexGroup of the Value to ReadWrite, only for IndexGroup/IndexOffset access.

  **IMPORTANT:** Please be aware, that writing data via IndexGroup/IndexOffset can overwrite data in the ProcessImage and possibly destabilizes the system.

  No validity check is done for the symbol alignment and therefore this should be done with highest care!

  Type: UInt32  
  Parameter Sets: (All)  
  Aliases: IG

  Required: True  
  Position: Named  
  Default value: 0  
  Accept pipeline input: False  
  Accept wildcard characters: False

- **IndexOffset**

  IndexOffset of the Value to write, only for IndexGroup/IndexOffset access.

  **IMPORTANT:** Please be aware, that writing data via IndexGroup/IndexOffset simply overwrites data in the ProcessImage and can destabilize the system.

  No validity check is done for the symbol alignment and therefore this should be done with highest care!

  If applicable writing data via symbolic information should be preferred!

  Type: UInt32  
  Parameter Sets: (All)  
  Aliases: IO

  Required: False  
  Position: Named  
  Default value: 0  
  Accept pipeline input: False  
  Accept wildcard characters: False

- **WriteValue**

  The value to write.

  If no additional Length parameter is set, the Write-TcValue Cmdlet marshals this value to its appropriate size.

  To not overwrite data of other symbols within the process image, special attention must be taken (see the Confirm and Whatif parameters).

  Type: Object  
  Parameter Sets: (All)  
  Aliases:
Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

**-WriteLength**

The Length of the data that will be overwritten within the process image.

By default the marshal size of the object used in the -WriteValue parameter is taken.

This parameter is used to override the marshal size and can be helpful to secure the write operation - to not overwrite more data then expected.

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

**-ReadLength**

The Length of the data that will be read from the process image.

By default, when not specifying this parameter the marshalling size of the -ReadType parameter will be taken.

This -ReadLength parameter is only helpful when the marshalling size cannot be determined from the read type (e.g. `byte[]`)?

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

**-ReadType**

Use the ReadType parameter to specify the Read/Return type of the data.

If not used, this cmdlet returns the raw `byte[]` as result.

Type: Type
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: `System.Byte[]`
Accept pipeline input: False
Accept wildcard characters: False

**-NetId**

The ADS target NetID(s) of the system(s) where to read/write the Value.

More than one target will be supported.
When not specified, this argument defaults to AmsNetId.Local.

**Type:** AmsNetId[]
**Parameter Sets:** NetIdPortIndexed
**Aliases:**

- **Required:** False
- **Position:** Named
- **Default value:** 172.17.60.167.1.1
- **Accept pipeline input:** False
- **Accept wildcard characters:** False

---

**-Route**

The target system (as Route) where to read/write the value.

**Type:** IRoute[]
**Parameter Sets:** RouteIndexed
**Aliases:** Destination

- **Required:** True
- **Position:** Named
- **Default value:** None
- **Accept pipeline input:** False
- **Accept wildcard characters:** False

---

**-Address**

The target address where to read/write the Value.

The Address can consist of RouteName, NetId, HostName or IPAddress.

Wildcards are permitted.

**Type:** String[]
**Parameter Sets:** AddressIndexed
**Aliases:**

- **Required:** True
- **Position:** Named
- **Default value:** None
- **Accept pipeline input:** False
- **Accept wildcard characters:** True

---

**-Session**

The session object represents the target session where to read/write the value.

**Type:** ISession[]
**Parameter Sets:** SessionIndexed
**Aliases:**

- **Required:** True
- **Position:** Named
- **Default value:** None
- **Accept pipeline input:** True (ByPropertyName, ByValue)
- **Accept wildcard characters:** False

---

**-SessionId**

The session ID represents the target session where to read/write the value.

**Type:** Int32[]
**Parameter Sets:** SessionIdIndexed
**Aliases:**

- **Required:** True
- **Position:** Named
- **Default value:** None
- **Accept pipeline input:** False
- **Accept wildcard characters:** False
-Port

The Port, where to read/write the value.

This Parameter is used in combination with the NetId, Route or Address input parameter.

Type: Int32
Parameter Sets: NetIdPortIndexed, AddressIndexed, RouteIndexed
Aliases:

Required: True
Position: Named
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False

-Encoding

Specifies the Encoding for strings.

The Default is ANSI with actual code page.

Type: Encoding
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: System.Text.SBCSCodePageEncoding
Accept pipeline input: False
Accept wildcard characters: False

-Async

Starts the write on different threads.

Only for internal use and test purposes.

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: True
Accept pipeline input: False
Accept wildcard characters: False

-Force

Suppress the 'ShouldProcess' message and forces the write.

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Confirm

Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -
InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -
WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession[]

The session object represents the target session where to read/write the value.

OUTPUTS

NOTES

6.35 Set-AdsState

SYNOPSIS

Set the ADS State of a TwinCAT Target.

SYNTAX

NetIdPort (Default)
Quiet] [-Force] [-Timeout <Int32>] [-WhatIf] [-Confirm] [CommonParameters>]

AddressStr
Force] [-Timeout <Int32>] [-WhatIf] [-Confirm] [CommonParameters>]

Route
Timeout <Int32>] [-WhatIf] [-Confirm] [CommonParameters>]

Default value: None
Accept pipeline input: False
Accept wildcard characters: False
**Session**


**SessionId**


**DESCRIPTION**

This Cmdlet sets the ADS State of the specified TwinCAT Targets (E.g. Start / Stop / Config / Reconfig)

**EXAMPLES**

**EXAMPLE 1**

PS > Set-AdsState Config -Reinitialize

**EXAMPLE 2**

PS > Set-AdsState Run 1.2.3.4,CX_0130C7

**EXAMPLE 3**

PS > get-AdsRoute | set-AdsState Config

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
<th>OK</th>
<th>Time (ms)</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>work-nb2</td>
<td>Config</td>
<td>True</td>
<td>0</td>
<td>172.17.242.54.1.1</td>
</tr>
<tr>
<td>CX_0130C7</td>
<td>Config</td>
<td>True</td>
<td>0</td>
<td>5.1.48.199.1.1</td>
</tr>
</tbody>
</table>

**PARAMETERS**

**-State**

The state/value to set.

Possible values: Invalid, Idle, Reset, Init, Start, Run, Stop, SaveConfig, LoadConfig, PowerFailure, PowerGood, Error, Shutdown, Suspend, Resume, Config, Reconfig, Stopping, Incompatible, Exception

Type: AdsState
Parameter Sets: (All)
Aliases:
  Accepted values: Invalid, Idle, Reset, Init, Start, Run, Stop, SaveConfig, LoadConfig, PowerFailure, PowerGood, Error, Shutdown, Suspend, Resume, Config, Reconfig, Stopping, Incompatible, Exception

Required: True
Position: 0
Default value: Invalid
Accept pipeline input: False
Accept wildcard characters: False

**-NetId**

The NetId address where to set the state (Local system by default)

Type: AmsNetId[]
Parameter Sets: NetIdPort
Aliases:
Required: False
Position: 1  
Default value: 172.17.60.167.1.1  
Accept pipeline input: False  
Accept wildcard characters: False

### -Port

The AmsPort where to set the state (Port 1000, SystemService by default)

Type: Int32  
Parameter Sets: NetIdPort, AddressStr  
Aliases:

<table>
<thead>
<tr>
<th>Required</th>
<th>Position</th>
<th>Default value</th>
<th>Accept pipeline input</th>
<th>Accept wildcard characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>2</td>
<td>10000</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

### -InputObject

Target route(s), where to set the state.

Type: IRoute[]  
Parameter Sets: Route  
Aliases: Destination, Route

<table>
<thead>
<tr>
<th>Required</th>
<th>Position</th>
<th>Default value</th>
<th>Accept pipeline input</th>
<th>Accept wildcard characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>1</td>
<td>None</td>
<td>True (ByValue)</td>
<td>False</td>
</tr>
</tbody>
</table>

### -Address

The address of the system where to set the state.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

Type: String[]  
Parameter Sets: AddressStr  
Aliases: Name

<table>
<thead>
<tr>
<th>Required</th>
<th>Position</th>
<th>Default value</th>
<th>Accept pipeline input</th>
<th>Accept wildcard characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>1</td>
<td>None</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>

### -Session

The Session to use for the Cmdlet.

Type: ISession[]  
Parameter Sets: Session  
Aliases:

<table>
<thead>
<tr>
<th>Required</th>
<th>Position</th>
<th>Default value</th>
<th>Accept pipeline input</th>
<th>Accept wildcard characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>Named</td>
<td>None</td>
<td>True (ByValue)</td>
<td>False</td>
</tr>
</tbody>
</table>

### -SessionId

Specifies the Session (with unique ID) to use for the Cmdlet.
### -Reinitialize

Reinitializes the target system before it is set to the target state (goes over STOP)

<table>
<thead>
<tr>
<th>Type</th>
<th>SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases</td>
<td></td>
</tr>
</tbody>
</table>

- Required: False
- Position: Named
- Default value: False
- Accept pipeline input: False
- Accept wildcard characters: False

### -Quiet

Sets the Quiet mode of the command.

The Cmdlet then returns a $true or $false but not the actual states of the targets. The return value will be $true if at least one SetState operations succeed and it will be $false if have failed.

<table>
<thead>
<tr>
<th>Type</th>
<th>SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases</td>
<td></td>
</tr>
</tbody>
</table>

- Required: False
- Position: Named
- Default value: False
- Accept pipeline input: False
- Accept wildcard characters: False

### -Force

Forces the command (no questions asked, ResetFailFastHandler)

<table>
<thead>
<tr>
<th>Type</th>
<th>SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases</td>
<td></td>
</tr>
</tbody>
</table>

- Required: False
- Position: Named
- Default value: False
- Accept pipeline input: False
- Accept wildcard characters: False

### -Timeout

Communication timeout in ms.

<table>
<thead>
<tr>
<th>Type</th>
<th>Int32[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets</td>
<td>SessionId</td>
</tr>
<tr>
<td>Aliases</td>
<td></td>
</tr>
</tbody>
</table>

- Required: True
- Position: Named
- Default value: None
- Accept pipeline input: False
- Accept wildcard characters: False
-Confirm

Prompts you for confirmation before running the cmdlet.

<table>
<thead>
<tr>
<th>Type:</th>
<th>SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases:</td>
<td>cf</td>
</tr>
</tbody>
</table>

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

<table>
<thead>
<tr>
<th>Type:</th>
<th>SwitchParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Sets:</td>
<td>(All)</td>
</tr>
<tr>
<td>Aliases:</td>
<td>wi</td>
</tr>
</tbody>
</table>

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute[]
Target route(s), where to set the state.

TwinCAT.ISession[]
The Session to use for the Cmdlet.

OUTPUTS

NOTES

6.36 Set-AmsRouterEndpoint

SYNOPSIS

Sets the AmsConfiguration (Loopback address and port, RouterEndpoint).

SYNTAX

Set-AmsRouterEndpoint [-IP <IPAddress>] [-Port <Int32>] [-Force] [-WhatIf] [-Confirm] [<CommonParameters>]
DESCRIPTION

This Cmdlet sets the AmsConfiguration setting of the current running process. By default the AdsClients and AdsServers are connected to the TwinCAT Router. The involved communication uses the local Loopback address with port 0xBF02 by default. To enable virtualization scenarios, where AdsClient / AdsServer / Router applications run in their own (virtual) environment, this address has to be changed for the actual running process.

1. Setting of the following Environment Variables before this Powershell Module (TcXaeMgmt) is loaded:
   $env:AmsConfiguration:LoopbackAddress = "168.0.1.1" $env:AmsConfiguration:LoopbackPort = "1234"
   Both Variables are optional

2. Set the AmsConfiguration with the 'Set-AmsRouterEndpoint' Cmdlet.

When processing this Cmdlet, all prexisting open AdsSessions will be invalid. The Default RouterEndpoint is IPAddress:127.0.0.1 and Port 0xBF02.

EXAMPLES

EXAMPLE 1

PS > Set-AmsRouterEndpoint -IP '168.0.1.1' -Port 1234

PARAMETERS

-IP

The state/value to set.

Type: IPAddress
Parameter Sets: (All)
Aliases:

Required: False
Position: 0
Default value: 127.0.0.1
Accept pipeline input: False
Accept wildcard characters: False

-Port

The state/value to set.

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: 1
Default value: 48898
Accept pipeline input: False
Accept wildcard characters: False

-Force

The state/value to set.

Type: SwitchParameter
Parameter Sets: (All)
Aliases:
Required: False  
Position: Named  
Default value: False  
Accept pipeline input: False  
Accept wildcard characters: False

-Confirm

Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter  
Parameter Sets: (All)  
Aliases: cf

Required: False  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

Type: SwitchParameter  
Parameter Sets: (All)  
Aliases: wi

Required: False  
Position: Named  
Default value: None  
Accept pipeline input: False  
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

OUTPUTS

NOTES

6.37 Set-IoFreeRun

SYNOPSIS

Sets the IO FreeRun state of the target.

SYNTAX

NetIdPortList (Default)

Set-IoFreeRun [-Timeout <Int32>] [-Quiet] [-WhatIf] [-Confirm] [CommonParameters]
**NetIdPort**

```
Set-IoFreeRun [-NetId <AmsNetId>] [-Timeout <Int32>] [-Quiet] [-WhatIf] [-Confirm] [<CommonParameters>]
```

**AddressStr**

```
Set-IoFreeRun [-Address] <String> [-Timeout <Int32>] [-Quiet] [-WhatIf] [-Confirm] [<CommonParameters>]
```

**Session**

```
Set-IoFreeRun -Session <ISession> [-Timeout <Int32>] [-Quiet] [-WhatIf] [-Confirm] [<CommonParameters>]
```

**DESCRIPTION**

Sets the IO FreeRun state of a TwinCAT target.

The target must be in Config mode, otherwise an error will be produced.

**EXAMPLES**

**EXAMPLE 1**

```
PS> Set-IoFreeRun -NetId 5.62.192.46.1.1
```

**PARAMETERS**

- **-NetId**

  Enter the target system address(es).

  Type: AmsNetId
  Parameter Sets: NetIdPort
  Aliases:
  
  Required: False
  Position: Named
  Default value: 172.17.60.167.1.1
  Accept pipeline input: False
  Accept wildcard characters: False

- **-Address**

  The address(es) where to get Devices.

  This can be the RouteName, NetId, the HostName or the IPAddress.

  Wildcards are permitted.

  Type: String
  Parameter Sets: AddressStr
  Aliases:
  
  Required: True
  Position: 1
  Default value: None
  Accept pipeline input: False
  Accept wildcard characters: True

- **-Session**

  The Session to use for the Cmdlet
Type: ISession
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-**Timeout**

ADS Communication Timeout in ms

Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

-**Quiet**

The Quiet parameter suppresses the 'ShouldProcess' message and the Cmdlet will proceed without further question.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-**Confirm**

Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-**WhatIf**

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
CommonParameters
This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS
TwinCAT.ISession
The Session to use for the Cmdlet

OUTPUTS

NOTES

6.38 Stop-AdsComputer

SYNOPSIS
Stops (shuts down) local and remote TwinCAT computers.

SYNTAX
NetIdPort (Default)

Route

AddressStr

Session

SessionId
Stop-AdsComputer [-Delay <Int32>] -SessionId <Int32[]> [-Timeout <Int32>] [-Force] [-WhatIf] [-Confirm] [<CommonParameters>]

DESCRIPTION
The Stop-AdsComputer cmdlet shuts the operating system on the local and remote TwinCAT computers down.
You can use the parameters of Stop-AdsComputer to specify available ADS target systems to shutdown.
The shutdown can be done delayed if Users are logged into the target system (existant Session UI) or forced immediately.

**EXAMPLES**

**EXAMPLE 1**

PS> Stop-AdsComputer CX_1111,CX_2222 -force

Immediate shutdown of the computers CX_1111 and CX_2222.

The Force parameter supresses the ShouldProcess query.

**EXAMPLE 2**

PS> Stop-AdsComputer -netId '1.2.3.4.1.1', '1.2.3.5.1.1' -Delay 30 -force

Stops the TwinCAT targets '1.2.3.4.1.1' and 1.2.3.5.1.1 without ShouldProcess query after a delay of 30 Seconds (if a user is logged in).

**PARAMETERS**

- **-Delay**
  
The delay time for the reboot/shutdown of the target system(s) in seconds.
  
The default is 120 Seconds.
  
If no user is logged in the target system the reboot/shutdown occurs always immediatly without warning.
  
In case of a log in, a Warning message with countdown will be presented to the user.

  Type: Int32
  Parameter Sets: (All)
  Aliases:

  - Required: False
  - Position: Named
  - Default value: 0
  - Accept pipeline input: False
  - Accept wildcard characters: False

- **-NetId**

  NetId(s) of the target system.

  Type: AmsNetId[]
  Parameter Sets: NetIdPort
  Aliases:

  - Required: False
  - Position: 1
  - Default value: 172.17.60.167.1.1
  - Accept pipeline input: False
  - Accept wildcard characters: False

- **-InputObject**

  The ADS routes to shutdown/reboot.

  Type: IRoute[]
  Parameter Sets: Route
  Aliases: Destination, Route

  - Required: True
  - Position: 1
  - Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-**Address**

Target names/addresses.

These can consist of RouteName, NetID, HostName or IPAddress.

This parameter supports wildcards.

Type: `String[]`
Parameter Sets: `AddressStr`
Aliases: `Name`

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-**Session**

The Session(s) to use for addressing the target systems.

Type: `ISession[]`
Parameter Sets: `Session`
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-**SessionId**

Specifies the Sessions (with unique ID) to use for addressing the target systems.

Type: `Int32[]`
Parameter Sets: `SessionId`
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-**Timeout**

The ADS timeout in milliseconds for sending the Shutdown/Reboot request.

The Default is 5 Seconds.

Type: `Int32`
Parameter Sets: `(All)`
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False

-**Force**

Suppress the 'ShouldProcess' message and forces Shutdown/Reboot.
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Confirm

Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute[]

The ADS routes to shutdown/reboot.

TwinCAT.ISession[]

The Session(s) to use for addressing the target systems.
6.39 Test-AdsRoute

SYNOPSIS
Test the specified route connection.

SYNTAX

AddressStr (Default)
[<CommonParameters>]

NetId
Test-AdsRoute [-NetId] <AmsNetId[]> [[-Port] <Int32[]>] [[-SourceRoute <RouteTarget>] [-Mode <PingStrategy>]]
[<CommonParameters>]

Route
Test-AdsRoute [[-Port] <Int32[]>] [[-SourceRoute <RouteTarget>] [[-InputObject] <RouteTargetCollection>]
[-OnlinePorts] [-Quiet] [<CommonParameters>]

DESCRIPTION
This Cmdlet establishes a connection to the specified target system and tests if the connection is working.
A Port scan can be executed.

EXAMPLES

EXAMPLE 1
PS > Test-AdsRoute -Port 851

PARAMETERS

-Name
The name(s) or address(es) of the systems to test.
These can consist of RouteName, NetID, HostName or IPAddress.
Wildcards are permitted.

Type: String[]
Parameter Sets: AddressStr
Aliases: Address
Required: False
Position: 1
Default value: None
-NetId

The NetId(s) of the target system to test (AmsNetId.Local by default)

Type: AmsNetId[]
Parameter Sets: NetId
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-Port

The Port(s) of the target system to test.

Type: Int32[]
Parameter Sets: (All)
Aliases:

Required: False
Position: 2
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-SourceRoute

The source system where to test the Route.

Type: RouteTarget
Parameter Sets: (All)
Aliases: Source

Required: False
Position: Named
Default value: Local
Accept pipeline input: False
Accept wildcard characters: False

-InputObject

The routes targets to test with this Cmdlet.

Type: RouteTargetCollection
Parameter Sets: Route
Aliases: Destination

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-Mode

The Ping Strategy (PingStrategy.Ads by default)

Possible values: None, IP, HostName, IPOrHostName, Ads, AdsGetState, AdsTestConnection, Default

Type: PingStrategy
Parameter Sets: (All)
Aliases:

Accepted values: None, IP, HostName, IPOrHostName, Ads, AdsGetState, AdsTestConnection, Default
**-TimeoutSeconds**

Sets the delay value for each single request test.

The test fails if a (single) response isn't received before the timeout expires.

The default is 2 seconds.

Type: Int32
Parameter Sets: (All)
Aliases: TTL, TimeToLive,

**-Count**

Specifies the number of echo requests to send.

The default value is 1.

Type: Int32
Parameter Sets: (All)

**-Delay**

Specifies the interval between pings, in seconds.

The default value is 1.

Type: Int32
Parameter Sets: (All)

**-DefaultPorts**

Tests all default ports.

The following ports will be tested: 10000, 300, 301, 302, 303, 305, 801, 811, 821, 831, 851, 852, 853, 854, 855, 19200 The 'DefaultPorts' switch overrides the 'Port' parameter.
Accept pipeline input: False
Accept wildcard characters: False

-OnlinePorts

Determines all active/online ports from the target and tests them.

The 'OnlinePorts' switch overrides the 'DefaultPorts' and 'Port' parameters.

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Quiet

The Quiet mode.

Returns a boolean only ($true, if one ping succeeded and $false if all failed)

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.Ads.AmsNetId[]
The NetId(s) of the target system to test (AmsNetId.Local by default)

TwinCAT.RouteTargetCollection
The routes targets to test with this Cmdlet.

OUTPUTS

NOTES

6.40 Unregister-AdsHandle

SYNOPSIS

Unregisters a symbol handle.
### SYNTAX

**NetIdPortHandle (Default)**

```powershell
Unregister-AdsHandle [-NetId <AmsNetId>] -Port <Int32> [-Handle] <UInt32[]> [CommonParameters]
```

**NetIdPortInfo**

```powershell
Unregister-AdsHandle [-NetId <AmsNetId>] -Port <Int32> [-InputObject] <AdsHandleInfo[]> [CommonParameters]
```

**RouteHandle**

```powershell
Unregister-AdsHandle -Route <IRoute> -Port <Int32> [-Handle] <UInt32[]> [CommonParameters]
```

**RouteInfo**

```powershell
Unregister-AdsHandle -Route <IRoute> -Port <Int32> [-InputObject] <AdsHandleInfo[]> [CommonParameters]
```

**AddressHandle**

```powershell
Unregister-AdsHandle -Address <String> -Port <Int32> [-Handle] <UInt32[]> [CommonParameters]
```

**AddressInfo**

```powershell
Unregister-AdsHandle -Address <String> -Port <Int32> [-InputObject] <AdsHandleInfo[]> [CommonParameters]
```

**SessionHandle**

```powershell
Unregister-AdsHandle -Session <ISession> [-Handle] <UInt32[]> [CommonParameters]
```

**SessionInfo**

```powershell
Unregister-AdsHandle -Session <ISession> [-InputObject] <AdsHandleInfo[]> [CommonParameters]
```

**SessionIdHandle**

```powershell
Unregister-AdsHandle -SessionId <Int32> [-Handle] <UInt32[]> [CommonParameters]
```

**SessionIdInfo**

```powershell
Unregister-AdsHandle -SessionId <Int32> [-InputObject] <AdsHandleInfo[]> [CommonParameters]
```

### DESCRIPTION

This Cmdlet unregisters an already registered symbol handle from the target system.

The Cmdlet supports raw \[uint\] handles or AdsHandleInfo objects.

### EXAMPLES

**EXAMPLE 1**

```powershell
PS> $session = New-TcSession -NetId '1.2.3.4.5.6' -Port 851
PS> $handleInfo = $session | get-AdsHandle -Path 'TwinCAT_SystemInfoVarList.AppInfo.ProjectName'
PS> $handleInfo = register-AdsHandle -Path 'TwinCAT_SystemInfoVarList.AppInfo.ProjectName' -Session $s
PS> $handleInfo
```
<table>
<thead>
<tr>
<th>InstancePath</th>
<th>Result Handle</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT_SystemInfoVarList._AppInfo.ProjectName</td>
<td>NoError 0x428000FC (1115685116)</td>
</tr>
</tbody>
</table>

```powershell
PS> Read-TcValue -Session $session -IndexGroup SymbolValueByHandle -IndexOffset $handleInfo.Handle -Type String
MyProject
PS> $handle | Unregister-AdsHandle -Session $session
PS> $session | Close-tcsession
```

Opens a new device session, registers a Symbol Handle to the ProjectName of the running PLC Project, Reads the value by handle unregisters the handle and closes the session again.

**PARAMETERS**

- `-NetId`
  The NetId part of the device target address.
  
  Type: AmsNetId  
  Parameter Sets: NetIdPortHandle, NetIdPortInfo  
  Aliases:
  
<table>
<thead>
<tr>
<th>Required</th>
<th>Position</th>
<th>Default Value</th>
<th>Accept Pipeline Input</th>
<th>Accept Wildcard Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>Named</td>
<td>172.17.60.167.1.1</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

- `-Route`
  Specifies the target system.
  
  Type: IRoute  
  Parameter Sets: RouteHandle, RouteInfo  
  Aliases: Destination
  
<table>
<thead>
<tr>
<th>Required</th>
<th>Position</th>
<th>Default Value</th>
<th>Accept Pipeline Input</th>
<th>Accept Wildcard Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>Named</td>
<td>None</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

- `-Address`
  The target address of the system.
  The Address can consist of RouteName, NetId, IPAddress or HostName.
  Wildcards are permitted and ArgumentCompleter is supported.
  
  Type: String  
  Parameter Sets: AddressHandle, AddressInfo  
  Aliases:
  
<table>
<thead>
<tr>
<th>Required</th>
<th>Position</th>
<th>Default Value</th>
<th>Accept Pipeline Input</th>
<th>Accept Wildcard Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>Named</td>
<td>None</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>

- `-Session`
  The Session object (instead of specifieing the target system address).
  
  Type: ISession  
  Parameter Sets: SessionHandle, SessionInfo  
  Aliases:
### -SessionId

Specifies the Session (with unique ID) to use (instead of specifying the address).

**Type:** Int32  
**Parameter Sets:** SessionIdHandle, SessionIdInfo  
**Aliases:**

<table>
<thead>
<tr>
<th>Required</th>
<th>Position</th>
<th>Default value</th>
<th>Accept pipeline input</th>
<th>Accept wildcard characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>Named</td>
<td>-1</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

### -Port

The address Port to use.

ClearText names for the Port and ArgumentCompleter are supported.

**Type:** Int32  
**Parameter Sets:** NetIdPortHandle, NetIdPortInfo, RouteHandle, RouteInfo, AddressHandle, AddressInfo  
**Aliases:**

<table>
<thead>
<tr>
<th>Required</th>
<th>Position</th>
<th>Default value</th>
<th>Accept pipeline input</th>
<th>Accept wildcard characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>Named</td>
<td>10000</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

### -Handle

The instance path of the symbol to read (symbolic access).

This parameter supports wildcards.

**Type:** UInt32[]  
**Parameter Sets:** NetIdPortHandle, RouteHandle, AddressHandle, SessionHandle, SessionIdHandle  
**Aliases:**

<table>
<thead>
<tr>
<th>Required</th>
<th>Position</th>
<th>Default value</th>
<th>Accept pipeline input</th>
<th>Accept wildcard characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>1</td>
<td>None</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>

### -InputObject

The AdsHandleInfo object (produced by Register-AdsHandle Cmdlet)

**Type:** AdsHandleInfo[]  
**Parameter Sets:** NetIdPortInfo, RouteInfo, AddressInfo, SessionInfo, SessionIdInfo  
**Aliases:** HandleInfo

<table>
<thead>
<tr>
<th>Required</th>
<th>Position</th>
<th>Default value</th>
<th>Accept pipeline input</th>
<th>Accept wildcard characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>1</td>
<td>None</td>
<td>None</td>
<td>False</td>
</tr>
</tbody>
</table>
CommonParameters
This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession
The Session object (instead of specifying the target system address).

TwinCAT.Management.Automation.AdsHandleInfo
The AdsHandleInfo object (produced by Register-AdsHandle Cmdlet)

OUTPUTS

NOTES

6.41 Write-TcValue

SYNOPSIS
Write values to TwinCAT devices.

SYNTAX

NetIdPortSymbol (Default)

NetIdPortIndexed

RouteIndexed

RouteSymbol
Write-TcValue -Route <IRoute[]> -Port <Int32> [-Value <Object>] [-Path] <String> [-Encoding <Encoding>] [-Force] [-WhatIf] [-Confirm] [<CommonParameters>]

AddressIndexed
**DESCRIPTION**

This Cmdlet writes values to TwinCAT Devices.

The devices can be accessed via different ValueProviders.

All sorts of ADS-addressing will be supported by this Cmdlet: Addressing by IndexGroup / IndexOffset (see IndexGroup, IndexOffset parameters) Addressing by Instance Path (see path parameter) Addressing by Symbol (see InputObject / Symbol parameter)

**IMPORTANT:** Writing values should be done with highest care because it could destabilize the TwinCAT System when the write operation is not addressed properly.

While writing with available symbol information is not critical and should be preferred the size and position of symbol data is known within the process image), the access via Instance path is less secure.

The size of the overwritten data is not known and therefore not checked by the Cmdlet.

The highest attention should be taken with write IndexGroup / IndexOffset write operations because beneath the unknown data size even the position of the data is not checked.

The data is written directly into the process image.

**EXAMPLES**

**EXAMPLE 1**

PS> $session = New-TcSession -NetId 1.2.3.4.5.6 -Port 851
PS> $projectNameSymbol = $session | Get-TcSymbol -path "*ProjectName"
<table>
<thead>
<tr>
<th>InstanceName</th>
<th>DataType</th>
<th>Size</th>
<th>InstancePath</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProjectName</td>
<td>STRING(63)</td>
<td>64</td>
<td>TwinCAT_SystemInfoVarList._AppInfo.ProjectName</td>
</tr>
</tbody>
</table>

PS> $projectNameSymbol | Read-TcValue

OldProjectName

PS> $projectNameSymbol | Write-TcValue -Value "NewProjectName" -force
PS> $projectNameSymbol | ReadTcValue

NewProjectName

This example shows how to create a session, determining the Symbol 'ProjectName within the _AppInfo Struct on a running PLC project and reading its value.

After that, the Value will be overwritten with 'NewProjectName'.

**EXAMPLE 2**

Write-TcValue -session $session -IndexGroup 0x4040 -IndexOffset 0x1247A8 -Value "NewProjectName"

Writes a string typed Value to the specified IndexGroup/IndexOffset Address.

**PARAMETERS**

- **-NetId**

  The ADS target NetID(s) of the system(s) where to write the Value.

  More than one target will be supported.

  Type: AmsNetId[]
  Parameter Sets: NetIdPortSymbol, NetIdPortIndexed
  Aliases:
  Required: False
  Position: Named
  Default value: 172.17.60.167.1.1
  Accept pipeline input: False
  Accept wildcard characters: False

- **-Route**

  The target system (as Route) where to write the value.

  Type: IRoute[]
  Parameter Sets: RouteIndexed, RouteSymbol
  Aliases: Destination

  Required: True
  Position: Named
  Default value: None
  Accept pipeline input: False
  Accept wildcard characters: False

- **-Address**

  The target address where to write the Value.

  The Address can consist of RouteName, NetId, HostName or IPAddress.

  Wildcards are permitted.

  Type: String[]
  Parameter Sets: AddressIndexed, AddressSymbol
  Aliases:

  Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-Session

The session object represents the target session where to write the value.

Type: ISession[]
Parameter Sets: SessionIndexed, SessionSymbol
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName, ByValue)
Accept wildcard characters: False

-SessionId

The session ID represents the target session where to write the value.

Type: Int32[
Parameter Sets: SessionIdIndexed, SessionIdSymbol
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

-Port

The Port, where to write the value.

This Parameter is used in combination with the NetId, Route or Address input parameter.

Type: Int32
Parameter Sets: NetIdPortSymbol, NetIdPortIndexed, RouteIndexed, RouteSymbol, AddressIndexed, AddressSymbol
Aliases:

Required: True
Position: Named
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False

-Value

The value to write.

If no additional Length parameter is set, the Write-TcValue Cmdlet marshalls this value to its appropriate size.

To not overwrite data of other symbols within the process image, special attention must be taken (see the Confirm and Whatif parameters).

Type: Object
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
-IndexGroup

IndexGroup of the Value to write, only for IndexGroup/IndexOffset access.

IMPORTANT: Please be aware, that writing data via IndexGroup/IndexOffset simply overwrites data in the ProcessImage and can destabilize the system.

No validity check is done for the symbol alignment and therefore this should be done with highest care!

If applicable writing data via symbolic information should be preferred!

<table>
<thead>
<tr>
<th>Type:</th>
<th>UInt32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>NetIdPortIndexed, RouteIndexed, AddressIndexed, SessionIndexed, SessionIdIndexed</td>
</tr>
<tr>
<td>Aliases:</td>
<td>IG</td>
</tr>
</tbody>
</table>

Required: True

Position: 1

Default value: 0

Accept pipeline input: False

Accept wildcard characters: False

-IndexOffset

IndexOffset of the Value to write, only for IndexGroup/IndexOffset access.

IMPORTANT: Please be aware, that writing data via IndexGroup/IndexOffset simply overwrites data in the ProcessImage and can destabilize the system.

No validity check is done for the symbol alignment and therefore this should be done with highest care!

If applicable writing data via symbolic information should be preferred!

<table>
<thead>
<tr>
<th>Type:</th>
<th>UInt32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>NetIdPortIndexed, RouteIndexed, AddressIndexed, SessionIndexed, SessionIdIndexed</td>
</tr>
<tr>
<td>Aliases:</td>
<td>IO</td>
</tr>
</tbody>
</table>

Required: False

Position: 2

Default value: 0

Accept pipeline input: False

Accept wildcard characters: False

-Size

The Length of the data that will be overwritten within the process image.

IMPORTANT: Please be aware, that writing data via IndexGroup/IndexOffset simply overwrites data in the ProcessImage and could destabilize the system.

No further validity check is done for the symbol alignment and therefore this should be done with highest care (best with use of the -Confirm and -Whatif Cmdlet arguments).

If applicable writing data via symbolic information should be preferred!

<table>
<thead>
<tr>
<th>Type:</th>
<th>Int32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>NetIdPortIndexed, RouteIndexed, AddressIndexed, SessionIndexed, SessionIdIndexed</td>
</tr>
<tr>
<td>Aliases:</td>
<td>Length, WriteSize</td>
</tr>
</tbody>
</table>

Required: False

Position: Named

Default value: -1

Accept pipeline input: False

Accept wildcard characters: False

-Path

The instance path to the symbol to write (Symbolic access).

Wildcards are permitted.
Type: String
Parameter Sets: NetIdPortSymbol, RouteSymbol, AddressSymbol, SessionSymbol, SessionIdSymbol
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True

-InputObject

The symbol object on which to write the value.

Type: ISymbol
Parameter Sets: InputObject
Aliases: Symbol

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False

-Encoding

Specifies the Encoding for strings.

The Default is ANSI with actual code page.

Type: Encoding
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: System.Text.SBCSCodePageEncoding
Accept pipeline input: False
Accept wildcard characters: False

-Force

Suppress the 'ShouldProcess' message and forces the write.

Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False

-Confirm

Prompts you for confirmation before running the cmdlet.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi
Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession[]

The session object represents the target session where to write the value.

TwinCAT.TypeSystem.ISymbol

The symbol object on which to write the value.

OUTPUTS

NOTES
More Information:
www.beckhoff.de/te1000