# 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 **Installation** ..................................................................................................................................... 9  
   3.1 **System requirements** ............................................................................................................... 9  
   3.2 **Installation** ................................................................................................................................ 9  
   3.3 **Licensing** ................................................................................................................................... 9  

4 **Technical introduction** .................................................................................................................. 12  
   4.1 **Available languages** .............................................................................................................. 12  
      4.1.1 **Setting Microsoft SAPI** ................................................................................................... 15  
      4.1.2 **Setting Amazon Polly** .................................................................................................... 16  

5 **Quick start: speech input (ASR)** .................................................................................................... 17  
   5.1 **Creating a configuration** ......................................................................................................... 17  
   5.2 **Configuring speech input** ....................................................................................................... 18  
   5.3 **Programming the PLC** .......................................................................................................... 24  

6 **Quick start: speech output (TTS)** ................................................................................................. 27  
   6.1 **Creating a configuration** ......................................................................................................... 27  
   6.2 **Configuring speech output** ..................................................................................................... 28  
   6.3 **Programming the PLC** .......................................................................................................... 35  

7 **Quick start EventLogger** ............................................................................................................... 38  
   7.1 **Creating a configuration** ......................................................................................................... 38  
   7.2 **Configuring Event-To-Speech** ............................................................................................... 39  
   7.3 **Configuring Speech-To-Event** ............................................................................................... 41  

8 **Configuration** .................................................................................................................................. 44  
   8.1 **User interface** .......................................................................................................................... 44  
   8.2 **General Configurations** ......................................................................................................... 44  
   8.3 **Testing the sound card** .......................................................................................................... 50  
   8.4 **Configuring a grammar file** .................................................................................................. 52  
   8.5 **Microsoft SAPI: installing additional languages** .................................................................. 55  

9 **SPS API** .......................................................................................................................................... 56  
   9.1 **Data Types** ........................................................................................................................... 56  
      9.1.1 **Enums** ............................................................................................................................. 56  
   9.2 **Function Blocks** ...................................................................................................................... 58  
      9.2.1 **License Overview** ........................................................................................................... 58  
      9.2.2 **FB_Play** .......................................................................................................................... 58  
      9.2.3 **FB_SetRuleState** ............................................................................................................ 60  
      9.2.4 **FB_SpeechRecognition** ................................................................................................. 61  
      9.2.5 **FB_TextToSpeech** .......................................................................................................... 63  

10 **Samples** ....................................................................................................................................... 65  
   10.1 **Sample 01: Simple ASR and TTS sample** .......................................................................... 65
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.2 Sample 02: Complex ASR</td>
<td>66</td>
</tr>
<tr>
<td>10.3 Sample 03: EventLogger</td>
<td>66</td>
</tr>
<tr>
<td>10.4 Sample 04: File Playback</td>
<td>67</td>
</tr>
<tr>
<td>10.5 Sample 05: Activation/deactivation of individual recognition tags</td>
<td>67</td>
</tr>
<tr>
<td>10.6 Sample 06: TwinCAT HMI with TwinCAT Speech Integration</td>
<td>68</td>
</tr>
<tr>
<td>11 FAQ</td>
<td>69</td>
</tr>
<tr>
<td>12 Appendix</td>
<td>70</td>
</tr>
<tr>
<td>12.1 Error List</td>
<td>70</td>
</tr>
<tr>
<td>12.2 Third-party Components</td>
<td>71</td>
</tr>
<tr>
<td>12.3 Support and Service</td>
<td>72</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!

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

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](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](https://www.beckhoff.com/secinfo).
2 Overview

TwinCAT Speech enables people and machines to interact via speech input and output.

The interaction with TwinCAT takes place in different languages and can be used across industries in many applications from mechanical engineering to building automation. For example, the operating or maintenance staff can be informed about or influence the state of the controller.

TwinCAT Speech provides the following functions:

- Communication via the audio output of the PC-based controller (sound card with speakers, headset, etc.)
- Speech input and output in the languages supported by Microsoft SAPI.
- Speech output in the languages that Amazon Polly provides.
- Caching of generated speech outputs

Through the use of Microsoft SAPI and the caching of speech outputs that may be generated online, you can achieve offline use.

TwinCAT Speech is based on the two components Automatic Speech Recognition (ASR) and Text-to-Speech (TTS). ASR is the speech recognition and is responsible for ensuring that speech input, which is recorded via a microphone, is usable in the PLC. TTS is the text-to-speech output and enables the machine to output speech.

TwinCAT Speech has an online function that is provided with the aid of the Amazon text-to-speech service Polly. It is possible to set different voices and to cache the audio files generated online.

The speech input has an offline function, which is realized by the built-in functions of the Windows operating system.
3  **Installation**

### 3.1 System requirements

<table>
<thead>
<tr>
<th>Technical data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>Windows 10</td>
</tr>
<tr>
<td>Target platform</td>
<td>x64</td>
</tr>
<tr>
<td>TwinCAT version</td>
<td>3.1 Build 4024.12</td>
</tr>
<tr>
<td>TwinCAT licenses</td>
<td>TC1200, TF4500 on the XAR</td>
</tr>
<tr>
<td>Engineering</td>
<td>XaeShell 4024.0, Visual Studio 2017/ 2019</td>
</tr>
<tr>
<td>Audio devices</td>
<td>Windows compatible</td>
</tr>
<tr>
<td>Minimum TwinCAT HMI version</td>
<td>1.12.744.0</td>
</tr>
<tr>
<td></td>
<td>A TwinCAT HMI client license is required for the TwinCAT Speech connection.</td>
</tr>
</tbody>
</table>

### 3.2 Installation

A separate installer is used for the installation.

To install TwinCAT Speech, first select the desired language. Then follow the instructions in the setup wizard.

Installation is required on both the Engineering (XAE) and Runtime (XAR) side.

The .NET Core Runtime must be installed on the TwinCAT HMI Server side in order to use the TwinCAT HMI ServerExtension.

### 3.3 Licensing

The TwinCAT 3 function can be activated as a full version or as a 7-day test version. Both license types can be activated via the TwinCAT 3 development environment (XAE).

#### Licensing the full version of a TwinCAT 3 Function

A description of the procedure to license a full version can be found in the Beckhoff Information System in the documentation "TwinCAT 3 Licensing".

#### Licensing the 7-day test version of a TwinCAT 3 Function

- A 7-day test version cannot be enabled for a TwinCAT 3 license dongle.

1. Start the TwinCAT 3 development environment (XAE).
2. Open an existing TwinCAT 3 project or create a new project.
3. If you want to activate the license for a remote device, set the desired target system. To do this, select the target system from the **Choose Target System** drop-down list in the toolbar.

   - The licensing settings always refer to the selected target system. When the project is activated on the target system, the corresponding TwinCAT 3 licenses are automatically copied to this system.
4. In the Solution Explorer, double-click License in the SYSTEM subtree.

   ![Solution Explorer](image)
   
   The TwinCAT 3 license manager opens.

5. Open the Manage Licenses tab. In the Add License column, check the check box for the license you want to add to your project (e.g. “TF4100 TC3 Controller Toolbox”).

   ![Manage Licenses](image)

6. Open the Order Information (Runtime) tab.

   In the tabular overview of licenses, the previously selected license is displayed with the status “missing”.

   ![Order Information (Runtime)](image)
7. Click **7-Day Trial License...** to activate the 7-day trial license.

A dialog box opens, prompting you to enter the security code displayed in the dialog.

8. Enter the code exactly as it is displayed and confirm the entry.
9. Confirm the subsequent dialog, which indicates the successful activation.

   In the tabular overview of licenses, the license status now indicates the expiry date of the license.

10. Restart the TwinCAT system.

    The 7-day trial version is enabled.
4 Technical introduction

TwinCAT Speech uses the sound cards of the Windows operating system for speech input and output. These are typically connected via USB. Other sound cards can also be used as long as Windows recognizes them as sound devices.

With the configuration project, TwinCAT Speech represents a dedicated project system in the engineering in order to enable the selection of sound cards and languages as well as further connections to the TwinCAT HMI or TwinCAT EventLogger. The configuration is loaded to the corresponding target system by activation and is available for use from the next start of TwinCAT.

A TwinCAT HMI client can be used as a virtual sound card. This client can run on the same or another computers or indeed a mobile device, for example. All you need is an HTML5-compatible browser. However, an HMI user interface can also be directly voice-operated.

The following graphic illustrates the architecture of TwinCAT Speech:

![Architecture of TwinCAT Speech](image)

See also:

- TwinCAT HMI
- TwinCAT EventLogger

4.1 Available languages

TwinCAT Speech uses two subordinated components to realize speech recognition and speech output: Microsoft SAPI for ASR and TTS, and Amazon Polly for TTS only. This allows a variety of languages to be provided.

The following tables provide an overview of the languages that are available.
Microsoft SAPI - speech recognition (ASR)

<table>
<thead>
<tr>
<th>Language</th>
<th>LangID</th>
<th>Microsoft SAPI – ASR</th>
</tr>
</thead>
<tbody>
<tr>
<td>German</td>
<td>1031</td>
<td>de-DE</td>
</tr>
<tr>
<td>English</td>
<td>1033</td>
<td>en-US</td>
</tr>
<tr>
<td></td>
<td>2057</td>
<td>en-GB</td>
</tr>
<tr>
<td>French</td>
<td>1036</td>
<td>fr-FR</td>
</tr>
<tr>
<td>Japanese</td>
<td>1041</td>
<td>ja-JP</td>
</tr>
<tr>
<td>Spanish</td>
<td>1034</td>
<td>es-ES</td>
</tr>
<tr>
<td>Chinese</td>
<td>2052</td>
<td>zh-cn</td>
</tr>
<tr>
<td></td>
<td>3076</td>
<td>zh-hk</td>
</tr>
</tbody>
</table>

Note that the appropriate Windows language packs must be installed in order to be used. This is described here.

Microsoft SAPI - speech output (TTS)

<table>
<thead>
<tr>
<th>Language</th>
<th>LangID</th>
<th>Microsoft SAPI – TTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>German</td>
<td>1031</td>
<td>Hedda</td>
</tr>
<tr>
<td>English</td>
<td>1033</td>
<td>Zira, David</td>
</tr>
<tr>
<td></td>
<td>2057</td>
<td>Hazel</td>
</tr>
<tr>
<td>French</td>
<td>1036</td>
<td>Hortense</td>
</tr>
<tr>
<td>Italian</td>
<td>1040</td>
<td>Elsa</td>
</tr>
<tr>
<td>Japanese</td>
<td>1041</td>
<td>Haruka</td>
</tr>
<tr>
<td>Spanish</td>
<td>3082</td>
<td>Helena</td>
</tr>
<tr>
<td></td>
<td>2058</td>
<td>Sabina</td>
</tr>
<tr>
<td>Korean</td>
<td>1042</td>
<td>Heami</td>
</tr>
<tr>
<td>Polish</td>
<td>1045</td>
<td>Paulina</td>
</tr>
<tr>
<td>Portuguese</td>
<td>1046</td>
<td>Maria</td>
</tr>
<tr>
<td>Russian</td>
<td>1049</td>
<td>Irina</td>
</tr>
<tr>
<td>Chinese</td>
<td>2052</td>
<td>Huihui</td>
</tr>
<tr>
<td></td>
<td>3076</td>
<td>Tracy</td>
</tr>
<tr>
<td></td>
<td>1028</td>
<td>Hanhan</td>
</tr>
</tbody>
</table>

Please note that the appropriate Windows language packs must be installed in order to be used. This is described here [55].

Other languages may be available, but have not been tested with TwinCAT Speech. Some of these are reserved for Cortana and TwinCAT Speech cannot use them via the SAPI: https://support.microsoft.com/en-us/help/22805/windows-10-supported-narrator-languages-voices

The basic compatibility can be checked as follows.
1. Install the desired language.
2. Open the **Text-to-Speech** menu from the “Control Panel”.

![Control Panel](Image)

If languages are displayed here, they should also be usable with TwinCAT Speech.
Amazon Polly - speech output (TTS)

<table>
<thead>
<tr>
<th>Language</th>
<th>LangID</th>
<th>Amazon Polly – TTSTTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>German</td>
<td>1031</td>
<td>Hans, Marlene, Vicki</td>
</tr>
<tr>
<td>English</td>
<td>1033</td>
<td>Ivy, Joanna, Kendra, Kimberly, Salli, Joey, Justin, Kevin, Matthew</td>
</tr>
<tr>
<td></td>
<td>3081</td>
<td>Nicole, Russell</td>
</tr>
<tr>
<td></td>
<td>2057</td>
<td>Amy, Emma, Brian</td>
</tr>
<tr>
<td></td>
<td>16393</td>
<td>Aditi, Raveena</td>
</tr>
<tr>
<td>Danish</td>
<td>1030</td>
<td>Naja, Mads</td>
</tr>
<tr>
<td>Dutch</td>
<td>1043</td>
<td>Ruben, Lotte</td>
</tr>
<tr>
<td>French</td>
<td>1036</td>
<td>Celine, Léa, Mathieu</td>
</tr>
<tr>
<td></td>
<td>3084</td>
<td>Chantal</td>
</tr>
<tr>
<td>Icelandic</td>
<td>1039</td>
<td>Dora, Karl</td>
</tr>
<tr>
<td>Italian</td>
<td>1040</td>
<td>Carla, Bianca, Giorgio</td>
</tr>
<tr>
<td>Japanese</td>
<td>1041</td>
<td>Mizuki, Takumi</td>
</tr>
<tr>
<td>Norwegian</td>
<td>1044</td>
<td>Liv</td>
</tr>
<tr>
<td>Korean</td>
<td>1042</td>
<td>Seoyeon</td>
</tr>
<tr>
<td>Polish</td>
<td>1045</td>
<td>Ewa, Maja, Jacek, Jan</td>
</tr>
<tr>
<td>Portuguese</td>
<td>1046</td>
<td>Camila, Vitória/Vitoria, Ricardo</td>
</tr>
<tr>
<td></td>
<td>2070</td>
<td>Inês/Ines, Cristiano</td>
</tr>
<tr>
<td>Romanian</td>
<td>1048</td>
<td>Carmen</td>
</tr>
<tr>
<td>Russian</td>
<td>1049</td>
<td>Tatyana, Maxim</td>
</tr>
<tr>
<td>Spanish</td>
<td>1034</td>
<td>Conchita, Lucia, Enrique</td>
</tr>
<tr>
<td></td>
<td>2058</td>
<td>Mia</td>
</tr>
<tr>
<td>Swedish</td>
<td>1053</td>
<td>Astrid</td>
</tr>
<tr>
<td>Turkish</td>
<td>1055</td>
<td>Filiz</td>
</tr>
<tr>
<td>Welsh</td>
<td>1106</td>
<td>Gwyneth</td>
</tr>
</tbody>
</table>

Further information on Amazon Polly can be found here: [https://docs.aws.amazon.com/polly/latest/dg/voicelist.html](https://docs.aws.amazon.com/polly/latest/dg/voicelist.html)

To support new languages, you need to update TwinCAT Speech.

See also:

Microsoft SAPI: installing additional languages [55]

Also see about this

Microsoft SAPI: installing additional languages [55]

### 4.1.1 Setting Microsoft SAPI

Microsoft's SAPI speech synthesis service is usable without an active Internet connection. In the basic version, it provides the languages of the language packs installed on the computer and one voice per language.

The configuration of the SAPI takes place during the configuration of the respective sound card.

If you cannot select the desired language, you must install it; see Installing additional languages.
### 4.1.2 Setting Amazon Polly

The speech synthesis service Polly works on the basis of a cloud system and each query placed via Polly costs money. For this reason, you can only use Polly with an active Internet connection and an AWS account.

You create an account to use Polly via AWS Identity and Access Management (IAM).

2. Place the access key to use Polly in the TwinCAT Speech configuration.

3. If you configure Polly as a speech recognition service in the TTS Configuration Wizard, you must enter your access key.
5  Quick start: speech input (ASR)

The following chapter is intended to simplify the configuration of TwinCAT Speech for speech input. The instructions do not address more complex configurations at this point. These can be found in the chapter Configuration [44].

5.1  Creating a configuration

TwinCAT Speech is parameterized via its own configuration project in TwinCAT Engineering. New TwinCAT Speech configurations are created as follows:

1. Open TwinCAT Engineering and create a new project.
2. Select an empty TwinCAT Controller project, name it, and click OK.

   ![New Project Screen]

   The new TwinCAT Controller project opens.

3. Right-click the new TwinCAT Controller project.
4. Select **Add>New Item**.

![Add New Item window](image)

⇒ The **Add New Item** window opens.

5. Select TwinCAT Speech Configurator, name the configuration, and click **Add**.

![Add New Item - TwinCAT Controller window](image)

⇒ The start window for the TwinCAT Speech configuration opens.

6. Select the desired target system.

⇒ Assignment to a target system is important for TwinCAT Speech, because the hardware configuration of the sound cards is individual.

⇒ A new TwinCAT Speech configuration has now been created.

### 5.2 Configuring speech input

A configuration wizard is available for the configuration of the speech input (ASR). This guides you through the ASR configuration.
1. To open the ASR Configuration Wizard, right click TwinCAT Speech Configurator in the Solution Explorer.

2. Select **Start ASR Wizard** from the context menu.
   - The ASR Configuration Wizard opens.
   - In the lower section you will see a list of all already configured microphones, which is empty in the case of a new project. A selected microphone is listed in the upper section.
   - In the first part of the configuration, the wizard guides you through the configuration of the microphone input of the sound card.

3. To add a new microphone input, click the + button.
4. In the Select Device drop-down list, select the appropriate microphone input. If you select the Default Windows Input Device here, the default sound card set in the operating system will always be used.

5. Click the Save button.
   - The Device Wizard closes.

6. If necessary, mark the newly created microphone input and click Select.
   - The microphone input is added to the list Selected Device for ASR.

7. Click the Next button.
The second part of the configuration opens, which is where you configure the actual speech recognition service. As in the first step, a list of services that have already been configured is shown below. The selected service is in the list **Selected Service for ASR Configuration**.

8. To add a speech recognition service, click the + button.
   - The ASR Service Wizard opens and guides you through the configuration of a speech recognition service and an SRGS file.

9. Now configure a speech recognition service.
10. Select the desired provider from the **ASR Service** drop-down list.
11. You can give the speech recognition service its own name in the **ASR Service Name** input field.
12. Configure a grammar file with the sentences to be recognized.
13. To add a grammar file, click the + button.
14. Create a new grammar file by clicking **Create new Grammar File**.

15. Name the file in the **Grammar Name** input field.

16. Select the desired default language from the **Language** drop-down list. You can store several languages in the SRGS Editor and switch between them later, for example from the PLC or HMI.

17. Click the + button. A new row is added.

18. Click the **Save** button. The configured grammar file appears in the **Grammar File** drop-down list and the selected language is shown in the **Language** box.

19. If you want to save the speech inputs, check the **Retain Audio** check box.

20. In the **Folder Path** input box, enter the location where you want to store the speech inputs.

21. Click the **Save** button.
The ASR Service Wizard closes.

22. If speech recognition services have already been created, select the newly created speech recognition service and click **Select**.
   - The speech recognition service is added to the list in the section **Selected Service for ASR Configuration**.
23. Click the **Finish** button.
   - At the end of the configuration, TwinCAT Speech creates an identification number for it. This can be found in the list of ASR configurations under **Configuration Id** and you need it for the PLC programming of the TwinCAT Speech project.

   - The TwinCAT Speech configuration is activated on the target system and can be used by the PLC.
5.3 Programming the PLC

A PLC project must be programmed in order to use TwinCAT Speech. For a quick start, this is explained below on the basis of this sample.

1. Create a new PLC project.

2. Add the Tc3_Speech library.
3. Insert the following code from Sample01 into MainASR.

Declaration part:

```
PROGRAM MainASR
VAR
  // ASR Trigger Variables
  bListenTrigger : BOOL := FALSE;
  bListenStopTrigger : BOOL := FALSE;
  // TTS Command Configuration
  nConfigIdTTS : UINT := 200;
  // ASR Command Configuration
  nConfigIdASR : UINT := 100;
  // Language Id for TTS Output
  nLanguageId : UINT := 1033;

  // fbTTS is required for providing TTS responses upon successful recognitions in this sample.
  fbTTS : FB_TextToSpeech := (nConfigurationId := nConfigIdTTS);
  fbASR : FB_SpeechRecognition := (nConfigurationId := nConfigIdASR);
  fbRetrieveUtterance : FB_RetrieveUtterance;

  // Update Code before Release
  // TTS Variables (required for TTS responses upon recognition)
  bSpeak : BOOL := FALSE;
  sText2Speech : STRING(4096) := '<speak> TcSpeech beta demo project is greeting you </speak>';  
  // ASR Variables
  bListen : BOOL := FALSE;
  nLastRecoId : ULINT := 0;
  sLastRecognition : STRING(4096) := '';  
  fConfidenceThreshold : REAL := 0.7; // TcSpeech will inform PLC if recognition confidence is
  // larger than this threshold
  bInit : BOOL := FALSE;
END_VAR
```

Process part:

```
IF NOT bInit THEN  
  fbTTS.SetAmsNetAddr(GVL_SpeechDemo.sAmsNetId);
  fbASR.SetAmsNetAddr(GVL_SpeechDemo.sAmsNetId);
  bInit := TRUE;
ENDIF

// ASR - Automatic Speech Recognition
// Trigger Start/Stop Listening
IF bListenTrigger THEN
  bListenTrigger := FALSE;
  bListen := TRUE;
ENDIF

IF bSpeak THEN // for answering a recognized ASR command ...
  fbTTS(sUtterance := sText2Speech,bSpeak := TRUE, nConfigurationId:= nConfigIdTTS,
  nLanguageId:=nLanguageId);
  IF NOT fbTTS.bBusy THEN
    fbTTS(sUtterance := sText2Speech,bSpeak := FALSE, nConfigurationId:= nConfigIdTTS);
    bSpeak := FALSE;
  END_IF
ENDIF

IF bListen THEN
  IF bListenStopTrigger THEN
    fbASR(bListen := FALSE,nConfigurationId:= nConfigIdASR);
  ELSE
    fbASR(bListen := TRUE,nConfigurationId:= nConfigIdASR);
  END_IF
  IF NOT fbASR.bBusy THEN
    fbASR(bListen:= FALSE,nConfigurationId:= nConfigIdASR);
    bListen := FALSE;
    bListenStopTrigger := FALSE;
  END_IF
ENDIF

// Check if new Recognition is available
IF nLastRecoId <> fbASR.nRecognitionId THEN
  IF NOT bSpeak THEN
    IF fbAsr.fRecognitionConfidence > fConfidenceThreshold THEN // if recognition better than
      // treshold, answer via TTS. Next cycle will process via FB_SwitchLanguage
        fbRetrieveUtterance(nLanguageId := nLanguageId, sRecognitionTag :=
        fbASR.sRecognitionTag,fRecognitionConfidence := fbAsr.fRecognitionConfidence, sText2Speech =>
        sText2Speech);
        bSpeak := TRUE;
      END_IF
  END_IF
ENDIF
END_IF
END_VAR
```
4. Set "bListen" to TRUE to make a speech input.
   The TwinCAT configuration must have been activated beforehand.

   The identification number for the ASR configuration to be used is present in the code as `ConfigIdASR`.
6   Quick start: speech output (TTS)

The following chapter is intended to illustrate the configuration of TwinCAT Speech for speech output.

The instructions do not address more complex configurations at this point. These can be found in the chapter Configuration [44].

6.1   Creating a configuration

TwinCAT Speech is parameterized via its own configuration project in TwinCAT Engineering. New TwinCAT Speech configurations are created as follows:

1. Open TwinCAT Engineering and create a new project.
2. Select an empty TwinCAT Controller project, name it, and click OK.

![New Project Dialog]

- The new TwinCAT Controller project opens.
3. Right-click the new TwinCAT Controller project.
4. Select **Add>New Item**.

The **Add New Item** window opens.

5. Select TwinCAT Speech Configurator, name the configuration, and click **Add**.

The start window for the TwinCAT Speech configuration opens.

6. Select the desired target system.

- Assignment to a target system is important for TwinCAT Speech, because the hardware configuration of the sound cards is individual.
- A new TwinCAT Speech configuration has now been created.

### 6.2 Configuring speech output

A configuration wizard is available for the configuration of the speech output (TTS). This guides you through the TTS configuration.
Quick start: speech output (TTS)

1. To open the TTS Configuration Wizard, right-click the TwinCAT Speech configuration in the Solution Explorer.

2. In the context menu, select **Start TTS Wizard**
   - The TTS Configuration Wizard opens.
   - In the lower section you will see a list of all already configured playback devices, which is empty in the case of a new project. A selected playback device is listed in the upper section.
   - In the first part, the wizard guides you through the configuration of the playback device of a sound card.

3. To add a new playback device, click the + button.
4. In the **Select Device** drop-down list, select the appropriate playback device. If you select the "Default Windows Output Device" here, the default sound card set in the operating system will always be used.

5. Click the **Save** button.

   - The Device Wizard closes.

6. If necessary, mark the desired playback device and click the **Select** button.

   - The playback device is added to the list in the upper section **Selected Playback Device for TTS Configuration**.
7. Click on **Next**.

The second part of the configuration opens, where you configure the actual speech generation service (Text-To-Speech).

As in the first step, a list of services that have already been configured is shown below. The selected service is displayed above.

8. To add a speech service, click the + button.

The TTS Service Wizard opens.
9. Select the desired provider from the drop-down list **TTS Synthesis Service**. For Amazon Polly, the access key is then queried, which is stored in the solution and also on the target system.

![TTS Service Wizard](image)

10. You can give the speech recognition service its own name in the **TTS Service Name** input field.

11. Now set a language and voice for the speech output.

![TTS Service Wizard](image)

12. You can use the Magic Wand button to automatically generate the available languages and voices.

13. Alternatively, select the desired language in the first drop-down list **Select Language** and an available voice in the second drop-down list **Select Voice**.

14. Click the + button.

15. In the third drop-down list **Default Language**, select a default language.
   
   If you cannot select the desired language, install it on Windows SAPI. See

16. Click the **Save** button.
17. If speech recognition services have already been created, mark the desired speech recognition service in the list and click the **Select** button.

18. Click the **Next** button to save the TTS configuration.

19. To save and stop the configuration at this point, click the **Finish** button.

   When you click the **Next** button, the third part of the configuration opens. In the optional third part of the configuration, the wizard concludes by guiding you through the local saving of the generated TTS outputs on the target system. This allows you to reuse them at any time.

20. To determine whether the TTS outputs should be saved, click the + button.
The TTS Storage Wizard opens.

21. In the **Storage Name** input box, enter a name for the file to be saved.
22. In the **Folder Path** box, determine a memory location for the file to be saved.
23. In the **Memory Usage** area, you determine whether all TTS outputs are stored or whether the memory space should be limited. Select the appropriate radio button for this.
24. In the **Max. Size** input box, enter how much memory space should be allocated for the saved speech outputs.
25. Click the **Save** button.
26. If memory locations have already been created, mark the desired memory location in the list and click the **Select** button.

27. Click the **Finish** button.
At the end of the configuration, TwinCAT Speech creates an identification number for it. This can be found in the list of TTS configurations under **Configuration Id** and you need it for the PLC programming of the TwinCAT Speech project.

28. Activate the TwinCAT Speech configuration on the target system by clicking the **Activate Configuration** button.

The TwinCAT Speech configuration is activated on the target system and can be used by the PLC.

**Also see about this**

- Microsoft SAPI: installing additional languages [55]

### 6.3 Programming the PLC

A PLC project must be programmed in order to use TwinCAT Speech. For a quick start, this is explained below on the basis of this sample [65].
1. Create a new PLC project.

![New Project Window]

2. Add the Tc3_Speech library.

![Add Library Window]

3. Insert the following code from Sample01 into MainTTS.

Declaration part:
PROGRAM MainTTS
VAR
  // TTS Trigger Variables
  bSpeakTrigger : BOOL := FALSE;
  bSpeakStopTrigger : BOOL := FALSE;
  // TTS Command Configuration
  nConfigIdTTS : UINT := 200;
  // Language Id for TTS Output
  nLanguageId : UINT := 1033;
  // Update Code before Release
  fbTTS : FB_TextToSpeech := (nConfigurationId := nConfigIdTTS);
  fbRetrieveUtterance : FB_RetrieveUtterance;
  // TTS Variables
  bSpeak : BOOL := FALSE;
  // attribute 'TcEncoding':= 'UTF-8'
  sText2Speech : STRING(4095) := '<speak>TcSpeech beta demo project is greeting you.</speak>';
  bInit : BOOL := FALSE;
  nLanguageIdOld : UINT := 1033;
END_VAR

Process part:
IF NOT bInit THEN
  fbTTS.SetAmSNetAddr(GVL_SpeechDemo.sAmsNetId);
  bInit := TRUE;
END_IF
// TTS - Text To Speech
// Trigger Start/Stop Text Output
IF bSpeakTrigger THEN
  bSpeakTrigger := FALSE;
  bSpeak := TRUE;
END_IF
IF (nLanguageIdOld <> nLanguageId) THEN
  fbRetrieveUtterance(nLanguageId := nLanguageId, sText2Speech => sText2Speech, nOldLanguageId := nLanguageIdOld, sOldText2Speech := sText2Speech);
  nLanguageIdOld := nLanguageId;
END_IF
IF bSpeak THEN // if set manually via bSpeakTrigger OR for answering a recognized ASR command ...
  IF bSpeakStopTrigger THEN
    fbTTS(sUtterance := sText2Speech, bSpeak := FALSE, nConfigurationId := nConfigIdTTS);
  ELSE
    fbTTS(sUtterance := sText2Speech, bSpeak := TRUE, nConfigurationId := nConfigIdTTS, nLanguageId := nLanguageId);
  END_IF
END_IF
IF NOT fbTTS.bBusy THEN
  fbTTS(sUtterance := sText2Speech, bSpeak := FALSE, nConfigurationId := nConfigIdTTS);
  bSpeak := FALSE;
  bSpeakStopTrigger := FALSE;
END_IF
END_IF

4. Set "bSpeak" to TRUE to trigger a speech output via the default device. 
The TwinCAT configuration must have been activated beforehand.

The identification number for the TTS configuration that is to be used is present in the code as ConfigIdTTS.
7 Quick start EventLogger

The following chapter is intended to simplify the configuration of the interface between TwinCAT Speech and the TwinCAT EventLogger. Both a speech input to trigger a TwinCAT event (Speech-To-Event) and a speech output on occurrence of a TwinCAT event (Event-To-Speech) are enabled.

Also see about this
- Configuring Event-To-Speech [39]
- Configuring Speech-To-Event [41]

7.1 Creating a configuration

TwinCAT Speech is parameterized via its own configuration project in TwinCAT Engineering. New TwinCAT Speech configurations are created as follows:

1. Open TwinCAT Engineering and create a new project.
2. Select an empty TwinCAT Controller project, name it, and click OK.

   ![New Project Window]

   - The new TwinCAT Controller project opens.
3. Right-click the new TwinCAT Controller project.
4. Select **Add>New Item**.

5. Select TwinCAT Speech Configurator, name the configuration, and click **Add**.

6. Select the desired target system.
   - Assignment to a target system is important for TwinCAT Speech, because the hardware configuration of the sound cards is individual.
   - A new TwinCAT Speech configuration has now been created.

### 7.2 Configuring Event-To-Speech

A configuration wizard is available for the configuration of the speech output of events. This guides you through the Event-To-Speech configuration.

- A speech output should be configured beforehand, as it will be used here. To do this, start with the chapter **Quick start: speech output (TTS)**. [p. 27]
1. In order to open the Event-To-Speech wizard, right-click the EventLogger configuration in the Solution Explorer.

2. Select **Start Event-To-Speech Wizard** from the context menu.
   - The Event-To-Speech wizard starts.

3. Select the respective TwinCAT project under Main Configuration.
4. Add a new event by selecting the event class, the event and the TTS configuration from the respective drop-down list.

5. Click the Add button.

6. Click the Save button.
   - The configuration is created.

7. Activate the TwinCAT Speech configuration as usual.

7.3 Configuring Speech-To-Event

A configuration wizard is available for the configuration of the speech recognition with subsequent triggering of events. This guides you through the Speech-To-Events configuration.
1. In order to open the Speech-To-Event wizard, right-click the EventLogger configuration in the Solution Explorer.

![Solution 'TwinCAT Controller' (1 project)](image)

2. Select **Start Speech-To-Event Wizard** from the context menu.
   - The Speech-To-Event wizard starts.

3. Select the respective TwinCAT project under Main Configuration.
4. Add a new event by selecting the event class, the event and the Speech-To-Event configuration from the respective drop-down list.

5. Click the **Add** button.

6. Click the **Save** button.

   
   - The configuration is created.

   - The TwinCAT event is triggered when the corresponding speech recognition takes place.

The associated sample [67] provides a PLC project for triggering events in a simple way.
8 Configuration

The dialogs of the TwinCAT Speech Configurator project are described in this chapter. Quickstart instructions can be used to understand the process.

8.1 User interface

If you have created the TwinCAT Speech project, the window looks like this:

<table>
<thead>
<tr>
<th></th>
<th>Solution Explorer</th>
<th>The TwinCAT Speech Configurator is mapped as its own project type below a TwinCAT Controller project.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>General Configuration</td>
<td>Selection of the target system as well as activation and logging settings.</td>
</tr>
<tr>
<td>3</td>
<td>ASR Configuration</td>
<td>Combines a microphone with an ASR service. The configuration IDs are used, for example, in the PLC.</td>
</tr>
<tr>
<td>4</td>
<td>TTS Configuration</td>
<td>Combines a playback device with a TTS service. The configuration IDs are used, for example, in the PLC.</td>
</tr>
<tr>
<td>5</td>
<td>Devices</td>
<td>Configuration of microphones and playback devices. These are used in the ASR and TTS configurations. They are usually created by the ASR / TTS wizards.</td>
</tr>
<tr>
<td>6</td>
<td>ASR</td>
<td>Configuration of the speech recognition services. These are used in the ASR configurations. They are usually created by the ASR wizard.</td>
</tr>
<tr>
<td>7</td>
<td>TTS</td>
<td>Configuration of the speech output services. These are used in the TTS configuration. They are usually created by the TTS wizard.</td>
</tr>
<tr>
<td>8</td>
<td>EventLogger</td>
<td>Speech input and output based on TwinCAT Events.</td>
</tr>
<tr>
<td>9</td>
<td>HMI</td>
<td>Speech input and output in combination with the TwinCAT HMI.</td>
</tr>
</tbody>
</table>

8.2 General Configurations

The General Configurations window provides the possibility to make various settings.
"Activate Configuration": This transfers the configuration to the target system and restarts it for activation upon request.

**Note** Every change in the TwinCAT Speech Configurator has to be activated first!

Reload available targets.

**Target System**
Target system for which the configuration is intended. The configuration is transferred to this system with "Activate Configuration". The device configuration dialogs also obtain the available sound cards from this device.

The status of the currently connected target system is also displayed in the top right-hand corner.

When selecting a target system, it is important that the hardware requirements of the target system match the TwinCAT Speech configuration. For example, if the configured audio devices do not match the audio devices on the target system, the configuration will not work on the target system.

In addition, a warning is displayed if the target system is changed.

**Log Level**
Log Level (default setting: 3) describes the level of detail with which a log file is written. The file is used to diagnose problems and is located at `C:\TwinCAT\Functions\TF4500-Speech\Boot\TcSpeechLog.txt`

The higher the log level, the more disk space the file uses.

An archive of older logs is created, which can be used for long-term analysis if necessary.

**ASR Configurations**

Add a microphone.

Delete a microphone.

Edit the configuration of a microphone.

**List**
The list of currently configured speech recognitions: this combines an ASR service with a microphone and is used, for example, by the PLC to perform speech recognition.
**TTS Configurations**

Add a playback device.

Delete a playback device.

Edit the configuration of a playback device.

List

The list of currently configured speech outputs:

- this combines a TTS service with a playback device and is used, for example, by the PLC to perform a speech output.

**Devices**

Add a recording device.

Delete a recording device.

Edit the configuration of a recording device.

List

List of currently configured microphones (recording devices):

- these are usually created by the ASR wizard, but can also be created directly. They are combined with ASR services in the ASR configuration.
### Configuration

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Add a playback device.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete a playback device.</td>
</tr>
<tr>
<td>Edit</td>
<td>Edit the configuration of a playback device.</td>
</tr>
</tbody>
</table>

**List**

List of currently configured playback devices:
- these are usually created by the TTS wizard, but can also be created directly. They are combined with TTS services in the TTS configuration.

### ASR

**Add**

Add a speech recognition service.

**Delete**

Delete a speech recognition service.

**Edit**

Edit the configuration of a speech recognition service.

**List**

List of currently configured speech recognition services:
- these are usually created by the ASR wizard, but can also be created directly. They are combined with microphones (recording devices) in the ASR configuration.

### TTS

**Add**

Add a playback service.

**Delete**

Delete a playback service.

**Edit**

Edit the configuration of a playback service.

**List**

List of currently configured playback services:
- these are usually created by the TTS wizard, but can also be created directly. They are combined with playback devices in the TTS configuration.
**EventLogger**

The exact process for the use of the event logger integration is explained in the chapter Quickstart: EventLogger [38].

<table>
<thead>
<tr>
<th>Add an event.</th>
<th>Delete an event.</th>
<th>Edit the configuration of an event.</th>
</tr>
</thead>
</table>

**Event-To-Speech**

The events that should be output via speech output on occurrence are defined on the "Event-to-Speech" tab. A TTS configuration is used to reference the TTS service and playback device to be used. The default language of the TTS service is used.

<table>
<thead>
<tr>
<th>Add an event.</th>
<th>Delete an event.</th>
<th>Edit the configuration of an event.</th>
</tr>
</thead>
</table>

**Speech-To-Event**

The recognition tags that should trigger an event on recognition are defined on the "Speech-To-Event" tab. An ASR configuration is used to reference the ASR (with the associated recognition tags and languages) and microphone to be used.

<table>
<thead>
<tr>
<th>Add an event.</th>
<th>Delete an event.</th>
<th>Edit the configuration of an event.</th>
</tr>
</thead>
</table>
A TwinCAT HMI client license is required for the TwinCAT Speech connection.

The HMI integration is based on the previously created ASR and TTS services. In this case, the device used is the HMI client (browser), which grants access to the sound cards that exist there. This client can run on the same or another computers or indeed a mobile device, for example. All you need is an HTML5-compatible browser.

The TwinCAT HMI is always loaded by the browser from a TwinCAT HMI server. A TwinCAT HMI server extension is required as an intermediary for using TwinCAT Speech. Since the extension must necessarily run on the same HMI server, little preparation is required in the HMI project itself. The TwinCAT Speech extension must be loaded into the project via Nuget. See the corresponding TwinCAT HMI documentation.

TwinCAT Speech requires a TwinCAT HMI server user account, which must be created there.

Since the login data in the TwinCAT Speech configuration are stored unencrypted on the target system, it is recommended to restrict access to these required symbols for the account:

- Speech.ClientConfigs.List
- Speech.ClientConfigs.Register
- Speech.Clients.List
- Speech.Clients.Register
- Speech.Event.Raise
- Heartbeat

**Note** The user "__SpeechCore" is used for communication between TwinCAT Speech and the TwinCAT HMI server. This user is not suitable for logging in via the browser.

TwinCAT Speech must be notified of the host name of the HMI server and the selected user name in the general configuration (see screenshot). Please note that the host name must be valid on the target system at runtime.

TwinCAT HMI automatically reports the current language to TwinCAT Speech. To ensure that this language also changes in the speech recognition when switching to the HMI, an SRGS must be created that recognizes multiple languages. This is described under Configuring a grammar file [p. 52]. Note that only SRGS files created by the Simple SRGS Editor can be used for language switching.

The Recognition Tags defined in the SRGS file can trigger any actions in the HMI. See the TwinCAT HMI documentation.
8.3 Testing the sound card

The TwinCAT Speech Configurator allows you to test and identify sound cards from the target system for later use as a microphone for ASR or as a playback device for TTS. The devices are detected and also identified via the Windows operating system, so the configuration must normally be specific to a target system. An exception to this is the default sound card configured in Windows, which can also be used.

Testing the microphone

1. Go to the ASR node in the TwinCAT Speech project.
2. Start the ASR wizard by right-clicking the ASR node or double-clicking a microphone in the "ASR" list.
3. Click the + button or use the button for an existing device. The device wizard opens.
4. In the drop-down box, select the microphone that you want to test.
5. To test the microphone, click the button.
On clicking the button, the recording is started and played back immediately afterwards.

**Testing the playback device**

1. Click the TTS node in the TwinCAT Speech project.
2. Start the TTS wizard by right-clicking the TTS node or double-clicking a playback device in the "TTS" list.
3. Click the button.
   - The device wizard opens.
4. In the drop-down box, select the desired device.
5. Click on .
   - A test playback is output.
8.4 Configuring a grammar file

For speech recognition, it is necessary to store a grammar file. The required grammar files are stored in the format Speech Recognition Grammar Specification (SRGS). SRGS is a standard that describes how the grammars for speech recognition are specified.

SRGS is a powerful format. TwinCAT Speech supports this standard on 2 levels:

On the one hand, a "Simple SRGS Editor" is offered directly in the TwinCAT Speech Configurator when an ASR service is created. This essentially provides an assignment of tag and text, and the text can be stored in several languages.

On the other hand, external SRGSs, which are created with other editors and are more complex, can also be imported. An example of this use is Sample 03: EventLogger.

Some functions, such as FB_SetRuleState, cannot work in this case because they place special demands on the SRGS, which are ensured by the Simple SRGS Editor. This also affects the TwinCAT HMI connection, so that here too only SRGS created by the Simple SRGS Editor can be used in multiple languages. The opening of complex files in the Simple SRGS Editor is denied.

Simple SRGS Editor

The SRGS file is provided with the ASR service, which is typically created by the ASR wizard. Here you can either use the Simple SRGS Editor to edit a local SRGS file or reference an externally provided SRGS.

In the Simple SRGS Editor, "Recognition Texts" are assigned to the "Recognition Tags". These texts can also be arranged in groups, "Recognition Group", in order to enable or disable them via FB_SetRuleState.

Procedure with the Simple SRGS Editor

1. Click Create new Grammar File while creating or modifying an ASR service.
   - The Simple SRGS Editor opens.
2. Name the SRGS file and select a default language. This language is used if no other language is specified. This is also used for the Speech-To-Event integration.

3. Under Recognition Tag, enter a tag that is provided in the PLC upon recognition and can be reacted to accordingly. This Recognition Tag is also configured for a response within the HMI. See TwinCAT HMI documentation.

4. Under Recognition Text, enter a text to be recognized by ASR.

5. If necessary, enter a group under Recognition Group.

6. Click the Save button.
   - The SRGS file is fully configured.
   - The SRGS file is transferred to the target by activating the TwinCAT Speech Configurator project.

**Entering voice commands in different languages**

The Simple SRGS Editor can be used to recognize multilingual texts.

This is used in particular in connection with the TwinCAT HMI to switch between different languages. Switching does not work simultaneously; the language must always be switched first before a different language can be recognized.
1. To open the SRGS Grammar Translation Editor, click the button.

![SRGS Grammar Editor](image1)

When the editor is opened, all Recognition Tags already defined are displayed.

![SRGS Recognition Translations](image2)

2. Click the button to add a new language.
3. The tab selected on the left defines the language that belongs to the recognition texts.

**Previewing the SRGS file**

By clicking on the button, you can preview the code of the SRGS file in your default Windows editor for XML files.

<table>
<thead>
<tr>
<th>Grammar File:</th>
<th>Language:</th>
</tr>
</thead>
<tbody>
<tr>
<td>srgs</td>
<td>English (United States)</td>
</tr>
</tbody>
</table>
8.5 Microsoft SAPI: installing additional languages

To use both speech input (ASR) and speech output (TTS), you must install the appropriate language packs when using Microsoft SAPI in order to be able to use them with TwinCAT Speech.

Not all languages that Windows offers are compatible. See Available languages [12] for a list of tested languages.

Proceed as follows to install a language:

1. The controller must be able to communicate with the Microsoft server via the network.
2. Open Windows Settings>Time and Language>Language.
3. Click the Add Preferred Language button.
4. Select a language and click Next.
5. Select whether you only want to install the language pack or also set it as the display language.
6. Click Install.
7. The language pack is installed.
8. Click the installed language.
9. Click on Settings.
10. In this settings window you can make all settings that affect speech recognition via MS-SAPI.

If the Language option is missing, you must temporarily enable the Windows Update:

![Image of Windows Update settings window]
9 SPS API

9.1 Data Types

9.1.1 Enums

9.1.1.1 ETcsRecognitionEngineState

State of the speech recognition engine.

Syntax

Definition:

```
TYPE ETcsRecognitionEngineState : 
{ 
    TCS_ENGINE_UNKNOWN := 0,
    TCS_ENGINE_STARTING := 1,
    TCS_ENGINE_STARTED := 2,
    TCS_ENGINE_STOPPING := 3,
    TCS_ENGINE_STOPPED := 4,
    TCS_ENGINE_FAILURE := 5,
    TCS_ENGINE_PAUSED := 6,
    TCS_ENGINE_INACTIVE := 7
} UDINT;
END_TYPE
```

Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCS_ENGINE_UNKNOWN</td>
<td>ASR-Engine in unknown state.</td>
</tr>
<tr>
<td>TCS_ENGINE_STARTING</td>
<td>ASR-Engine is starting up.</td>
</tr>
<tr>
<td>TCS_ENGINE_STARTED</td>
<td>ASR-Engine is started.</td>
</tr>
<tr>
<td>TCS_ENGINE_STOPPING</td>
<td>ASR-Engine is stopping.</td>
</tr>
<tr>
<td>TCS_ENGINE_STOPPED</td>
<td>ASR-Engine is stopped.</td>
</tr>
<tr>
<td>TCS_ENGINE_FAILURE</td>
<td>ASR-Engine is in failure state. See ErrorId for details.</td>
</tr>
<tr>
<td>TCS_ENGINE_PAUSED</td>
<td>ASR-Engine is paused.</td>
</tr>
<tr>
<td>TCS_ENGINE_INACTIVE</td>
<td>ASR-Engine is inactive.</td>
</tr>
</tbody>
</table>

9.1.1.2 ETcsSpeechCommandExitCode

Speech command exit code.

Syntax

Definition:

```
TYPE ETcsSpeechCommandExitCode : 
{ 
    TCS_CMD_SUCCEEDED := 0,
    TCS_CMD_REQUIRED_SERVICE_NOT_FOUND := 10000,
    TCS_CMD_REQUIRED_SERVICE_NOT_INITIALIZED := 10001,
    TCS_CMD_INSUFFICIENT_PARAMETERS := 10100,
    TCS_CMD_ASR_RULES_SETFAILED := 20000,
    TCS_CMD_ASR_INSTANTIATION_FAILURE := 20010,
    TCS_CMD_ASR_STOP_FAILURE := 20020,
    TCS_CMD_ASR_START_FAILURE := 20030,
    TCS_CMD_ASR_PAUSE_FAILURE := 20040,
    TCS_CMD_TTS_STATUS_REQUEST_FAILURE := 30010,
    TCS_CMD_TTS_STOP_REQUEST_FAILURE := 30020,
} UDINT;
END_TYPE
```
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCS_CMD_SUCCEEDED</td>
<td>Speech command succeeded.</td>
</tr>
<tr>
<td>TCS_CMD_REQUIRED_SERVICE_NOT_FOUND</td>
<td>Referenced service not found. Wrong Id configured?</td>
</tr>
<tr>
<td>TCS_CMD_REQUIRED_SERVICE_NOT_INITIALIZED</td>
<td>Referenced service not initialized.</td>
</tr>
<tr>
<td>TCS_CMD_INSUFFICIENT_PARAMETERS</td>
<td>Speech command with insufficient parameters.</td>
</tr>
<tr>
<td>TCS_CMD_ASR_RULES_SETFAILED</td>
<td>Speech command SetRuleState failed.</td>
</tr>
<tr>
<td>TCS_CMD_ASR_INSTANTIATION_FAILURE</td>
<td>Speech engine could not be instantiated.</td>
</tr>
<tr>
<td>TCS_CMD_ASR_STOP_FAILURE</td>
<td>Speech engine could not be stopped.</td>
</tr>
<tr>
<td>TCS_CMD_ASR_PAUSE_FAILURE</td>
<td>Speech engine could not be paused.</td>
</tr>
<tr>
<td>TCS_CMD_TTS_STATUS_REQUEST_FAILURE</td>
<td>Speech engine’s status request failed.</td>
</tr>
<tr>
<td>TCS_CMD_TTS_STOP_REQUEST_FAILURE</td>
<td>Speech engine’s stop request failed.</td>
</tr>
<tr>
<td>TCS_CMD_TTS_SYNTH_ERROR</td>
<td>Generic speech synthesis related error.</td>
</tr>
<tr>
<td>TCS_CMD_TTS_SYNTH_ERROR_CONFIGURATION_DETAILS</td>
<td>Speech synthesis configuration related error.</td>
</tr>
<tr>
<td>TCS_CMD_TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_LANGUAGEID_MISSING</td>
<td>LanguageId is missing.</td>
</tr>
<tr>
<td>TCS_CMD_TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_LANGUAGEID_NOTSUPPORTED</td>
<td>LanguageId is not supported.</td>
</tr>
<tr>
<td>TCS_CMD_TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_INCOMPLETE</td>
<td>Speech synthesis configuration is incomplete.</td>
</tr>
<tr>
<td>TCS_CMD_TTS_SYNTH_ERROR_UTTERANCE_EMPTY</td>
<td>No utterance is specified.</td>
</tr>
<tr>
<td>TCS_CMD_TTS_SYNTH_ERROR_UTTERANCE_AND_PATH_EMPTY</td>
<td>No utterance or path is specified.</td>
</tr>
<tr>
<td>TCS_CMD_TTS_SYNTH_ERROR_SAPI</td>
<td>Speech synthesis with error of underlying SAPI engine.</td>
</tr>
<tr>
<td>TCS_CMD_TTS_SYNTH_ERROR_AWSPOLLY_</td>
<td>Speech synthesis with error of underlying Amazon Polly engine.</td>
</tr>
<tr>
<td>TCS_CMD_AUDIO_PLAYBACK_START_FAILURE</td>
<td>Audio playback could not be started.</td>
</tr>
<tr>
<td>TCS_CMD_AUDIO_PLAYBACK_START_FAILURE_INVALIDFORMAT</td>
<td>Audio playback could not be started. Invalid audio format.</td>
</tr>
<tr>
<td>TCS_CMD_AUDIO_PLAYBACK_START_FAILURE_FILENOTFOUND</td>
<td>Audio playback could not be started. File not found on target system.</td>
</tr>
</tbody>
</table>
9.2 Function Blocks

9.2.1 License Overview

TC3 Speech

FB_Play [58]
FB_SetRuleState [60]
FB_SpeechRecognition [61]
FB_TextToSpeech [63]

9.2.2 FB_Play

<table>
<thead>
<tr>
<th>bPlay</th>
<th>BOOL</th>
<th>bBusy</th>
</tr>
</thead>
<tbody>
<tr>
<td>sPath</td>
<td>STRING</td>
<td>bError</td>
</tr>
<tr>
<td>nConfigurationId</td>
<td>UINT</td>
<td>nErrorId</td>
</tr>
<tr>
<td>END_VAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nLastCommandExitCode</td>
<td>UINT</td>
<td></td>
</tr>
<tr>
<td>nPlaybackPosition</td>
<td>ULINT</td>
<td></td>
</tr>
<tr>
<td>nPlaybackTotal</td>
<td>ULINT</td>
<td></td>
</tr>
</tbody>
</table>

Playback function block.

Syntax

Definition:

FUNCTION_BLOCK FB_Play
VAR_INPUT
  bPlay : BOOL;
  sPath : STRING;
  nConfigurationId : UINT;
END_VAR
VAR_OUTPUT
  bBusy : BOOL;
  bError : BOOL;
  nErrorId : ETcsSpeechCommandExitCode;
  nLastCommandExitCode : UINT;
  nPlaybackPosition : ULINT;
  nPlaybackTotal : ULINT;
END_VAR

🔍 Inputs

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bPlay</td>
<td>BOOL</td>
<td>Rising flag triggers play command.</td>
</tr>
<tr>
<td>sPath</td>
<td>STRING</td>
<td>Path to audio file on target system. 16khz sample rate, 16 bit sample size, 1 channel expected.</td>
</tr>
<tr>
<td>nConfigurationId</td>
<td>UINT</td>
<td>ConfigurationId of configuration to be used in play request.</td>
</tr>
</tbody>
</table>
### Outputs

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td>Function block is busy.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>Function block is in error state.</td>
</tr>
<tr>
<td>nErrorId</td>
<td>ETcsSpeechCommandExit Code [56]</td>
<td>ErrorId providing details of the occurred error.</td>
</tr>
<tr>
<td>nLastErrorExitCode</td>
<td>UINT</td>
<td>Returncode of last executed command.</td>
</tr>
<tr>
<td>nLastCommandExitCode</td>
<td>ULIINT</td>
<td>Current playback position in file playback (milliseconds).</td>
</tr>
<tr>
<td>nPlaybackPosition</td>
<td>ULIINT</td>
<td>Current playback position in file playback (milliseconds).</td>
</tr>
<tr>
<td>nPlaybackTotal</td>
<td>ULIINT</td>
<td>Total length of file playback (milliseconds).</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetAmsNetAddr [59]</td>
<td>Method to be called once to specify the AmsNetId of destination TcSpeech service.</td>
</tr>
</tbody>
</table>

### Required License

TC3 Speech

### 9.2.2.1 SetAmsNetAddr

Method to be called once to specify the AmsNetId of destination TcSpeech service.

#### Syntax

**Definition:**

```plaintext
METHOD SetAmsNetAddr : HRESULT
  VAR_INPUT
    sAmsNetId : STRING;
  END_VAR
```

#### Inputs

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sAmsNetId</td>
<td>STRING</td>
<td>AmsNetId of destination TcSpeech service.</td>
</tr>
</tbody>
</table>

#### Return value

HRESULT
9.2.3 FB_SetRuleState

SetRuleState function block.

Syntax

Definition:

FUNCTION_BLOCK FB_SetRuleState
VAR_INPUT
  bExecute : BOOL;
  nConfigurationId : UINT;
  sGrammarFile : STRING;
  sGrammarRule : STRING;
  bGrammarActivate : BOOL;
END_VAR
VAR_OUTPUT
  bBusy : BOOL;
  bError : BOOL;
  nErrorId : ETcsSpeechCommandExitCode;
  nLastCommandExitCode : UINT;
END_VAR

**Inputs**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bExecute</td>
<td>BOOL</td>
<td>Rising edge sets the given RuleId.</td>
</tr>
<tr>
<td>nConfigurationId</td>
<td>UINT</td>
<td>ConfigurationId to be used (cmp. TcSpeech Configurator).</td>
</tr>
<tr>
<td>sGrammarFile</td>
<td>STRING</td>
<td>Path to SRGS file on target system. If left empty, the SRGS file specified in the configuration will be used.</td>
</tr>
<tr>
<td>sGrammarRule</td>
<td>STRING</td>
<td>RuleId, which should be used.</td>
</tr>
<tr>
<td>bGrammarActivate</td>
<td>BOOL</td>
<td>Specifies whether the RuleId should be activated or deactivated.</td>
</tr>
</tbody>
</table>

**Outputs**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td>Function block is busy.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>Function block is in error state.</td>
</tr>
<tr>
<td>nErrorId</td>
<td>ETcsSpeechCommandExitCode</td>
<td>ErrorId providing details of the occurred error.</td>
</tr>
<tr>
<td>nLastCommandExitCode</td>
<td>UINT</td>
<td>Returncode of last executed command.</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetAmsNetAddr</td>
<td>Method to be called once to specify the AmsNetId of destination TcSpeech service.</td>
</tr>
</tbody>
</table>
Required License
TC3 Speech

9.2.3.1 SetAmsNetAddr

Method to be called once to specify the AmsNetId of destination TcSpeech service.

Syntax
Definition:
METHOD SetAmsNetAddr : HRESULT
VAR_INPUT
  sAmsNetId : STRING;
END_VAR

Inputs
Name   Type       Description
sAmsNetId STRING AmsNetId of destination TcSpeech service.

Return value
HRESULT

9.2.4 FB_SpeechRecognition

Automatic Speech Recognition function block.

Syntax
Definition:
FUNCTION_BLOCK FB_SpeechRecognition
VAR_INPUT
  bListen                  : BOOL;
  nConfigurationId         : UINT;
END_VAR
VAR_OUTPUT
  bBusy                    : BOOL;
  bError                   : BOOL;
  nErrorId                 : ETcsSpeechCommandExitCode;
  eRecognitionEngineState  : ETcsRecognitionEngineState;
  nRecognitionId           : ULINT;
  fRecognitionConfidence   : REAL;
  sRecognitionRule         : STRING(255);
  sRecognitionTag          : STRING(255);
  sRecognitionUtterance    : STRING(4095);
### Inputs

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bListen</td>
<td>BOOL</td>
<td>Rising edge triggers listening.</td>
</tr>
<tr>
<td>nConfigurationId</td>
<td>UINT</td>
<td>ConfigurationId of configuration to be used in listen request.</td>
</tr>
</tbody>
</table>

### Outputs

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td>Function block is busy.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>Function block is in error state.</td>
</tr>
<tr>
<td>nErrorId</td>
<td>ETcsSpeechCommandExitCode</td>
<td>ErrorId providing details of the occurred error.</td>
</tr>
<tr>
<td>eRecognitionEState</td>
<td>ETcsRecognitionEngineState</td>
<td>Current state of the recognition engine.</td>
</tr>
<tr>
<td>nRecognitionId</td>
<td>ULINT</td>
<td>Last speech recognition (RecognitionId)</td>
</tr>
<tr>
<td>fRecognitionConfidence</td>
<td>REAL</td>
<td>Last recognition's confidence level.</td>
</tr>
<tr>
<td>nLastCommandExitCode</td>
<td>UINT</td>
<td>Returncode of last executed command.</td>
</tr>
<tr>
<td>sRecognitionTag</td>
<td>STRING(255)</td>
<td>Last recognition's tag.</td>
</tr>
<tr>
<td>sRecognitionRule</td>
<td>STRING(255)</td>
<td>Last recognition's triggered rule.</td>
</tr>
<tr>
<td>sRecognitionUtterance</td>
<td>STRING(4095)</td>
<td>Last recognition's utterance.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetAmsNetAddr</td>
<td>Method to be called once to specify the AmsNetId of destination TcSpeech service.</td>
</tr>
</tbody>
</table>

**Required License**

TC3 Speech

**9.2.4.1 SetAmsNetAddr**

Method to be called once to specify the AmsNetId of destination TcSpeech service.
Syntax

Definition:

METHOD SetAmsNetAddr : HRESULT
VAR_INPUT
   sAmsNetId : STRING;
END_VAR

Inputs

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sAmsNetId</td>
<td>STRING</td>
<td>AmsNetId of destination TcSpeech service.</td>
</tr>
</tbody>
</table>

Return value

HRESULT

9.2.5 FB_TextToSpeech

Text to speech function block.

Syntax

Definition:

FUNCTION_BLOCK FB_TextToSpeech
VAR_INPUT
   bSpeak       : BOOL;
   sUtterance   : STRING;
   nConfigurationId : UINT;
   nLanguageId  : UINT;
END_VAR
VAR_OUTPUT
   bBusy       : BOOL;
   bError      : BOOL;
   nErrorId    : ETcsSpeechCommandExitCode;
   nLastCommandExitCode : UINT;
   nPlaybackPosition : UUINT;
   nPlaybackTotal  : UUINT;
END_VAR

Inputs

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bSpeak</td>
<td>BOOL</td>
<td>Rising edge triggers speak command.</td>
</tr>
<tr>
<td>sUtterance</td>
<td>STRING</td>
<td>Utterance to speak.</td>
</tr>
<tr>
<td>nConfigurationId</td>
<td>UINT</td>
<td>ConfigurationId to be used (cmp. TcSpeech Configurator).</td>
</tr>
<tr>
<td>nLanguageId</td>
<td>UINT</td>
<td>LanguageId to be used in speech synthesis request. '0' defaults to DefaultLanguageId specified in configuration.</td>
</tr>
</tbody>
</table>
## Outputs

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td>Function block is busy.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>Function block is in error state.</td>
</tr>
<tr>
<td>nErrorId</td>
<td>ETcsSpeechCommandExitCode [56]</td>
<td>ErrorId providing details of the occurred error.</td>
</tr>
<tr>
<td>nLastCommandExitCode</td>
<td>UINT</td>
<td>Returncode of last executed command.</td>
</tr>
<tr>
<td>nPlaybackPosition</td>
<td>ULINT</td>
<td>Current playback position in synthesis playback (milliseconds).</td>
</tr>
<tr>
<td>nPlaybackTotal</td>
<td>ULINT</td>
<td>Total length of synthesis playback (milliseconds).</td>
</tr>
</tbody>
</table>

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetAmsNetAddr</td>
<td>Method to be called once to specify the AmsNetId of destination TcSpeech service.</td>
</tr>
</tbody>
</table>

### Required License

TC3 Speech

### 9.2.5.1 SetAmsNetAddr

Method to be called once to specify the AmsNetId of destination TcSpeech service.

**Syntax**

**Definition:**

```c
METHOD SetAmsNetAddr : HRESULT
VAR_INPUT
  sAmsNetId : STRING;
END_VAR
```

### Inputs

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sAmsNetId</td>
<td>STRING</td>
<td>AmsNetId of destination TcSpeech service.</td>
</tr>
</tbody>
</table>

### Return value

HRESULT
10  Samples

You will find several examples in this chapter. The samples are independent of each other. Each sample has a focus, based on which it is listed below.

Commissioning the samples

In principle, however, opening and configuration work the same way with all samples. The procedure is explained below on the basis of an example.

1. Unpack the downloaded ZIP file.
2. Open the zip file that it contains in TwinCAT 3 by clicking on Open Project.
3. Select your target system.
4. Activate your configuration on the TwinCAT Speech Configuration Solution by clicking .
5. Activate the configuration by clicking on . ☰ The sample is ready for operation.

10.1  Sample 01: Simple ASR and TTS sample

| Description | Example of PLC programming for ASR and TTS
Two simple programs show how to handle FB_TextToSpeech or FB_SpeechRecognition.
In the configuration for the speech recognition, the text "Hello Twincat" is recognized, transmitted to the PLC and evaluated by the MainASR program. Successful recognition is displayed for one second by "bRecognition := TRUE".
For the speech output, a greeting word is output as soon as the variable "bSpeak" in MainTTS is set to TRUE.
With all TwinCAT Speech samples, a configuration is stored that uses the operating system’s default sound card.
Note Activate the TwinCAT Speech configuration. |
| Sample project | https://infosys.beckhoff.com/content/1033/TF4500_TC3_TwinCAT_Speech/Resources/zip/10037207819.zip |
| Further information | Quick start: speech output (TTS) [p. 27] Quick start: speech input (ASR) [p. 17] |
10.2 Sample 02: Complex ASR

| Description | The sample illustrates the use of an SRGS that wasn't created with the Simple SRGS Editor. Two SRGS files are configured in the TwinCAT Speech configurator for this. After starting the PLC, set the variable MainASR.bListenTrigger to TRUE. This starts the speech recognition. These commands are then recognized and saved in nValue. Set value to <Number> Increment value by <Number> Decrement value by <Number> Note: the sample is based on a file named en-US.grxml from Microsoft, which you can download from https://www.microsoft.com/en-us/download/confirmation.aspx?id=14373. Then add the file to the PLC project in the subfolder SRGS using "Add existing item". The file is transferred to the target system on activating the TwinCAT Speech configuration. With all TwinCAT Speech samples, a configuration is stored that uses the operating system's default sound card. |

| Sample project | https://infosys.beckhoff.com/content/1033/TF4500_TC3_TwinCAT_Speech/Resources/zip/10037206155.zip |

| Further information | Configuring a grammar file |}

10.3 Sample 03: EventLogger

| Description | This sample illustrates the integration of the TwinCAT EventLogger with TwinCAT Speech. An event class "EventSpeechTestClass" is defined in the sample. This has 2 events, one for triggering on recognizing speech and one with which the speech output takes place. The PLC program is only used to send the corresponding event to the speech output. An event is triggered if the speech recognition recognizes "Hello TwinCAT". You control the TwinCAT events as usual in the "Logged Events" window. With all TwinCAT Speech samples, a configuration is stored that uses the operating system's default sound card. |

| Sample project | https://infosys.beckhoff.com/content/1033/TF4500_TC3_TwinCAT_Speech/Resources/zip/10037209483.zip |

| Further information | Quick start: EventLogger |
## 10.4 Sample 04: File Playback

| Description | This sample shows you how to play back a simple file by means of FB_Play.  
|             | The file to be played back is transferred as a file path. A sample file called 16bit_16khz_mono_SampleSound.wav is located in the PLC project and must be placed on the target system beforehand with the correct path.  
|             | With all TwinCAT Speech samples, a configuration is stored that uses the operating system's default sound card.  
|             | **Note** Activate the TwinCAT Speech configuration. |

### Sample project

[https://infosys.beckhoff.com/content/1033/TF4500_TC3_TwinCAT_Speech/Resources/zip/10037211147.zip](https://infosys.beckhoff.com/content/1033/TF4500_TC3_TwinCAT_Speech/Resources/zip/10037211147.zip)

### Further information

- FB_Play

Also see about this

- FB_Play

## 10.5 Sample 05: Activation/deactivation of individual recognition tags

| Description | This sample shows how individual recognition tags, which are stored as rules in the SRGS, can be enabled or disabled at runtime.  
|             | This can increase the recognition accuracy in particular with similar voice commands.  
|             | The SRGS contains a recognition tag "HELLO". This is enabled or disabled by the FB_SetRuleState in the PLC program depending on bSetRuleState.  
|             | Recognitions are displayed in bRecognition.  
|             | With all TwinCAT Speech samples, a configuration is stored that uses the operating system's default sound card.  
|             | **Note** Activate the TwinCAT Speech configuration. |

### Sample project

[https://infosys.beckhoff.com/content/1033/TF4500_TC3_TwinCAT_Speech/Resources/zip/10037204491.zip](https://infosys.beckhoff.com/content/1033/TF4500_TC3_TwinCAT_Speech/Resources/zip/10037204491.zip)

### Further information

- FB_SetRuleState
## 10.6 Sample 06: TwinCAT HMI with TwinCAT Speech Integration

| Description | This sample shows how TwinCAT Speech can be used in TwinCAT HMI. For this purpose, a TwinCAT HMI server is entered in the TwinCAT Speech configuration. The speech recognition and speech output then take place via the HMI client, i.e. the browser, for which the latter requests and needs the corresponding access rights.

The TwinCAT HMI project included offers some voice commands that are displayed in the HMI itself. |
| Sample project | https://infosys.beckhoff.com/content/1033/TF4500_TC3_TwinCAT_Speech/Resources/zip/10037212811.zip |
| Further information | TwinCAT Speech of the TwinCAT HMI |
11 FAQ

Sound card is not working

Probably the wrong device was configured.

1. Check the Configuration Id and assignment to the sound card
2. Take advantage of the testing possibilities of the TwinCAT Speech Configurator project
3. If in doubt, check the function of the sound card at the operating system level. The TwinCAT Speech Configurator displays all sound cards that the operating system of the target system has also detected.

CX2500-0020 (CX2000 extension module)

4. Enable the BUS used by the BIOS module:
   Chipset->PCH I-Configuration->PCH Azalia Configuration->Azalia -> “Enabled”

Too many false recognitions

TwinCAT Speech provides the PLC with all voice commands that have been recognized.

The PLC project must then filter by confidence thresholds.

The sample suggests filtering on two levels:

1. Main.fConfidenceThreshold:
   ◦ Confidence levels that are too low are directly discarded.
2. FB_RetrieveUtterance.fRecoThreshold:
   ◦ searches for repetitions.

XAE error message

Error messages are displayed in the TwinCAT XAE Error window.

An additional and detailed view of errors can be found in the log file:

TcSpeechLog_0.txt under C:\TwinCAT\Functions\TF4500-Speech\Boot
12 Appendix

12.1 Error List

The following errors are returned by the function blocks.
<table>
<thead>
<tr>
<th>Hex</th>
<th>Dec</th>
<th>Description</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>SUCCEEDED</td>
<td>The call was successful</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>GENERIC_ERROR</td>
<td>The call ended in a generic error that is not specified in more detail. Details in the LogFile.</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>ASR_NO_RECOGNITION</td>
<td>The ASR did not find any recognition.</td>
</tr>
<tr>
<td>200</td>
<td></td>
<td>TTS_SYNTH_WARNING_VOICENOTFOUND</td>
<td>The configured language could not be used.</td>
</tr>
<tr>
<td>1000</td>
<td></td>
<td>REQUIRED_SERVICE_NOT_FOUND</td>
<td>TwinCAT Speech internal: A required service could not be found.</td>
</tr>
<tr>
<td>1001</td>
<td></td>
<td>REQUIRED_SERVICE_NOT_INITIALIZED</td>
<td>TwinCAT Speech internal: A required service is not initialized.</td>
</tr>
<tr>
<td>1010</td>
<td></td>
<td>INSUFFICIENT_PARAMETERS</td>
<td>TwinCAT Speech internal: A call was made with incorrect parameters</td>
</tr>
<tr>
<td>2000</td>
<td></td>
<td>ASR_RULES_SETFAILED</td>
<td>The ASR was unable to activate the rules created in the SRGS.</td>
</tr>
<tr>
<td>20010</td>
<td></td>
<td>ASR_INSTANTIATION_FAILURE</td>
<td>An ASR could not be initialized. There may be a configuration error.</td>
</tr>
<tr>
<td>20020</td>
<td></td>
<td>ASR_STOP_FAILURE</td>
<td>An ASR could not be stopped.</td>
</tr>
<tr>
<td>20030</td>
<td></td>
<td>ASR_START_FAILURE</td>
<td>An ASR could not be started.</td>
</tr>
<tr>
<td>30010</td>
<td></td>
<td>TTS_STATUS_REQUEST_FAILURE</td>
<td>A TTS call failed.</td>
</tr>
<tr>
<td>30020</td>
<td></td>
<td>TTS_STOP_REQUEST_FAILURE</td>
<td>A TTS call could not be interrupted.</td>
</tr>
<tr>
<td>30030</td>
<td></td>
<td>TTS_SYNTH_ERROR</td>
<td>A TTS call could not be synthesized.</td>
</tr>
<tr>
<td>30040</td>
<td></td>
<td>TTS_SYNTH_ERROR_CONFIGURATION_DETAILS</td>
<td>A TTS call could not be synthesized because a configuration is faulty.</td>
</tr>
<tr>
<td>30050</td>
<td></td>
<td>TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_LANGUAGEID_MISSING</td>
<td>A TTS call could not be synthesized because the configuration includes an incorrect language ID.</td>
</tr>
<tr>
<td>30060</td>
<td></td>
<td>TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_LANGUAGEID_NOTSUPPORTED</td>
<td>A TTS call could not be synthesized because the configuration includes an unsupported language ID.</td>
</tr>
<tr>
<td>30070</td>
<td></td>
<td>TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_INCOMPLETE</td>
<td>A TTS call could not be synthesized because the configuration is incomplete.</td>
</tr>
<tr>
<td>30080</td>
<td></td>
<td>TTS_SYNTH_ERROR_UTTERANCE_EMPTY</td>
<td>A TTS call could not be synthesized because the input is empty.</td>
</tr>
<tr>
<td>31000</td>
<td></td>
<td>TTS_SYNTH_ERROR_SAPI</td>
<td>A TTS call could not be synthesized because the underlying Microsoft SAPI reported an error. Details in the LogFile</td>
</tr>
<tr>
<td>32000</td>
<td></td>
<td>TTS_SYNTH_ERROR_AWSPOLLY</td>
<td>A TTS call could not be synthesized because the underlying AWS Polly interface reported an error. Details in the LogFile</td>
</tr>
<tr>
<td>40010</td>
<td></td>
<td>AUDIO_PLAYBACK_START_FAILURE</td>
<td>Playback could not be started. Another playback may have just started.</td>
</tr>
</tbody>
</table>

### 12.2 Third-party Components

This software contains third-party components (OSS components). Please refer to the license file provided in the following folder for further information: C:\TwinCAT\Functions\TF4500-Speech\Licenses
12.3 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff’s branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff’s branch offices and representatives round the world can be found on her internet pages: https://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

• support
• design, programming and commissioning of complex automation systems
• and extensive training program for Beckhoff system components

Hotline: +49 5246 963 157
Fax: +49 5246 963 9157
e-mail: support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

• on-site service
• repair service
• spare parts service
• hotline service

Hotline: +49 5246 963 460
Fax: +49 5246 963 479
e-mail: service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG
Huelshorstweg 20
33415 Verl
Germany

Phone: +49 5246 963 0
Fax: +49 5246 963 198
e-mail: info@beckhoff.com
web: https://www.beckhoff.com
More Information:
www.beckhoff.com/tf4500