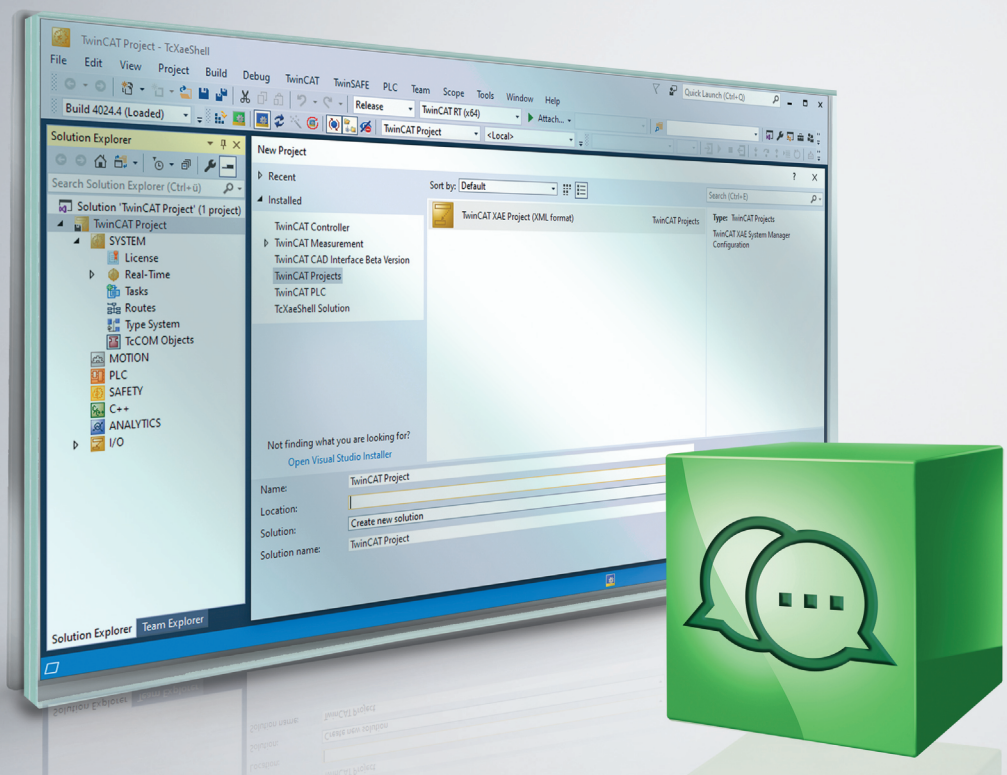
BECKHOFF New Automation Technology

Manual | EN

TF4500

TwinCAT 3 | Speech



2021-12-14 | Version: 1.5

Table of contents

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:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

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

Copyright

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

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

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

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

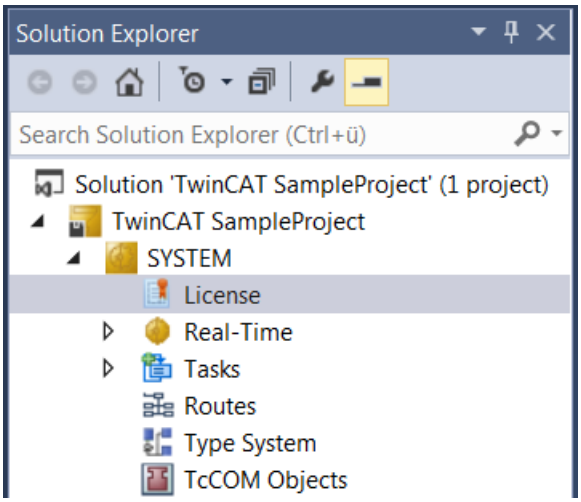
A description of the procedure to license a full version can be found in the Beckhoff Information System in the documentation "TwinCAT 3 Licensing".

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.

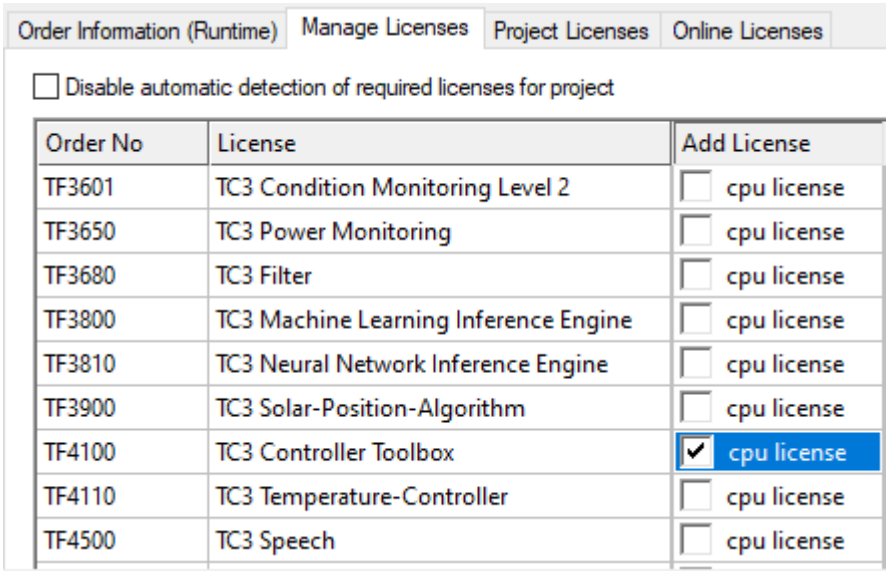
1. Start the TwinCAT 3 development environment (XAE).
2. Open an existing TwinCAT 3 project or create a new project.
3. If you want to activate the license for a remote device, set the desired target system. To do this, select the target system from the **Choose Target System** drop-down list in the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



⇨ 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").

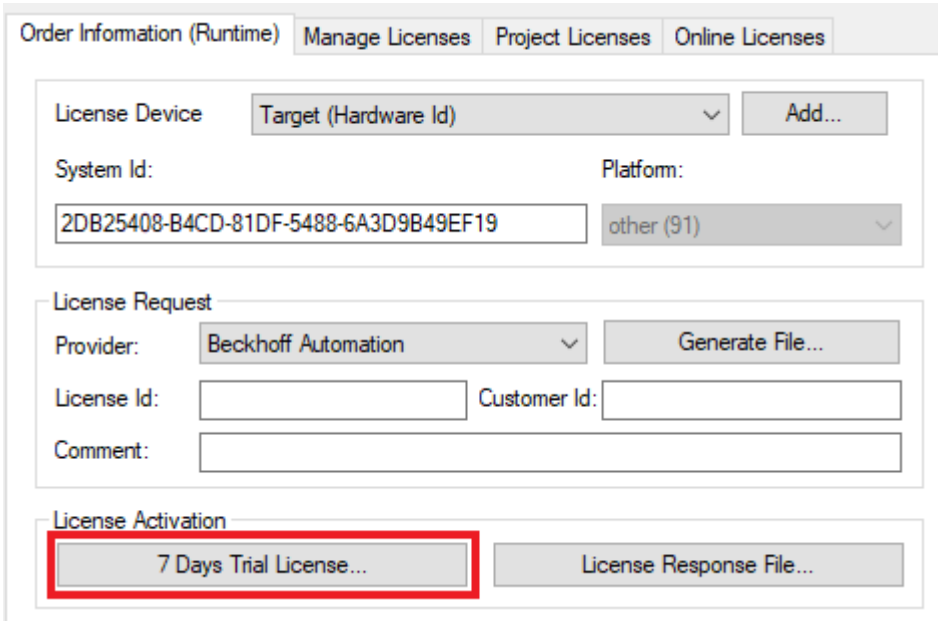


Order No	License	Add License
TF3601	TC3 Condition Monitoring Level 2	cpu license
TF3650	TC3 Power Monitoring	cpu license
TF3680	TC3 Filter	cpu license
TF3800	TC3 Machine Learning Inference Engine	cpu license
TF3810	TC3 Neural Network Inference Engine	cpu license
TF3900	TC3 Solar-Position-Algorithm	cpu license
TF4100	TC3 Controller Toolbox	cpu license
TF4110	TC3 Temperature-Controller	cpu license
TF4500	TC3 Speech	cpu license

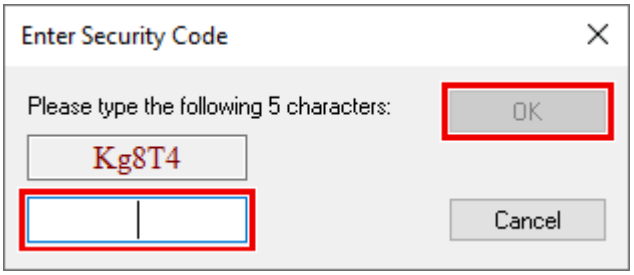
6. Open the **Order Information (Runtime)** tab.

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

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.

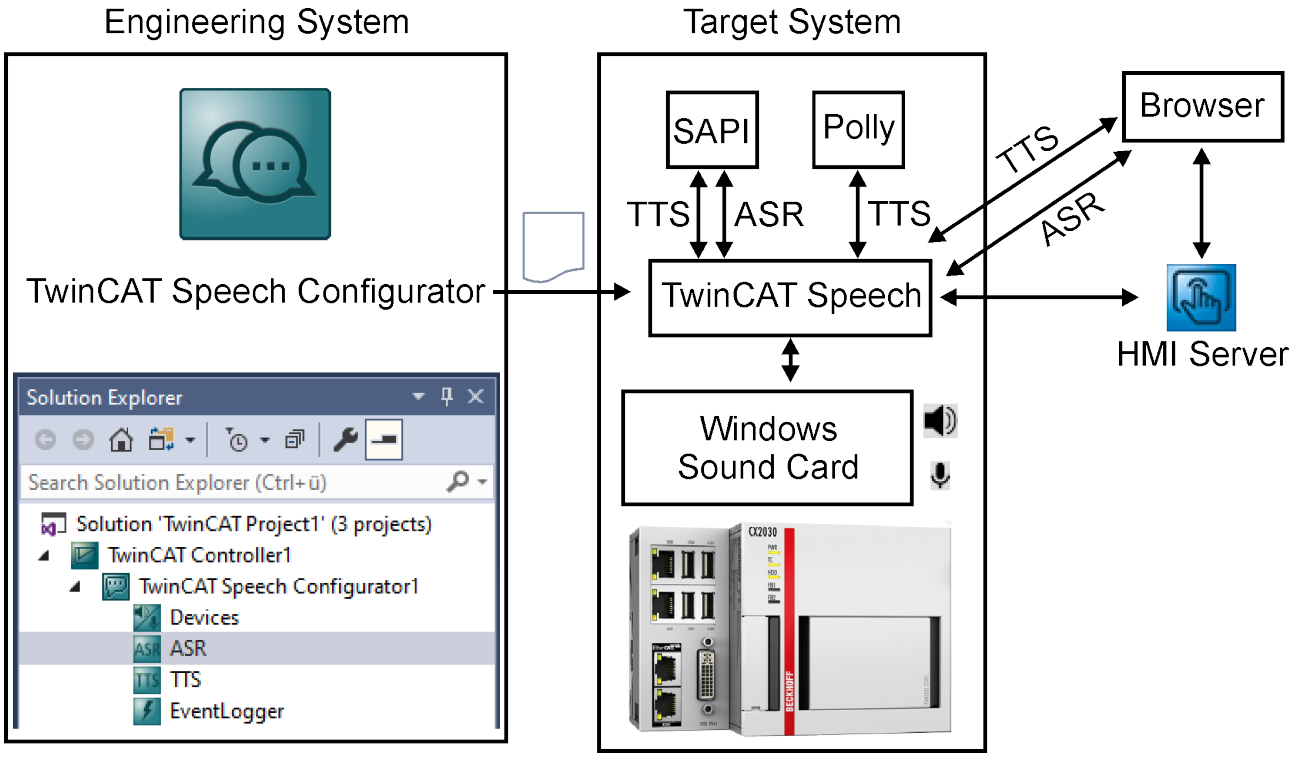
4 Technical introduction

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:



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)

Language	LangID	Microsoft SAPI – ASR
German	1031	de-DE
English	1033	en-US
	2057	en-GB
French	1036	fr-FR
Japanese	1041	ja-JP
Spanish	1034	es-ES
Chinese	2052	zh-cn
	3076	zh-hk

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

Microsoft SAPI - speech output (TTS)

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

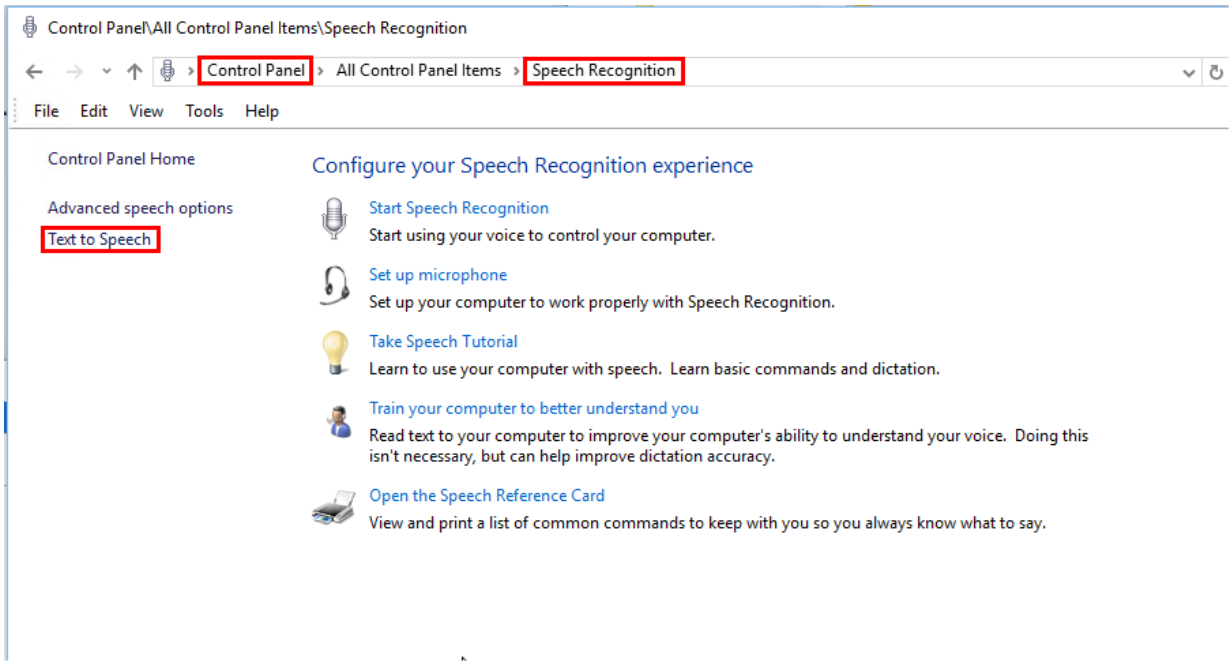
Please note that the appropriate Windows language packs must be installed in order to be used. This is described here [▶ 55].

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

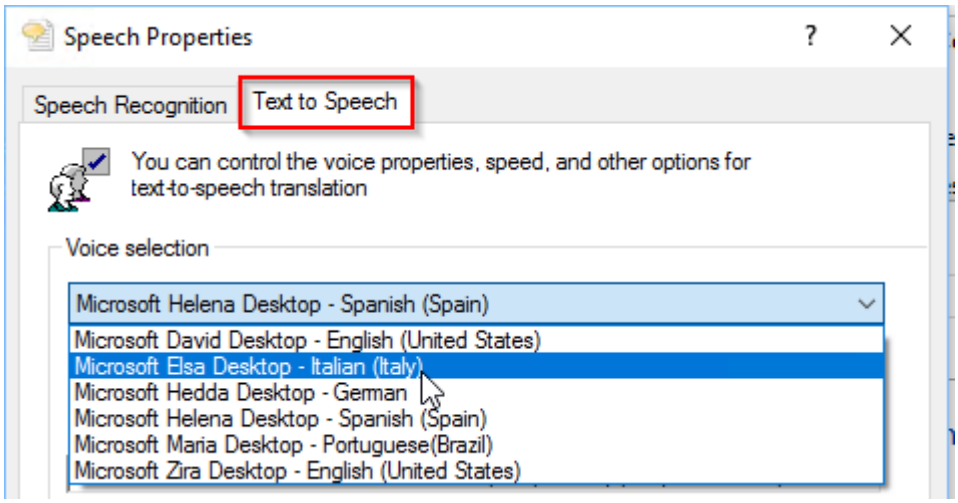
The basic compatibility can be checked as follows.

1. Install the desired language.

2. Open the **Text-to-Speech** menu from the "Control Panel".



⇨ If languages are displayed here, they should also be usable with TwinCAT Speech.



Amazon Polly - speech output (TTS)

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

Further information on Amazon Polly can be found here: 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.

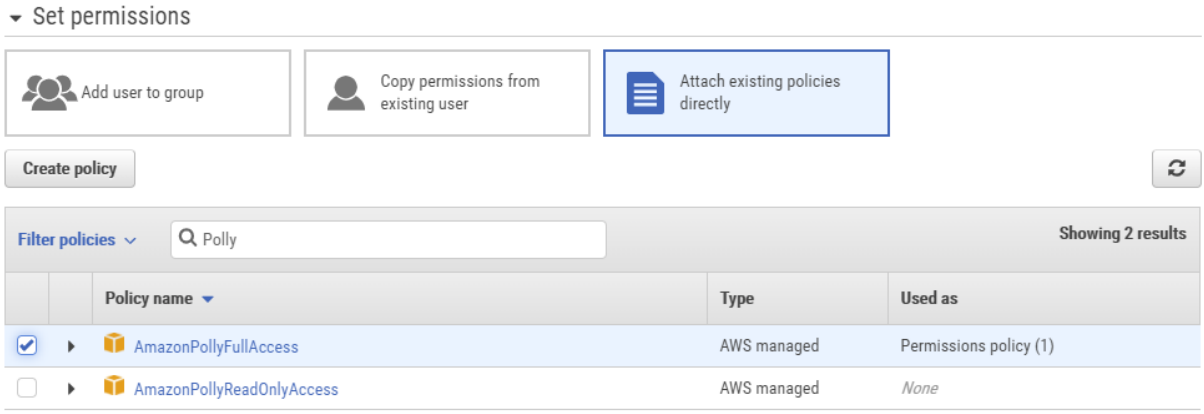
If you cannot select the desired language, you must install it; see Installing additional languages.

4.1.2 Setting Amazon Polly

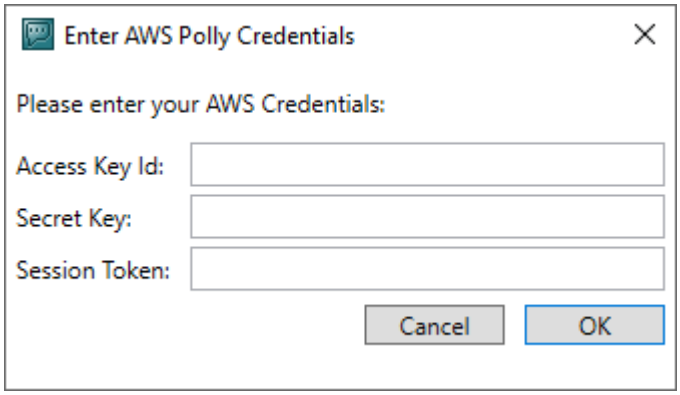
The speech synthesis service Polly works on the basis of a cloud system and each query placed via Polly costs money. For this reason, you can only use Polly with an active Internet connection and an AWS account.

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

1. Create an account on the website www.aws.amazon.com/de/polly.
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)

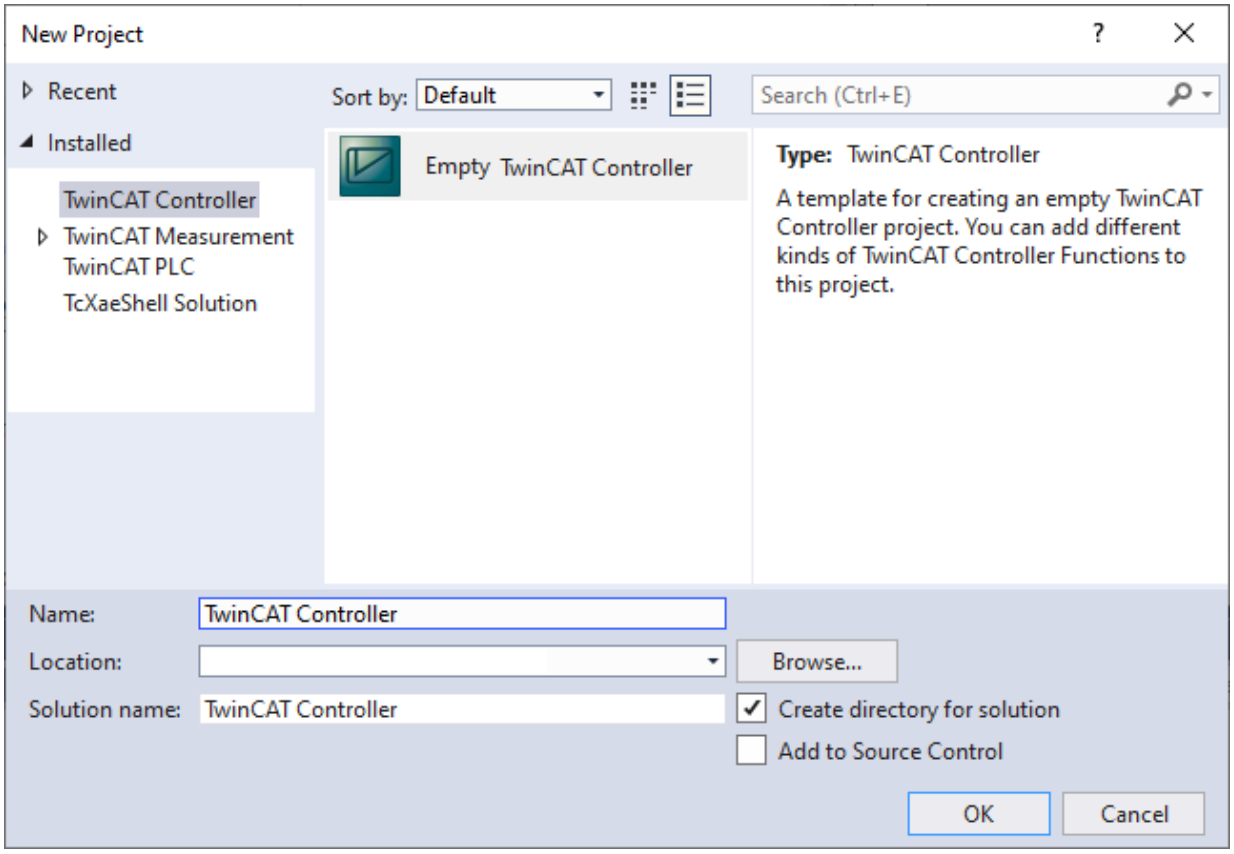
The following chapter is intended to simplify the configuration of TwinCAT Speech for speech input.

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:

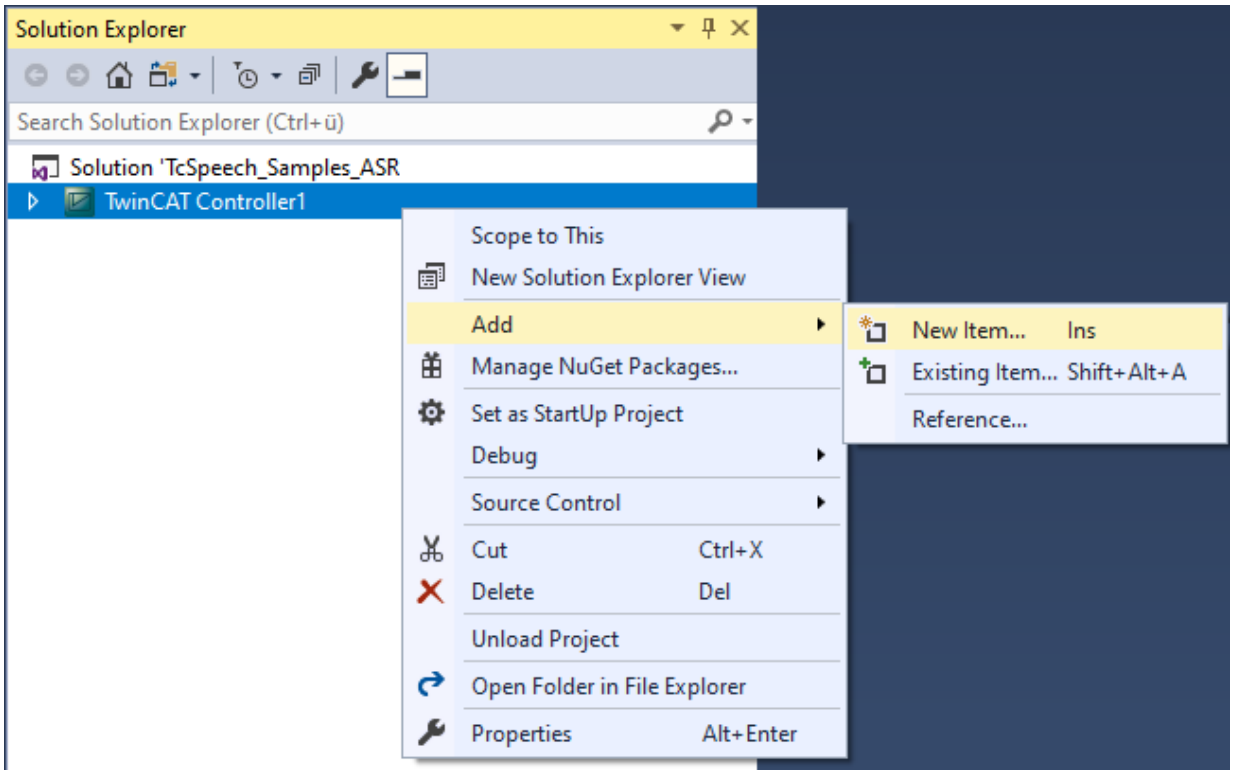
1. Open TwinCAT Engineering and create a new project.
2. Select an empty TwinCAT Controller project, name it, and click **OK**.



⇨ The new TwinCAT Controller project opens.

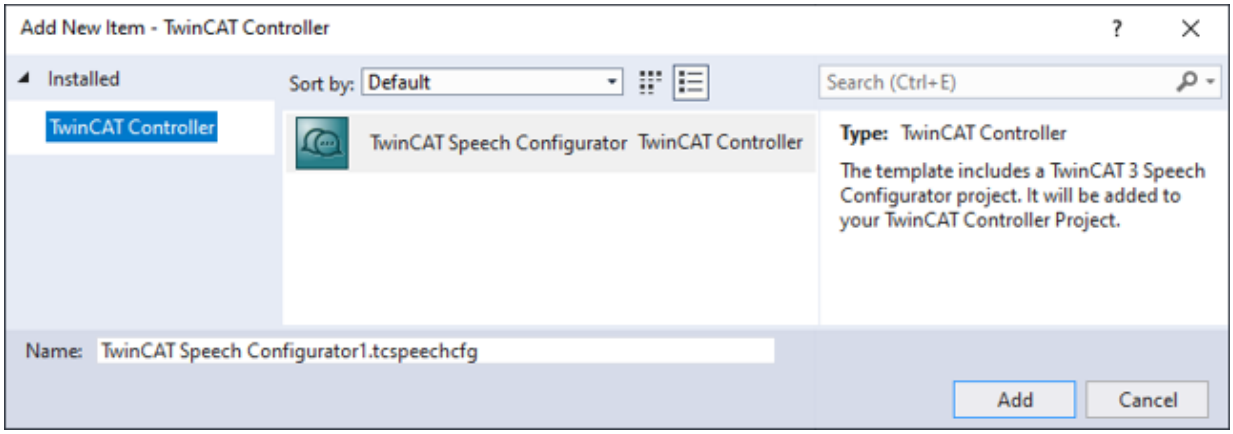
3. Right-click the new TwinCAT Controller project.

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.

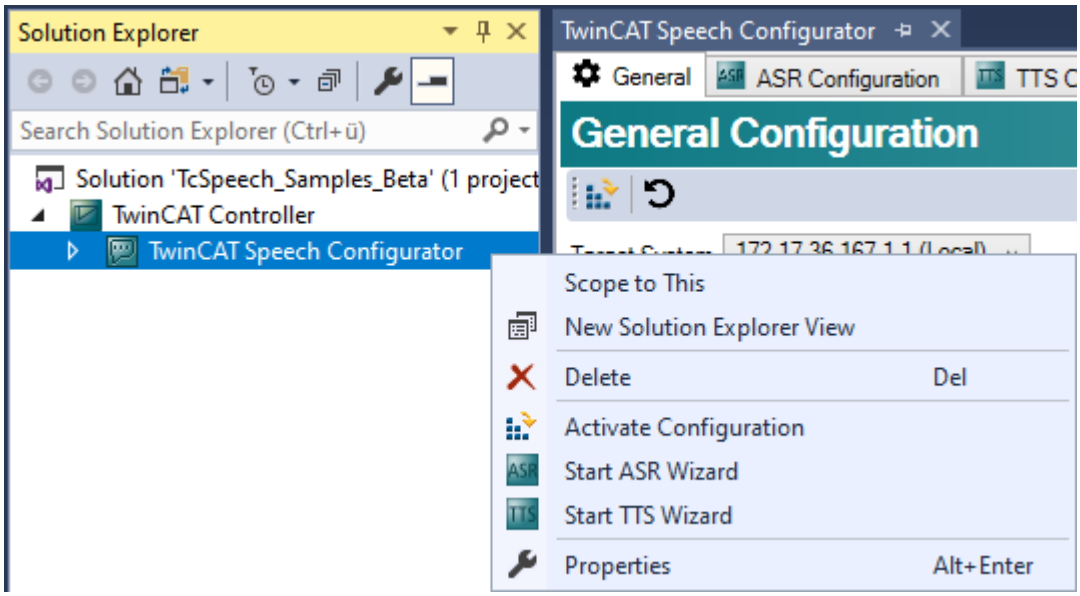
⇨ Assignment to a target system is important for TwinCAT Speech, because the hardware configuration of the sound cards is individual.

⇨ A new TwinCAT Speech configuration has now been created.

5.2 Configuring speech input

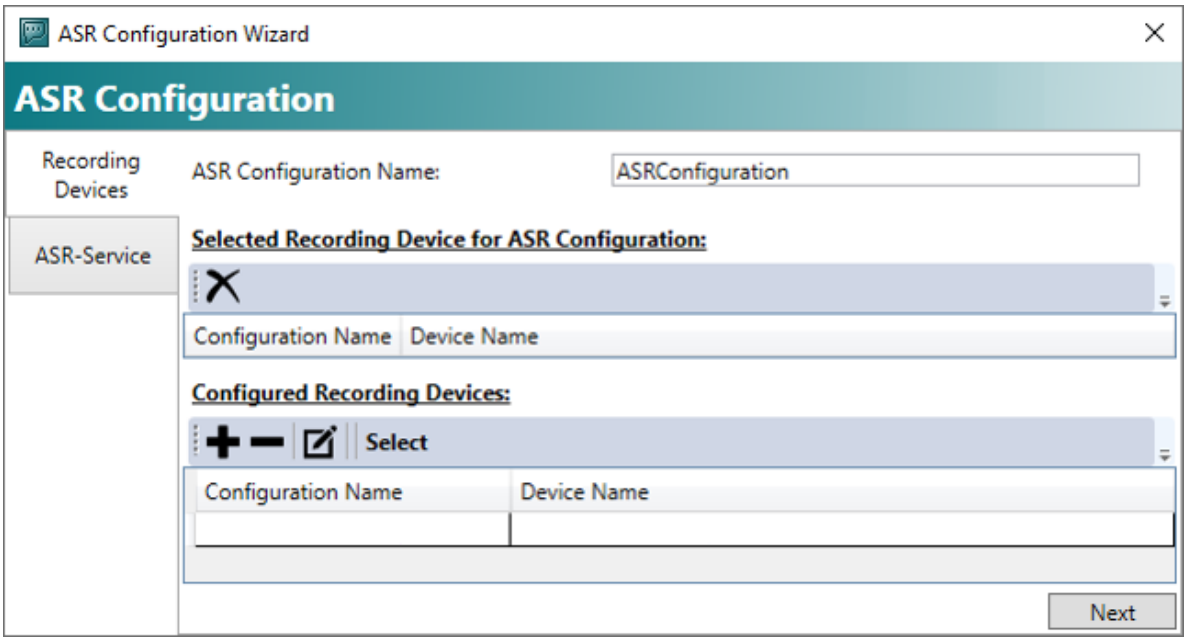
A configuration wizard is available for the configuration of the speech input (ASR). This guides you through the ASR configuration.

1. To open the ASR Configuration Wizard, right click TwinCAT Speech Configurator in the Solution Explorer.



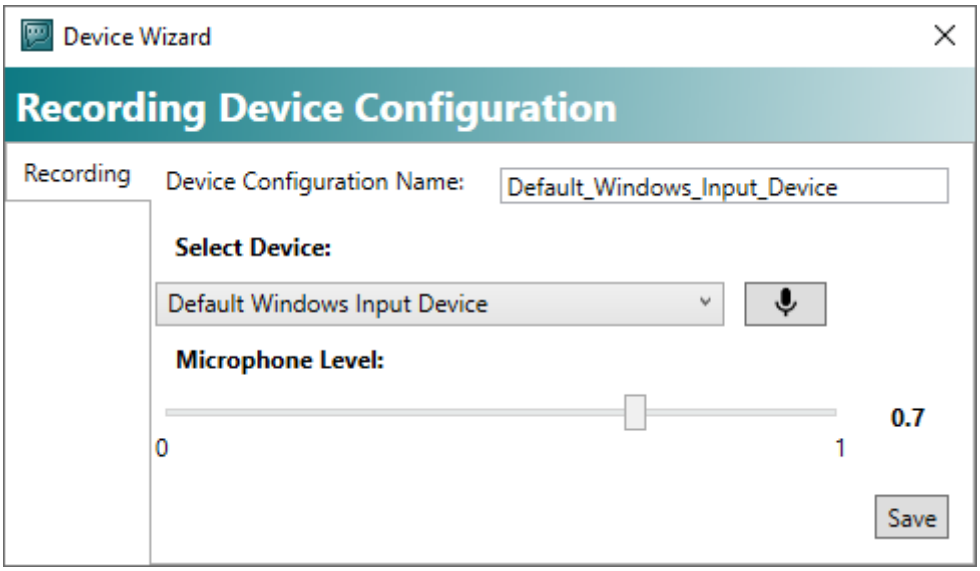
2. Select **Start ASR Wizard** from the context menu.

   ⇨ 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.



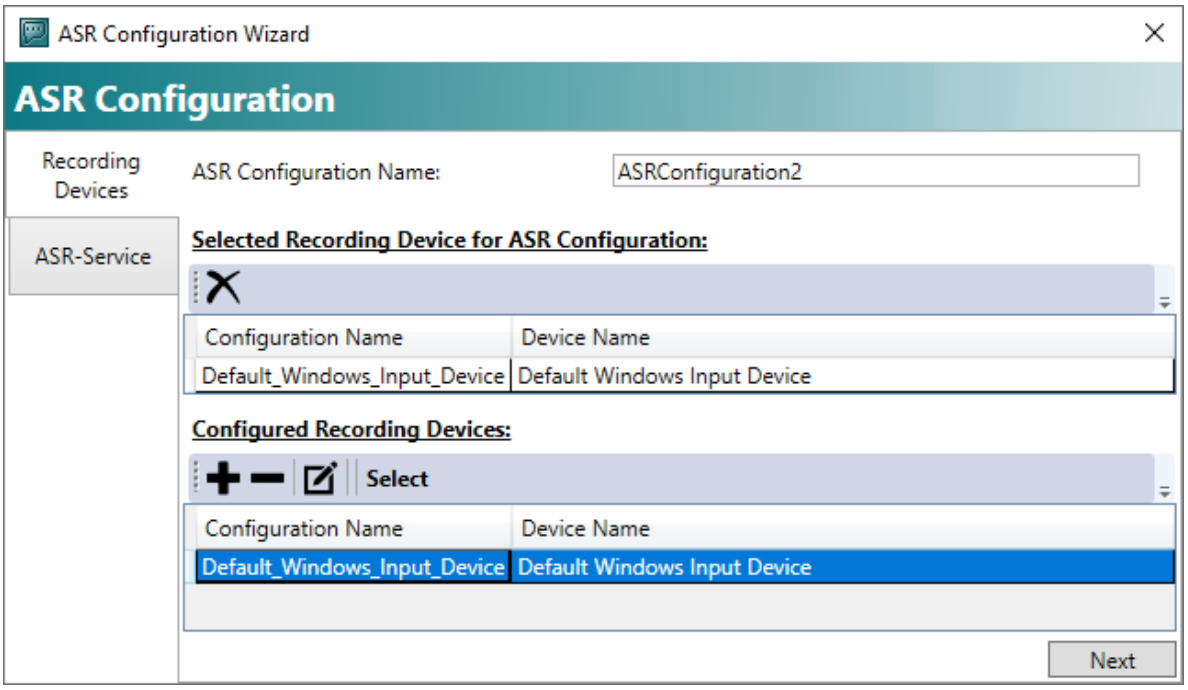
3. To add a new microphone input, click the **+** button.

⇨ The Device Wizard opens.



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.

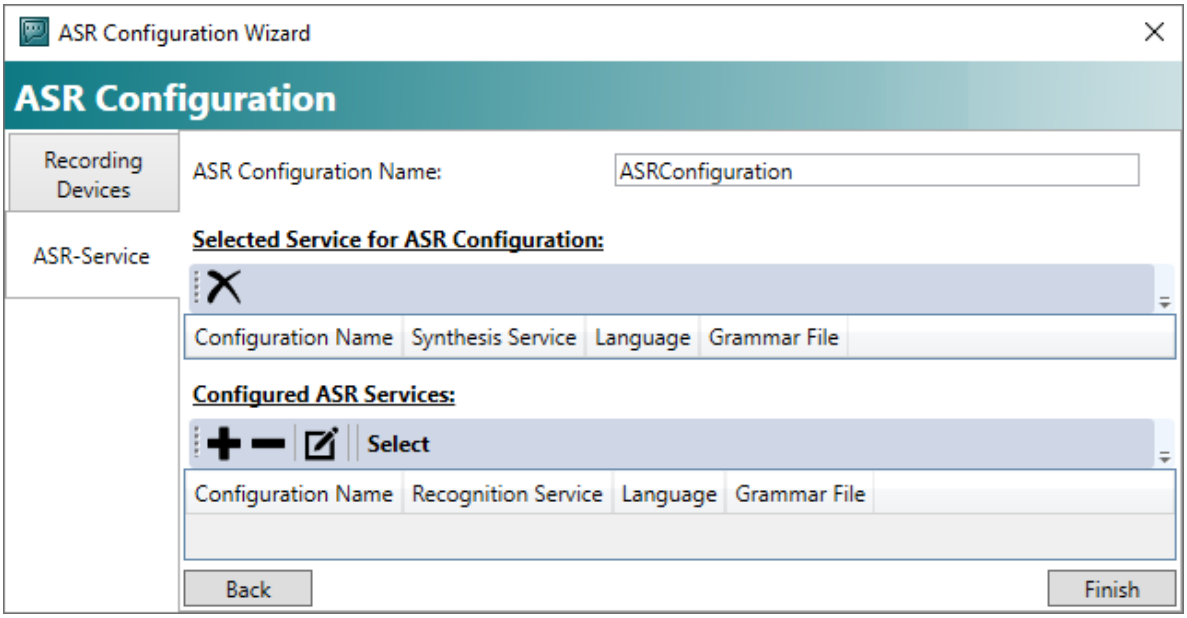
   ⇨ The Device Wizard closes.



6. If necessary, mark the newly created microphone input and click **Select**.

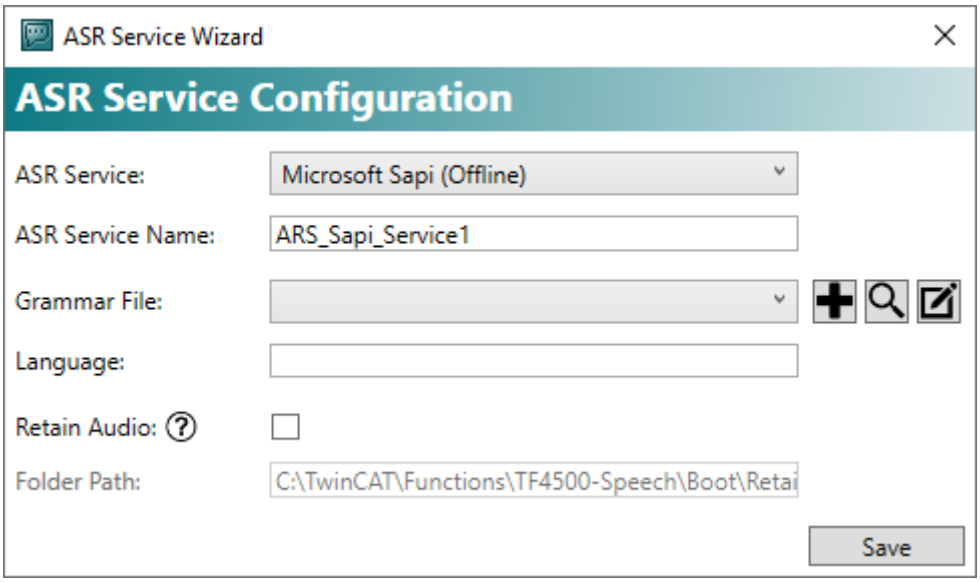
   ⇨ The microphone input is added to the list **Selected Device for ASR**.
7. Click the **Next** button.

⇨ 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**.



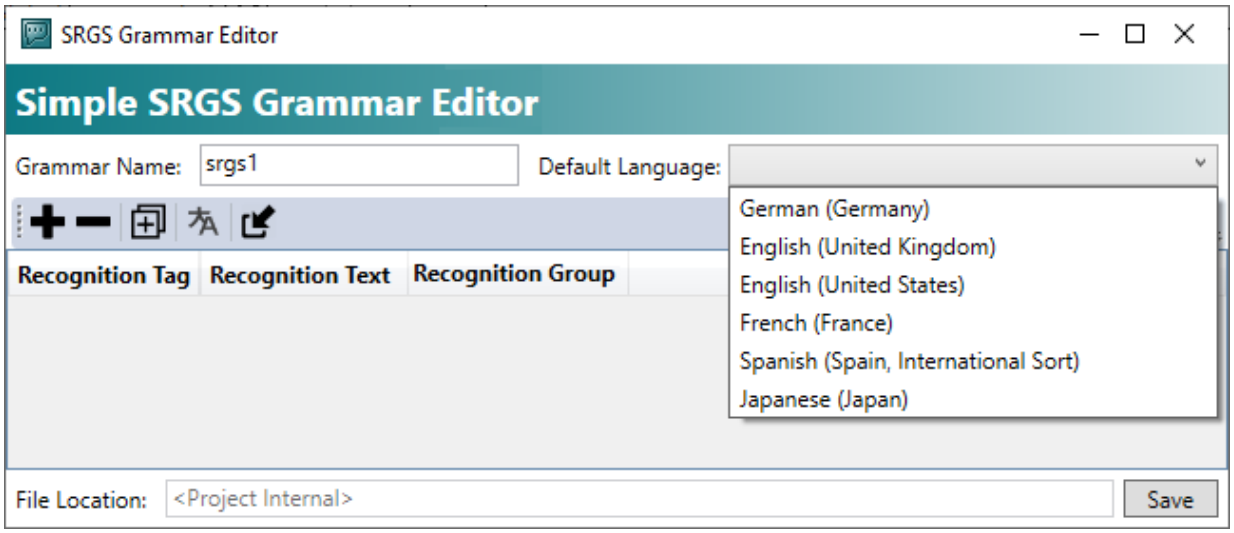
8. To add a speech recognition service, click the **+** button.

⇨ 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**.

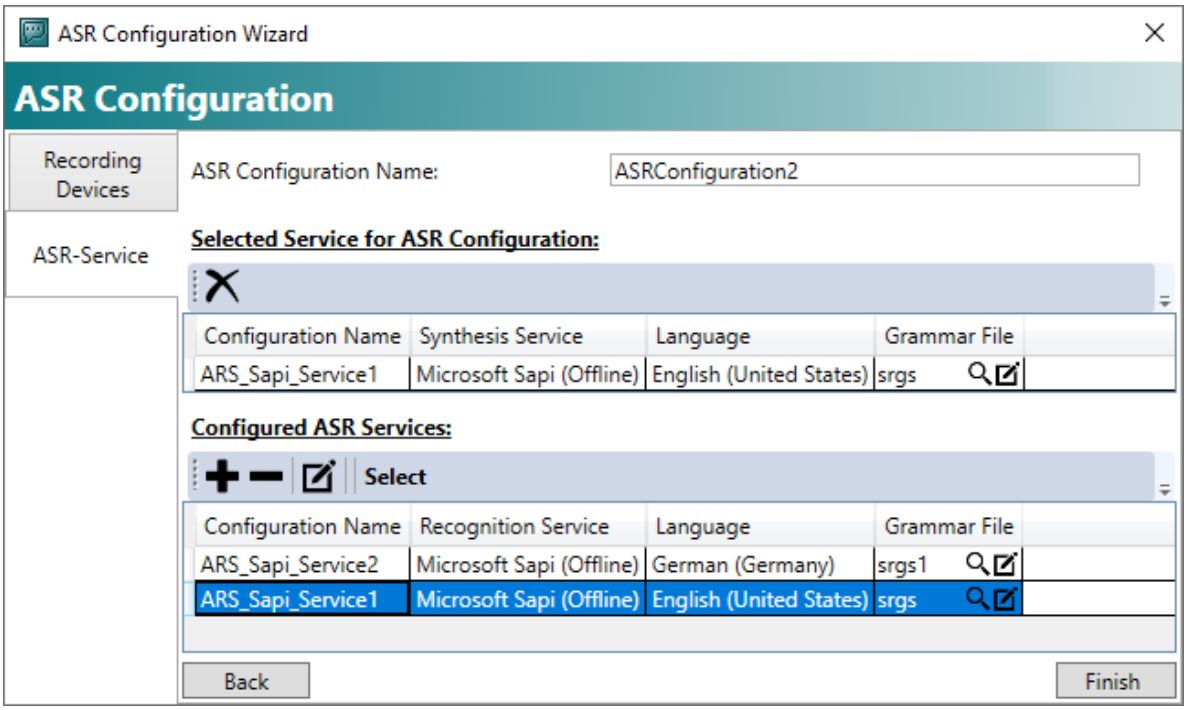


15. Name the file in the **Grammar Name** input field.
16. Select the desired default language from the **Language** drop-down list.
    You can store several languages in the SRGS Editor and switch between them later, for example from the PLC or HMI.
17. Click the **+** button.
    ⇨ A new row is added.
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.

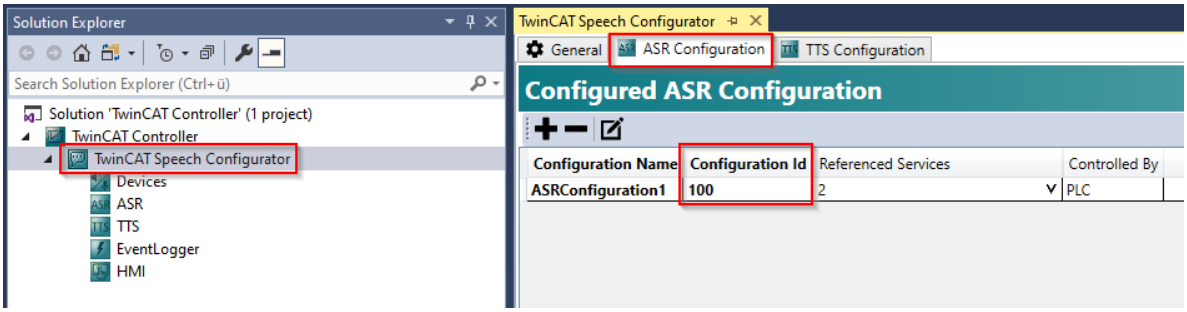


22. If speech recognition services have already been created, select the newly created speech recognition service and click **Select**.

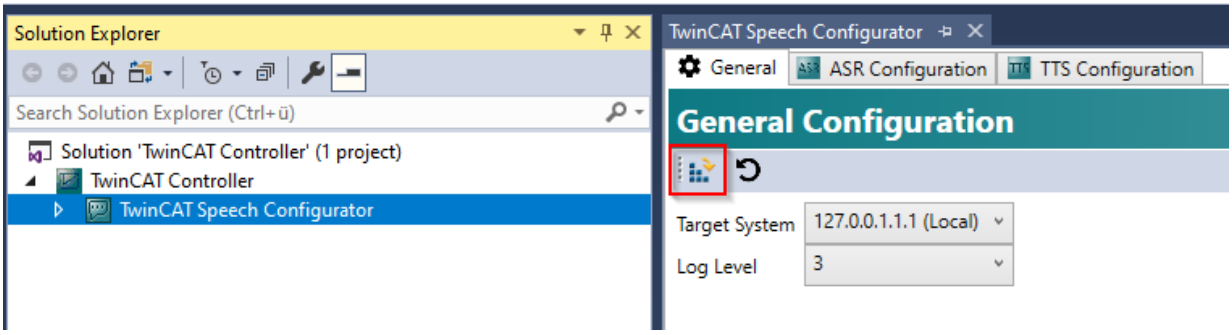
⇨ The speech recognition service is added to the list in the section **Selected Service for ASR Configuration**.

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.



24. Activate the TwinCAT Speech configuration on the target system.

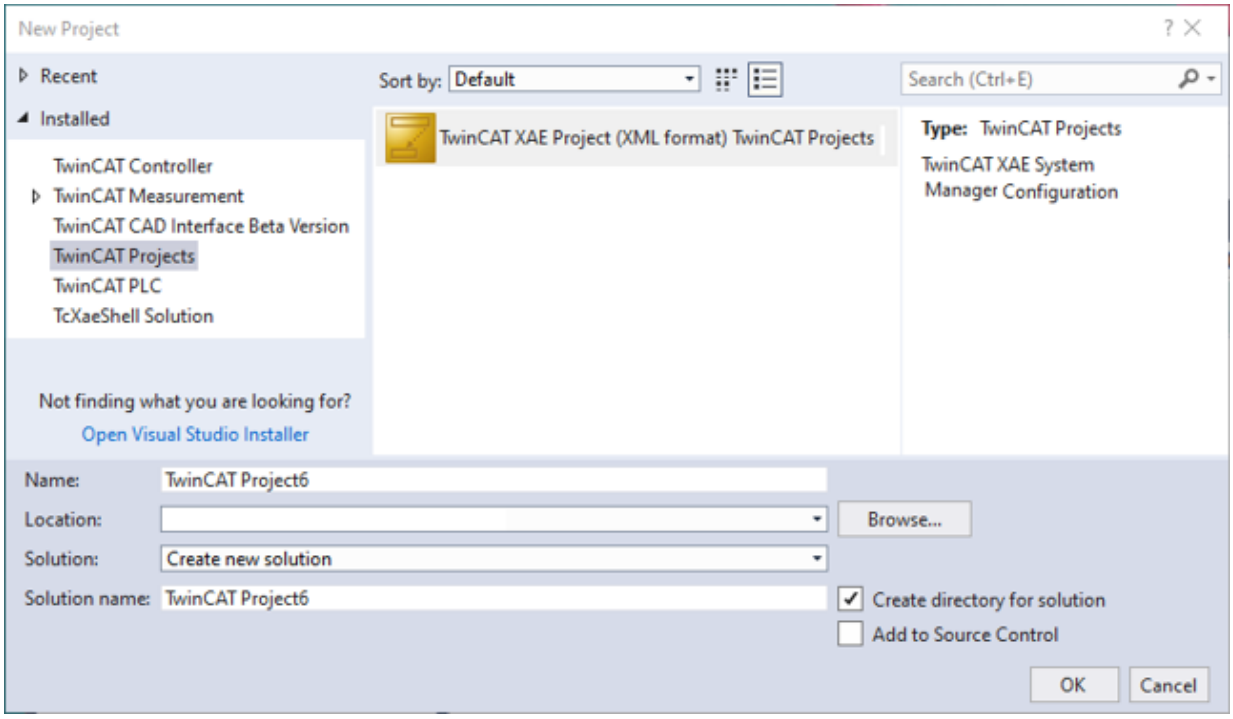


⇨ The TwinCAT Speech configuration is activated on the target system and can be used by the PLC.

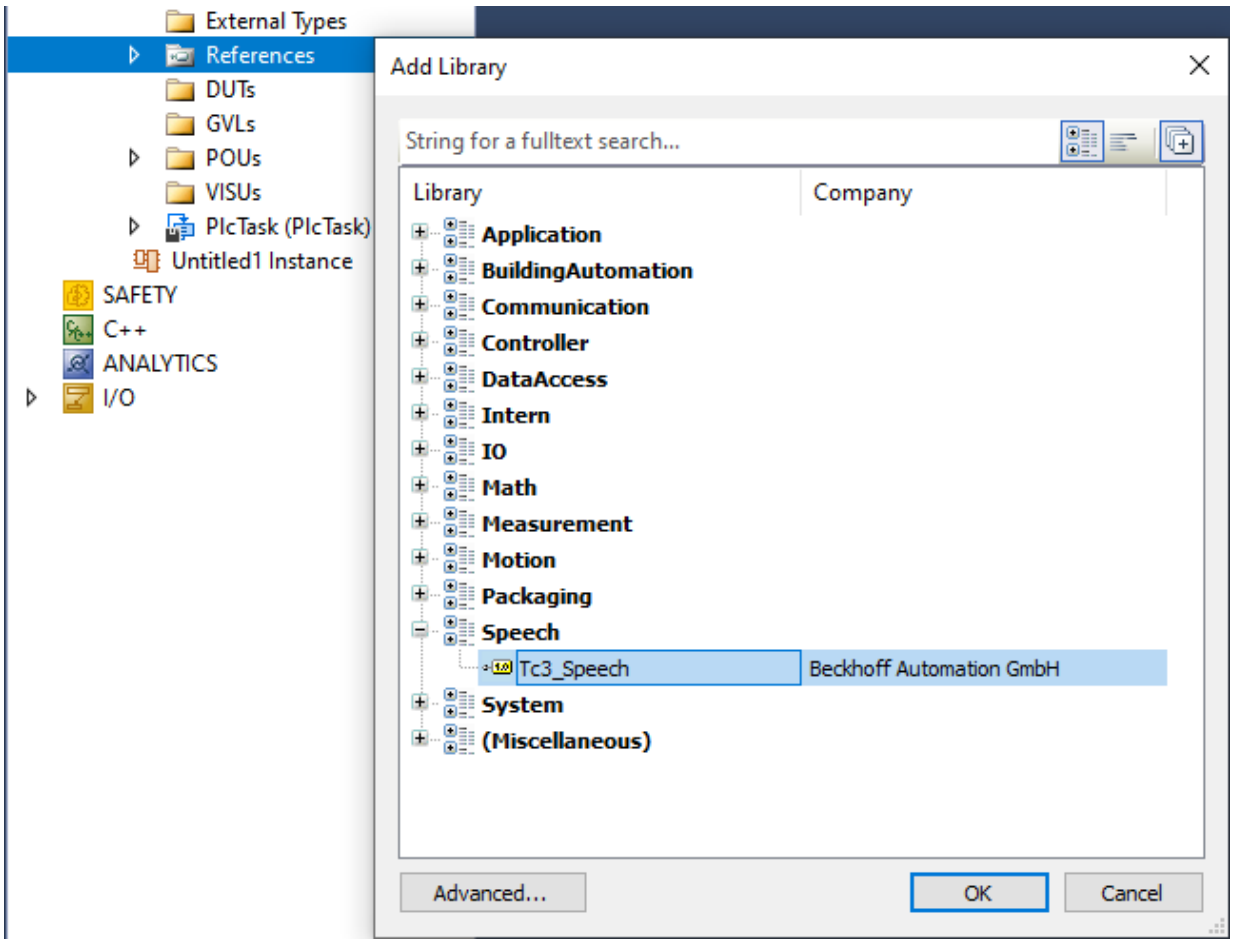
5.3 Programming the PLC

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 [▶ 65].

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;
{attribute 'TcEncoding':= 'UTF-8'}
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;
```

Process part:

```
IF NOT binit THEN
    fbTTS.SetAmsNetAddr(GVL_SpeechDemo.sAmsNetId);
    fbASR.SetAmsNetAddr(GVL_SpeechDemo.sAmsNetId);
    binit := TRUE;
END_IF
// ASR - Automatic Speech Recognition
// Trigger Start/Stop Listening
IF bListenTrigger THEN
    bListenTrigger:=FALSE;
    bListen := TRUE;
END_IF
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
END_IF
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
END_IF
// Check if new Recognition is available
IF nLastRecoId <> fbASR.nRecognitionId THEN
    IF NOT bSpeak THEN
        IF fbAsr.fRecognitionConfidence > fConfidenceThreshold THEN // if recognition better than
treshhold, answer via TTS. Next cycle will process via FB_SwitchLanguage
        // Check if Recognition Confidence is high enough
        nLastRecoId := fbASR.nRecognitionId;
        fbRetrieveUtterance(nLanguageId := nLanguageId, sRecognitionTag :=
fbASR.sRecognitionTag,fRecognitionConfidence := fbAsr.fRecognitionConfidence, sText2Speech =>
sText2Speech);
        bSpeak := TRUE;
        END_IF
    END_IF
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)

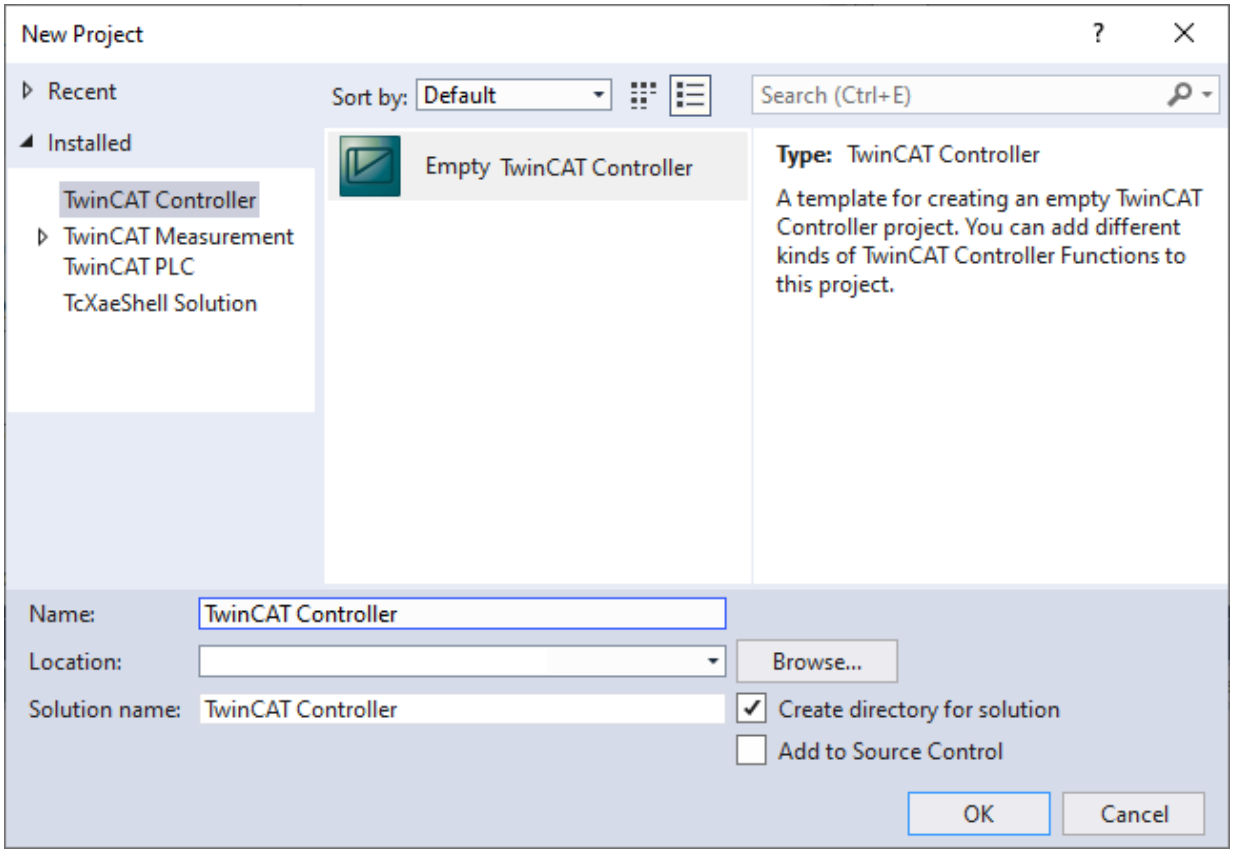
The following chapter is intended to illustrate the configuration of TwinCAT Speech for speech output.

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:

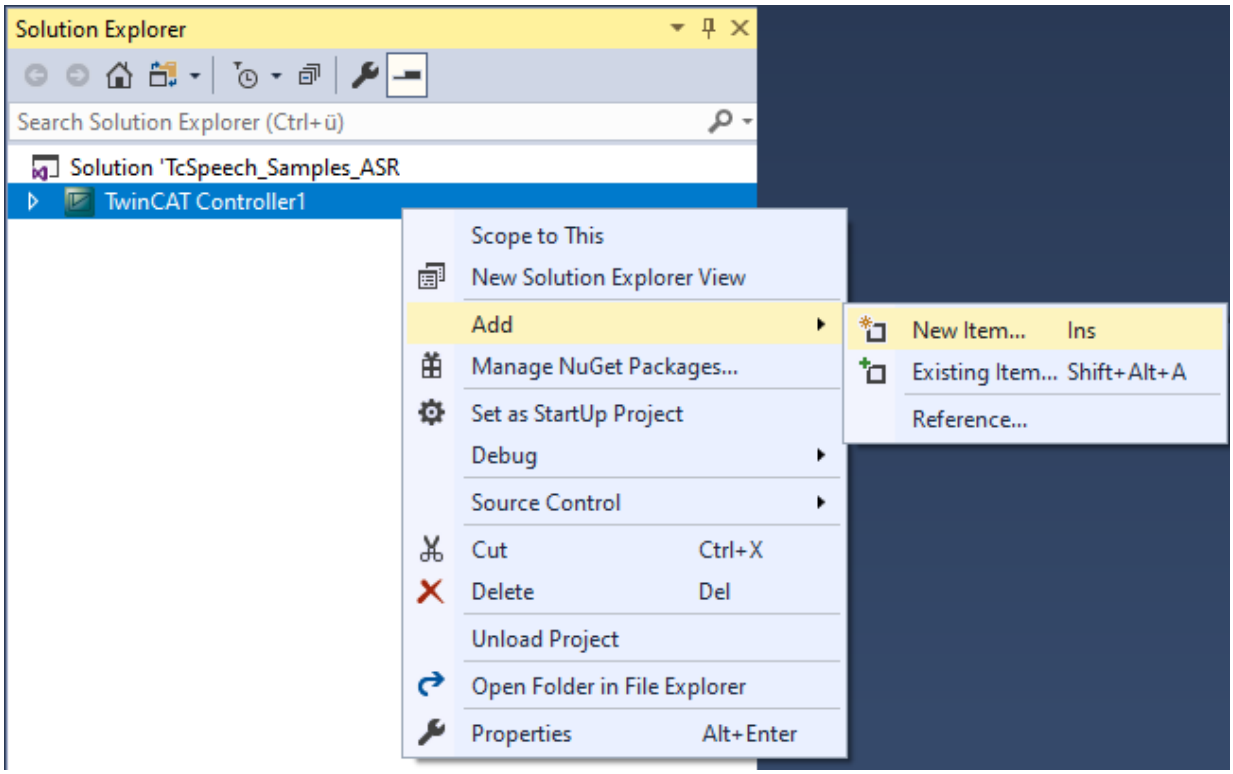
1. Open TwinCAT Engineering and create a new project.
2. Select an empty TwinCAT Controller project, name it, and click **OK**.



⇨ The new TwinCAT Controller project opens.

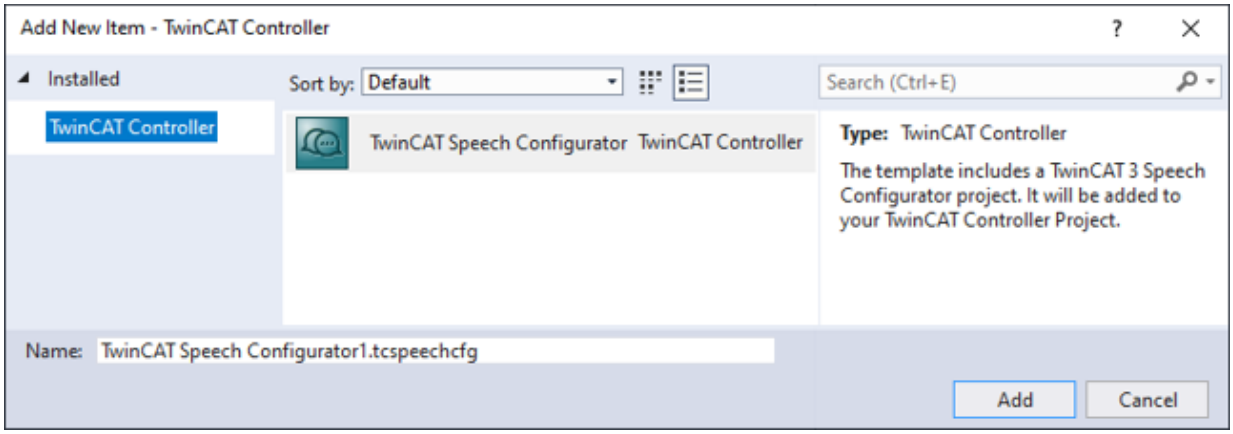
3. Right-click the new TwinCAT Controller project.

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.

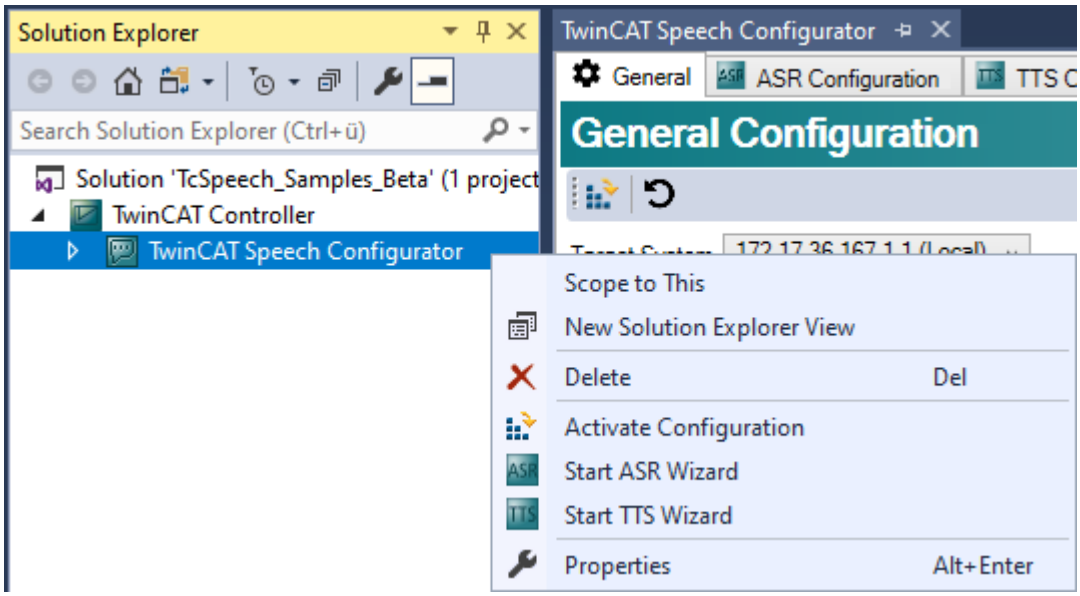
⇨ Assignment to a target system is important for TwinCAT Speech, because the hardware configuration of the sound cards is individual.

⇨ A new TwinCAT Speech configuration has now been created.

6.2 Configuring speech output

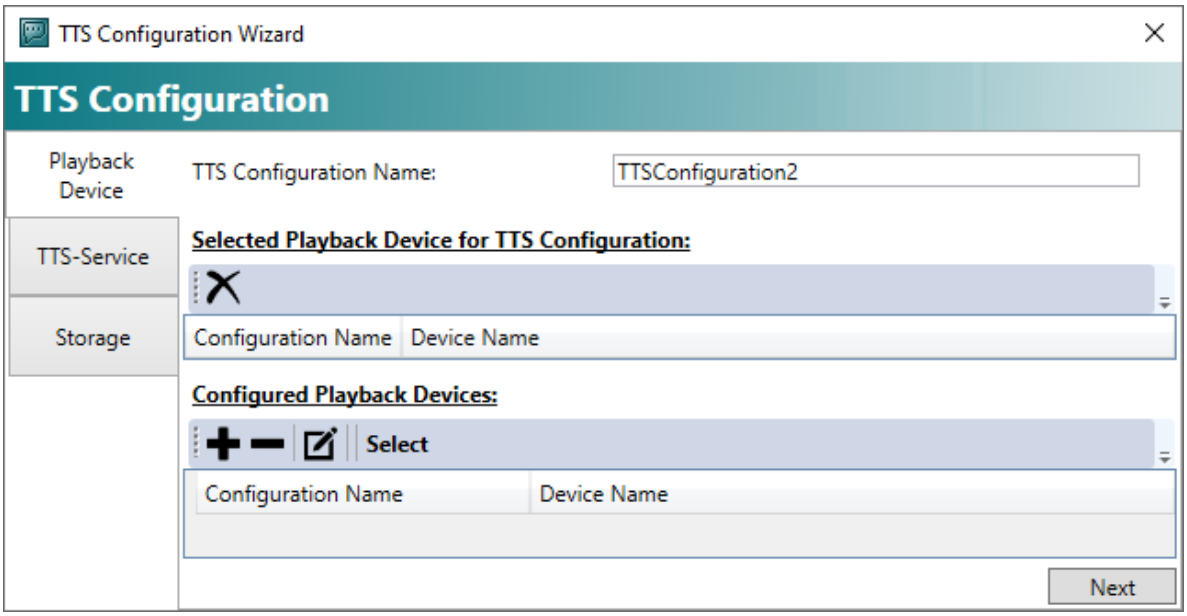
A configuration wizard is available for the configuration of the speech output (TTS). This guides you through the TTS configuration

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



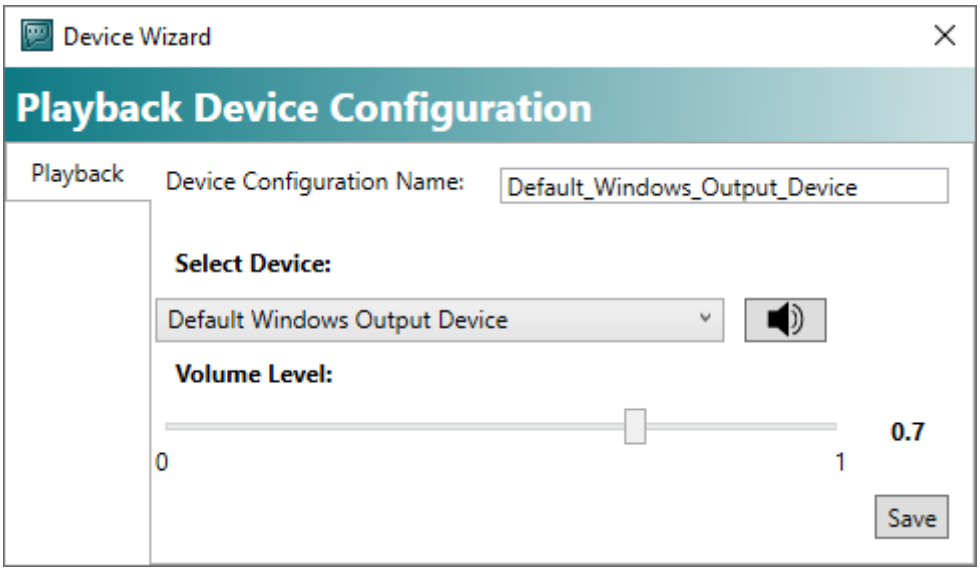
2. In the context menu, select **Start TTS Wizard**

   ⇨ 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.

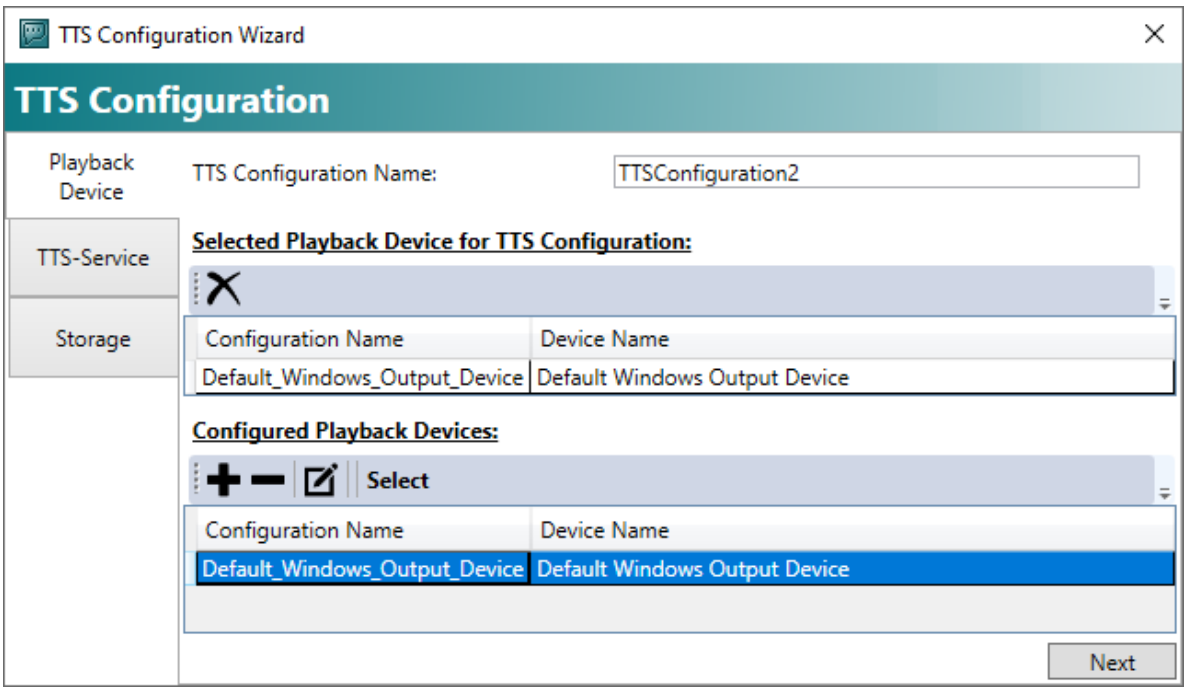


3. To add a new playback device, click the **+** button.

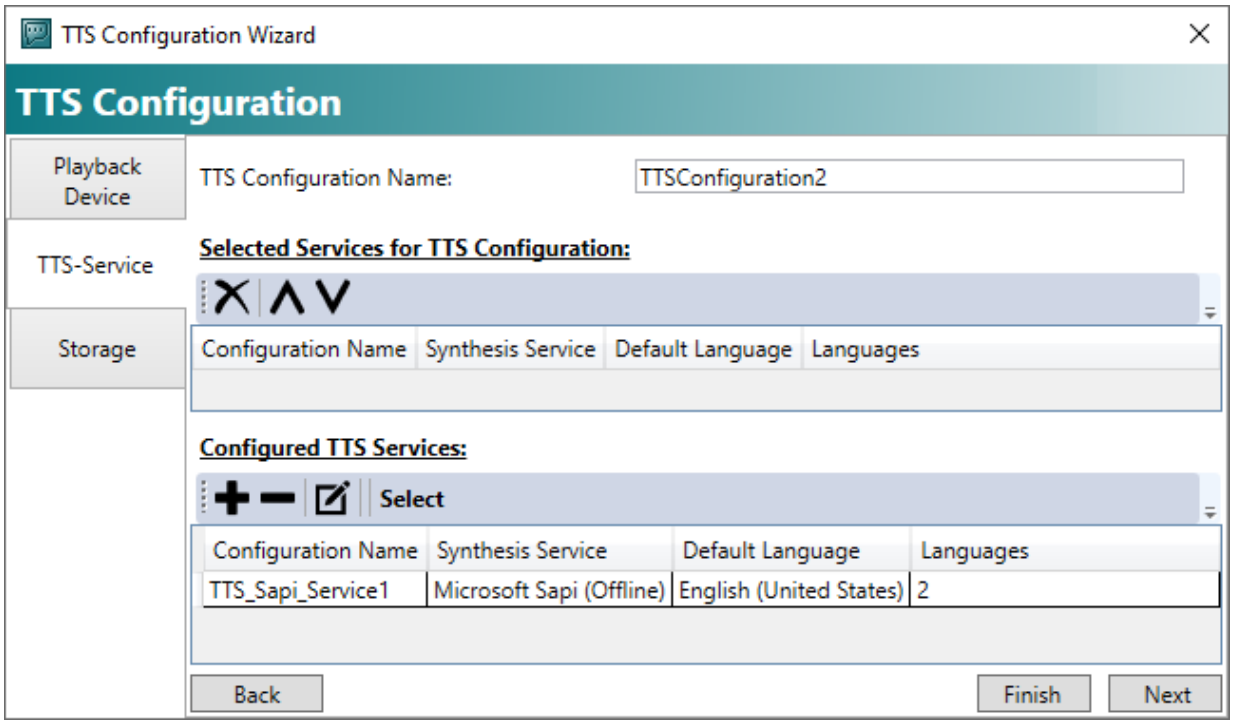
⇨ The Device Wizard opens.



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**.



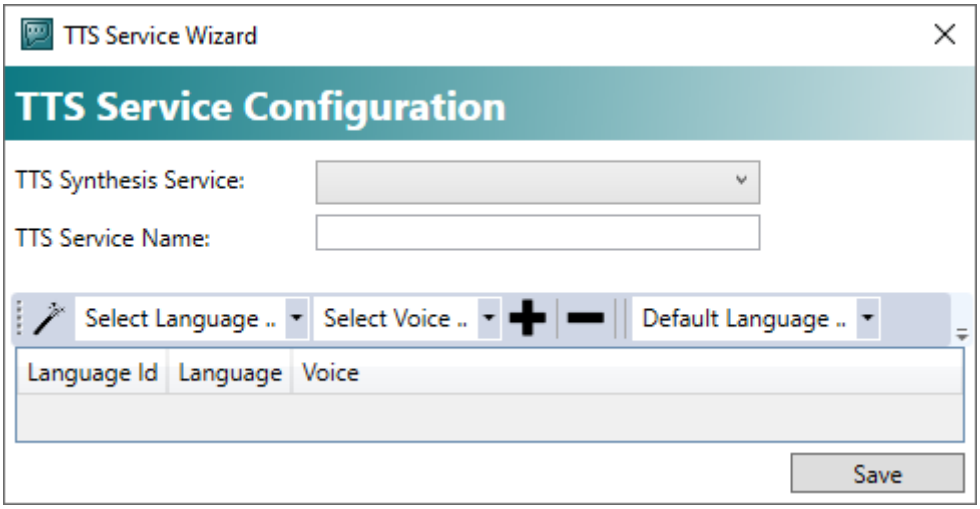
7. Click on **Next**.



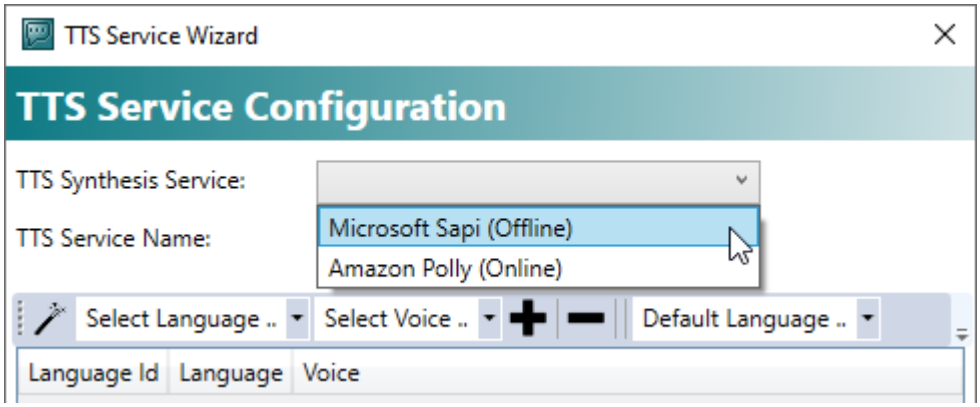
⇨ The second part of the configuration opens, where you configure the actual speech generation service (Text-To-Speech).
As in the first step, a list of services that have already been configured is shown below. The selected service is displayed above.

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

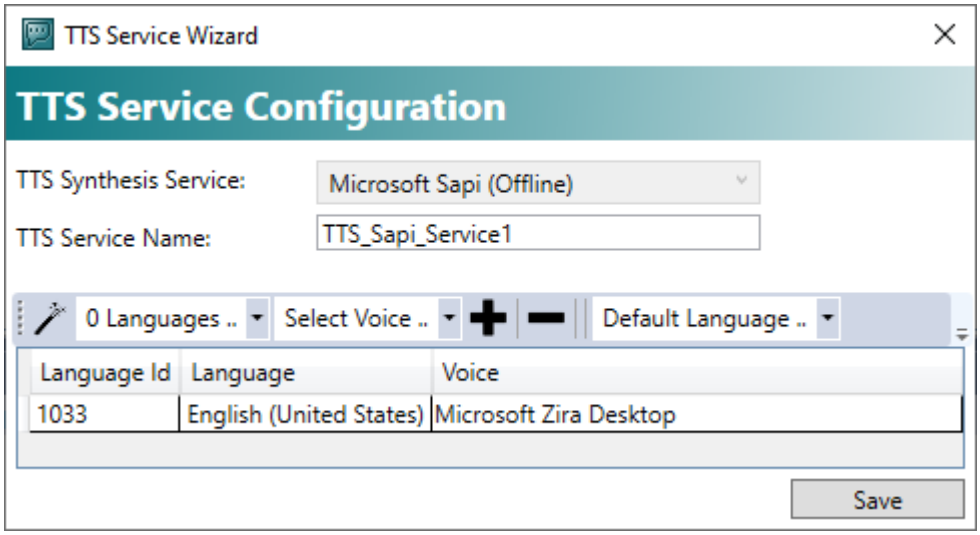
⇨ The TTS Service Wizard opens.



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.

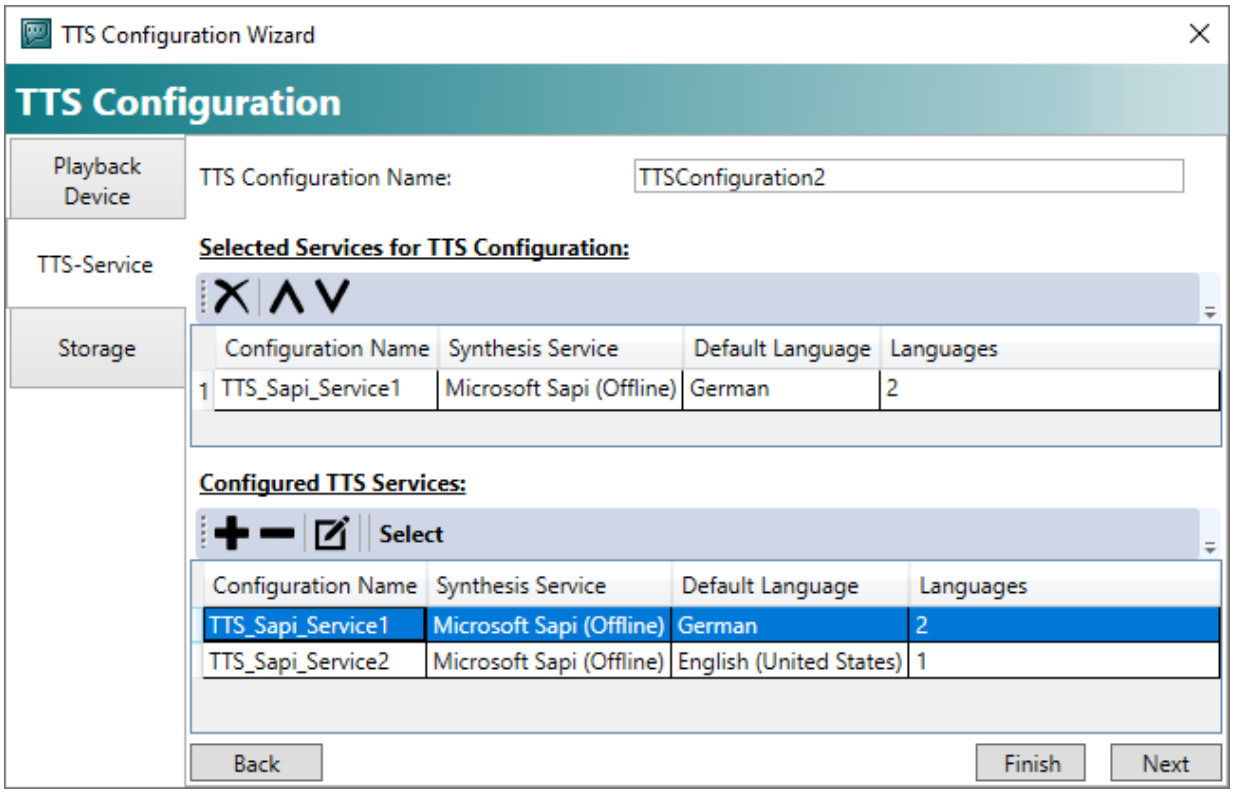


10. You can give the speech recognition service its own name in the **TTS Service Name** input field.
11. Now set a language and voice for the speech output.

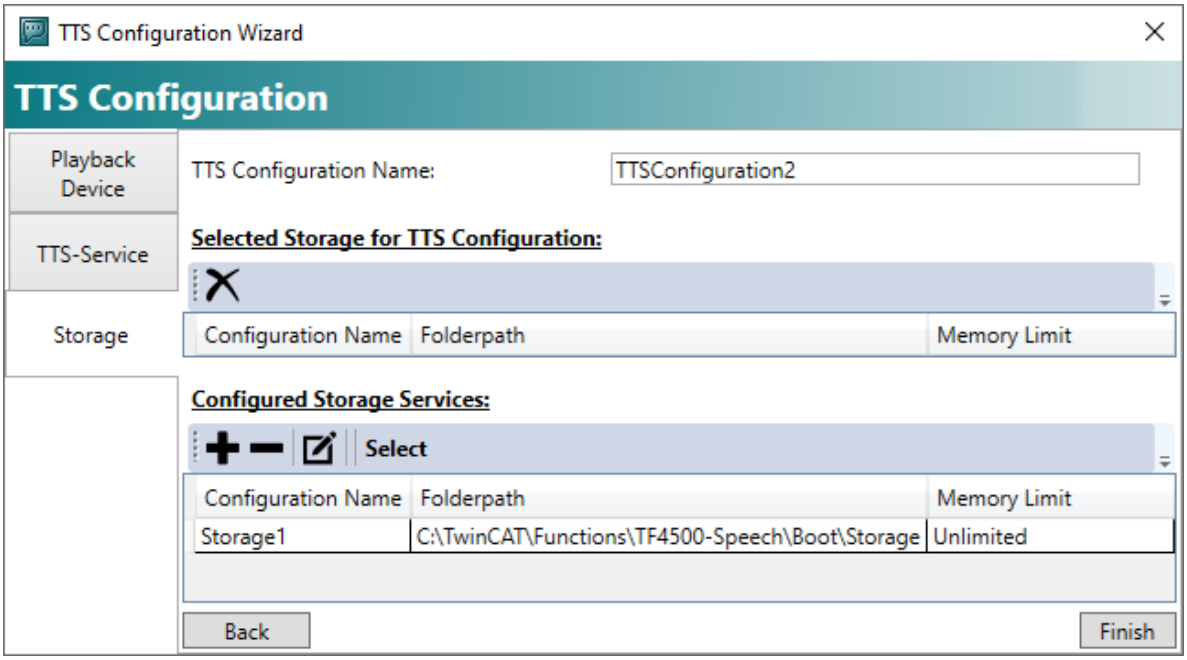


12. You can use the Magic Wand  button to automatically generate the available languages and voices.
13. Alternatively, select the desired language in the first drop-down list **Select Language** and an available voice in the second drop-down list **Select Voice**.
14. Click the **+** button.
15. In 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.

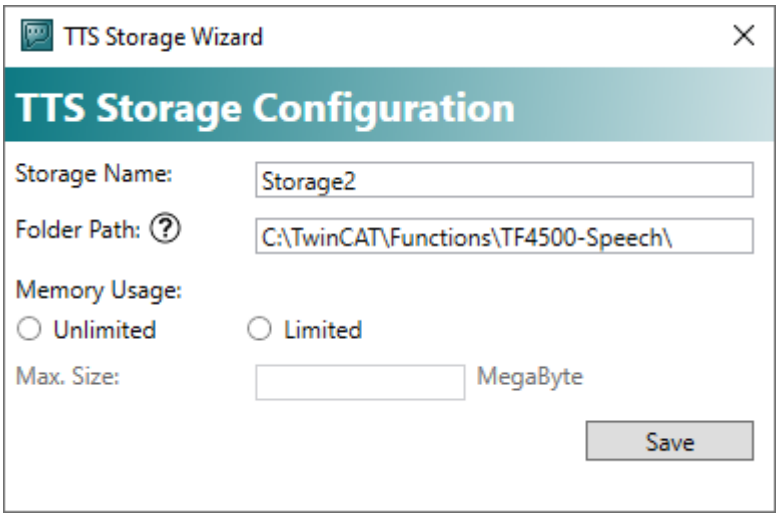


18. Click the **Next** button to save the TTS configuration.
19. To save and stop the configuration at this point, click the **Finish** button.
    ⇨ When you click the **Next** button, the third part of the configuration opens. In the optional third part of the configuration, the wizard concludes by guiding you through the local saving of the generated TTS outputs on the target system. This allows you to reuse them at any time.

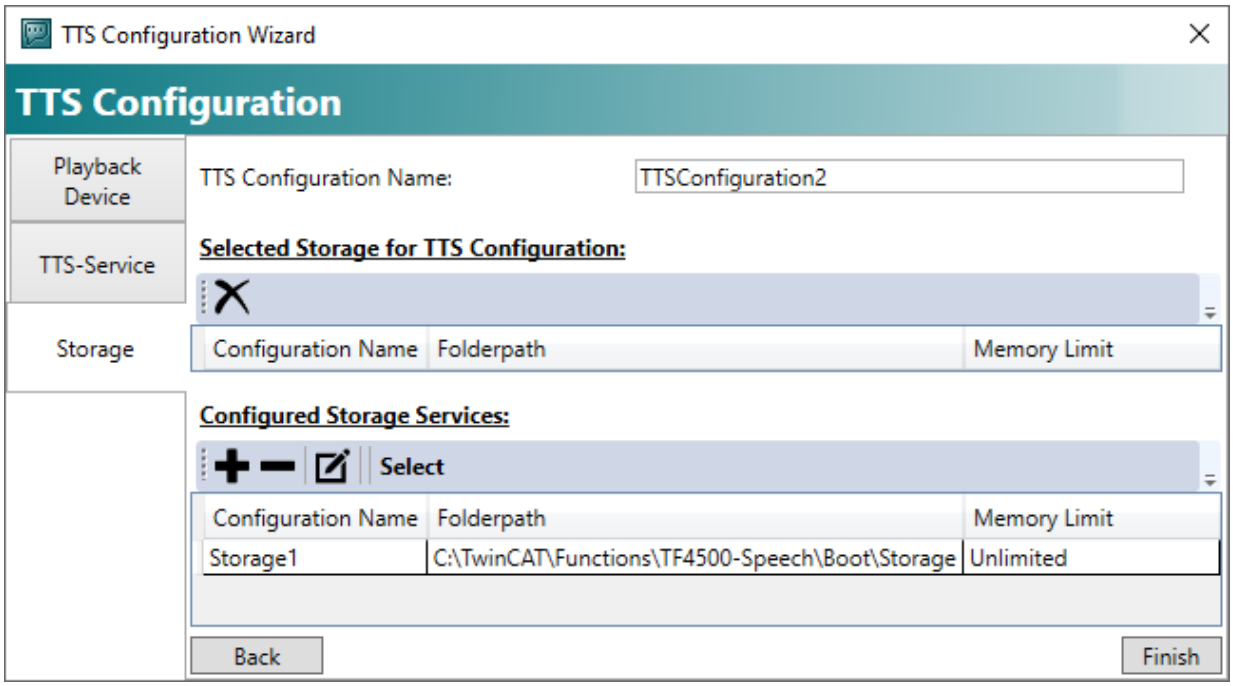


20. To determine whether the TTS outputs should be saved, click the **+** button.

⇨ The TTS Storage Wizard opens.

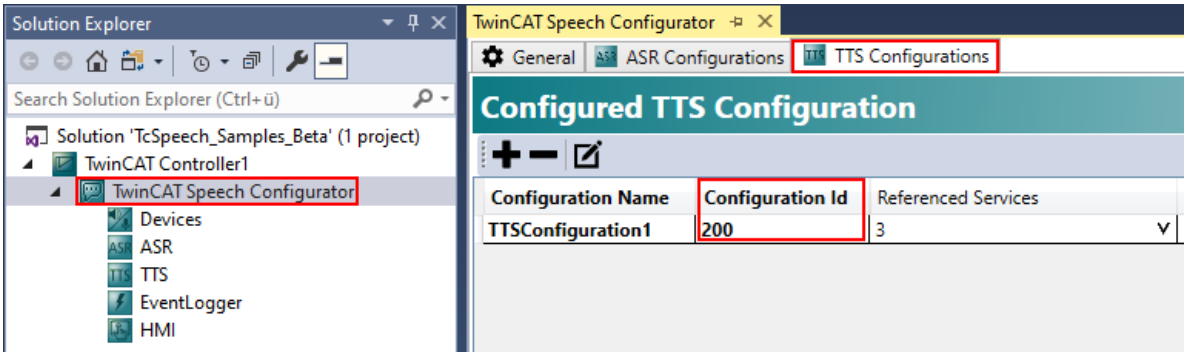


21. In the **Storage Name** input box, enter a name for the file to be saved.
22. In the **Folder Path** box, determine a memory location for the file to be saved.
23. In the **Memory Usage** area, you determine whether all TTS outputs are stored or whether the memory space should be limited. Select 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.

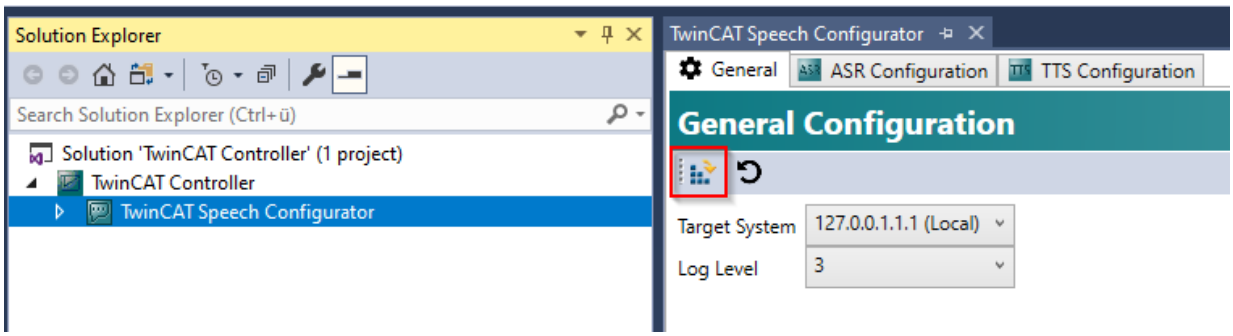


27. Click the **Finish** button.

⇨ 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.

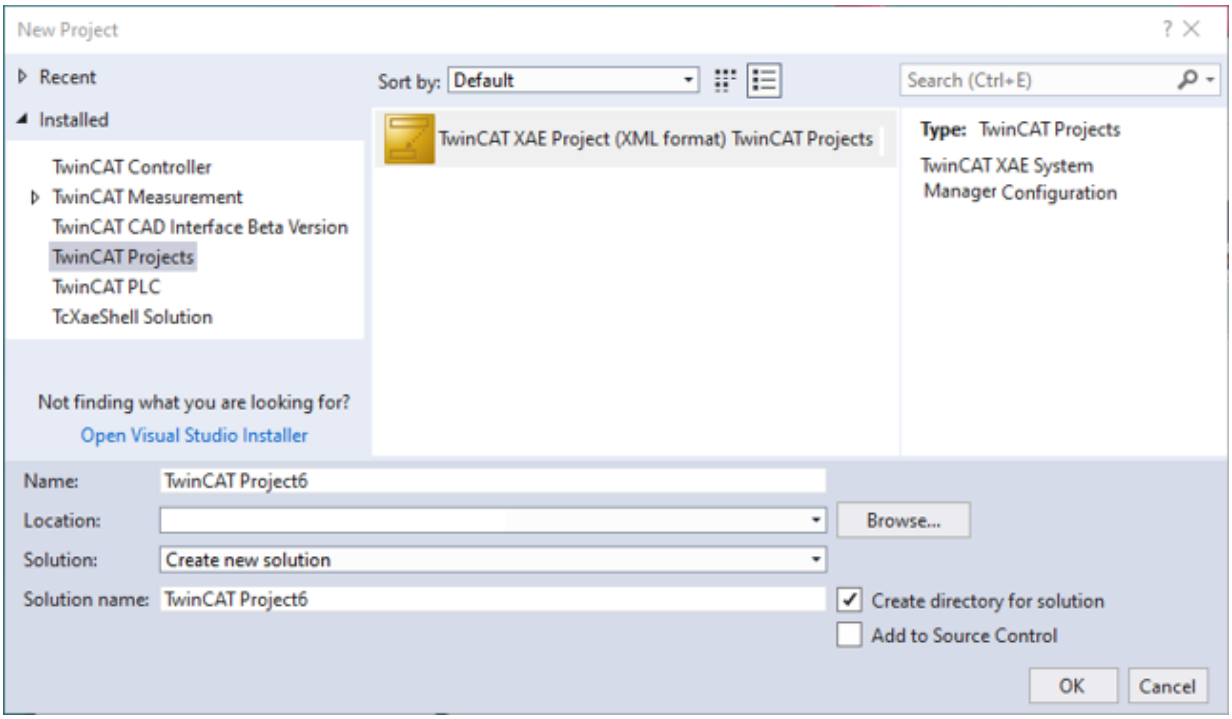
Also see about this

- Microsoft SAPI: installing additional languages [▶ 55]

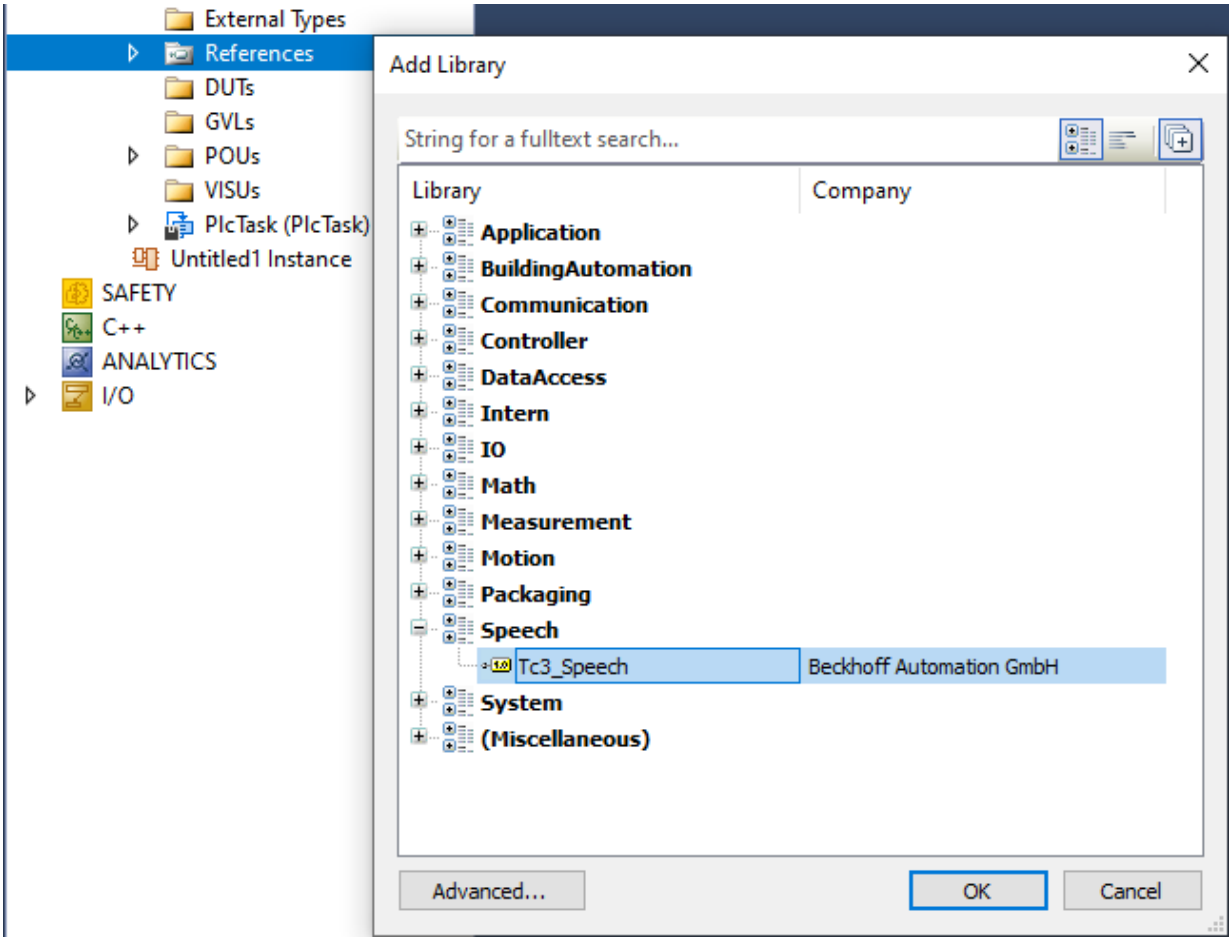
6.3 Programming the PLC

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.



2. Add the Tc3_Speech library.



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;
{attribute 'hide'}
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
    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.

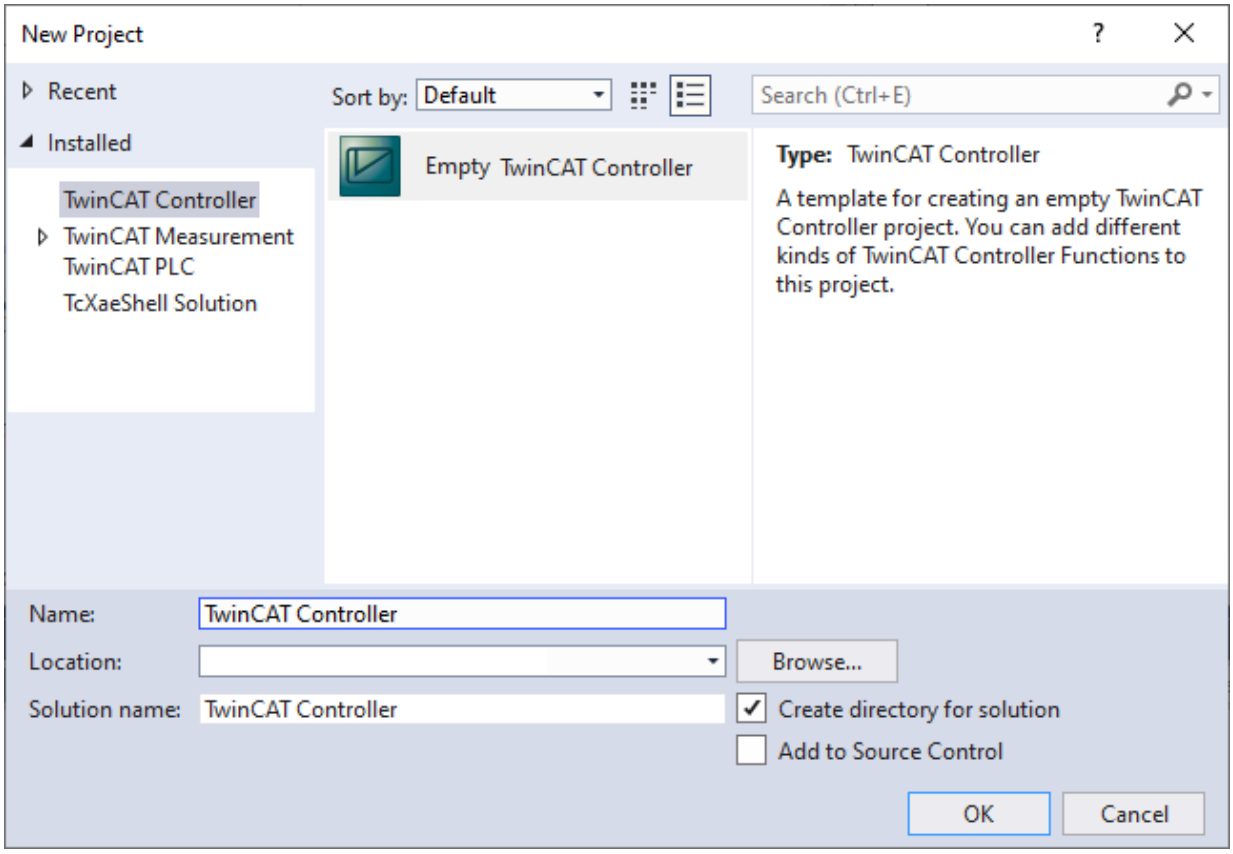
Also see about this

- 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:

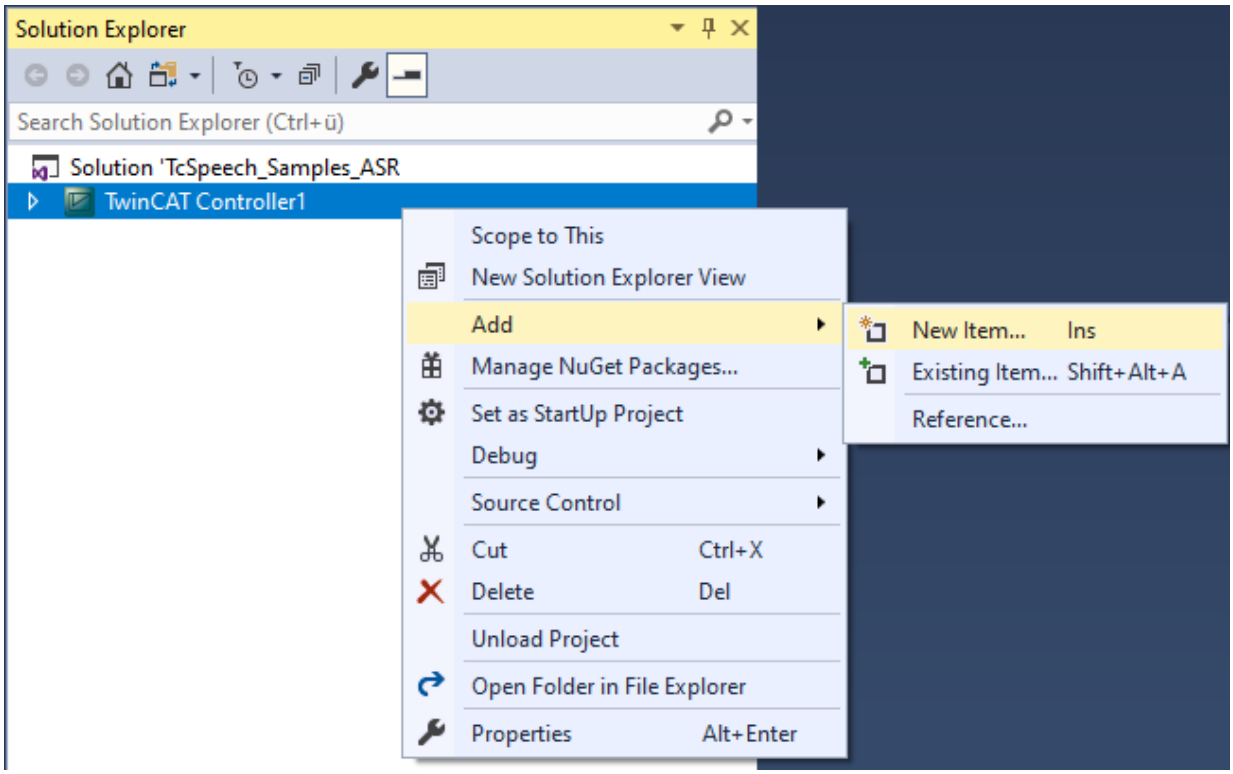
1. Open TwinCAT Engineering and create a new project.
2. Select an empty TwinCAT Controller project, name it, and click **OK**.



⇨ The new TwinCAT Controller project opens.

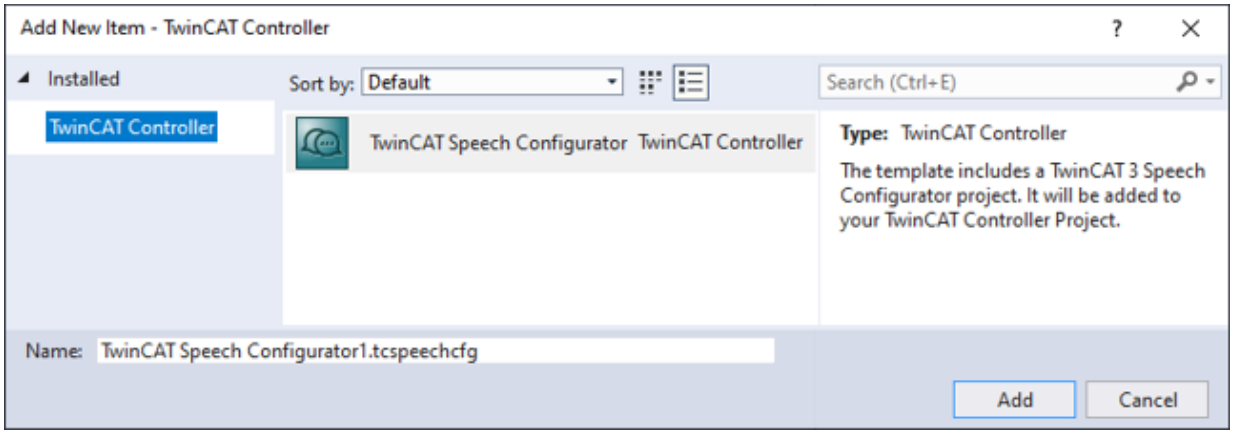
3. Right-click the new TwinCAT Controller project.

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.

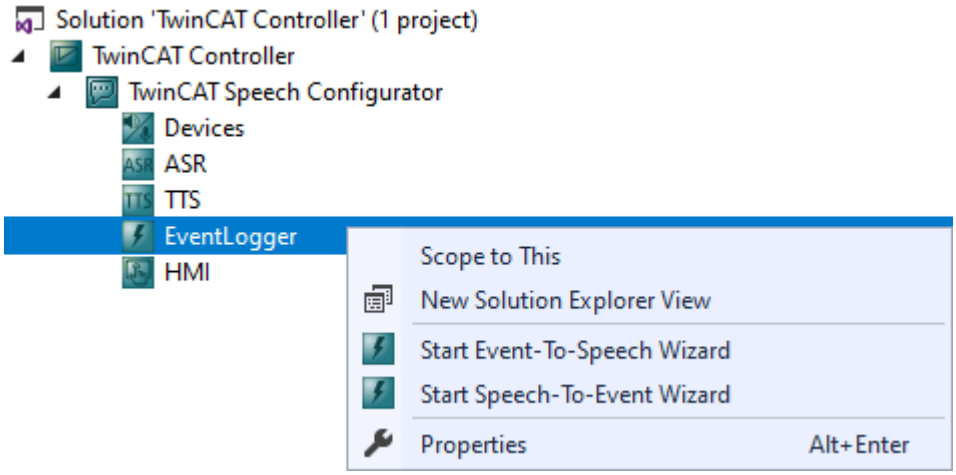
⇨ A new TwinCAT Speech configuration has now been created.

7.2 Configuring Event-To-Speech

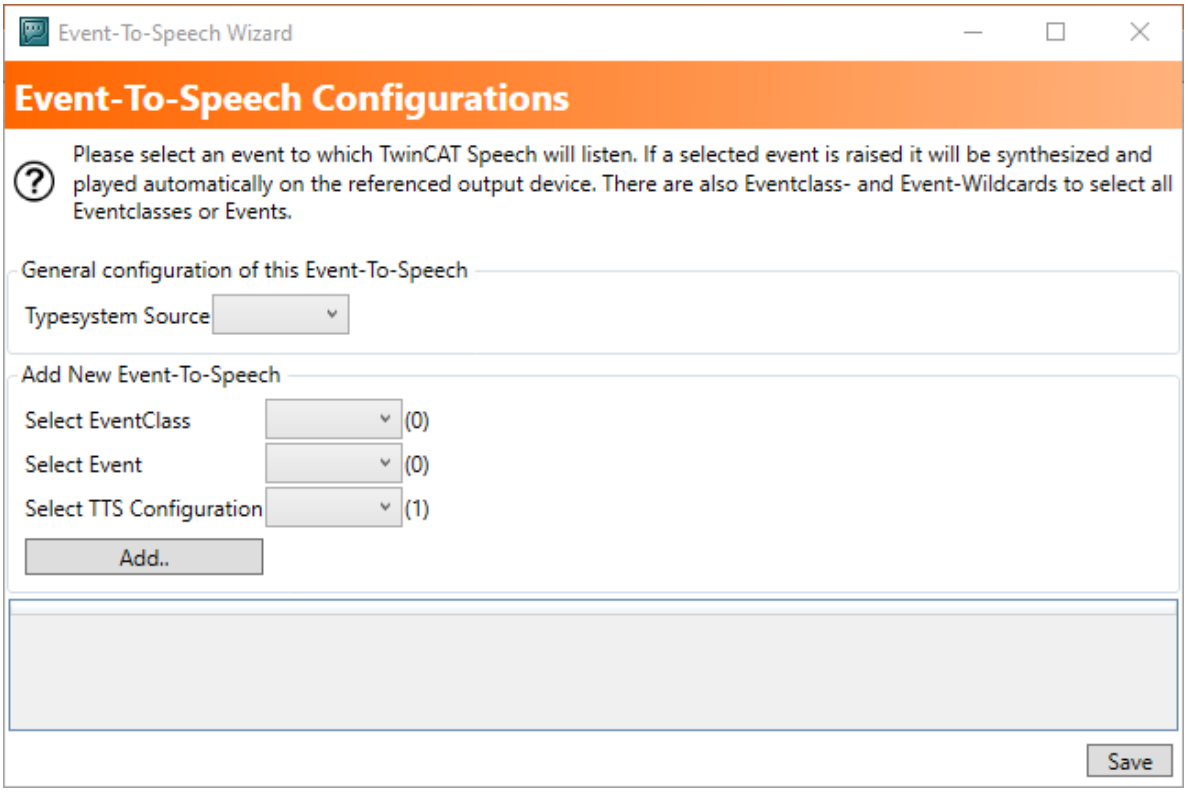
A configuration wizard is available for the configuration of the speech output of events. This guides you through the Event-To-Speech configuration.

✓ 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) [▶ 27].

1. In order to open the Event-To-Speech wizard, right-click the EventLogger configuration in the Solution Explorer.

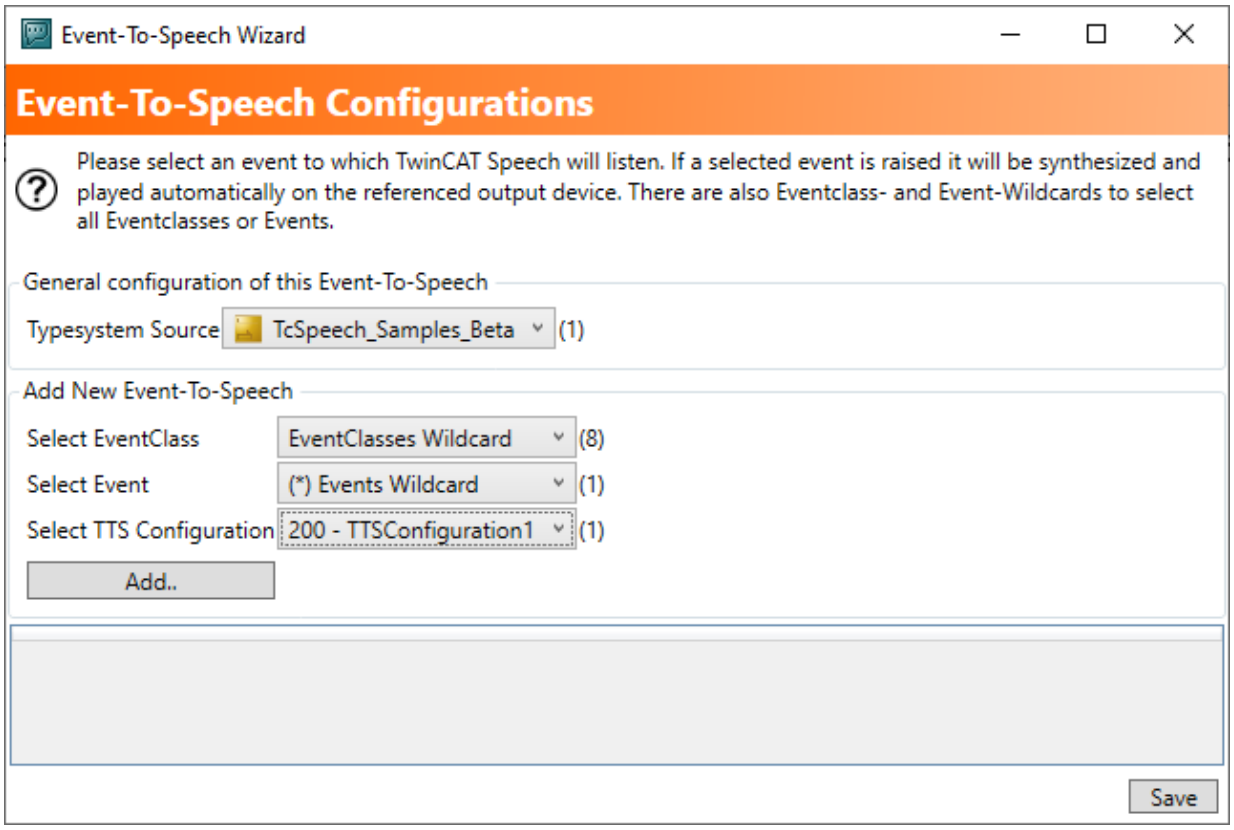


2. Select **Start Event-To-Speech Wizard** from the context menu.
   ⇨ The Event-To-Speech wizard starts.

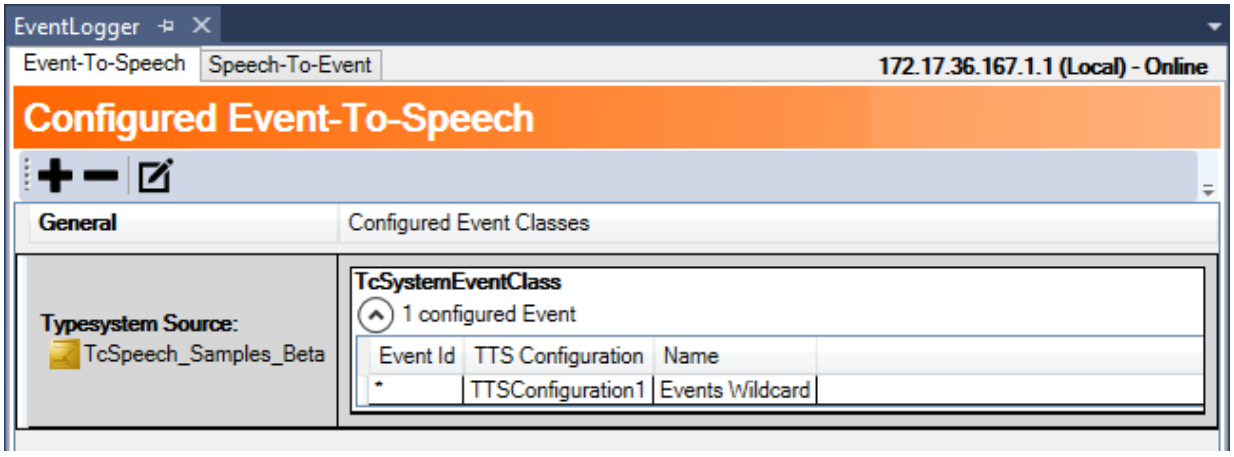


3. Select the respective TwinCAT project under Main Configuration.

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.



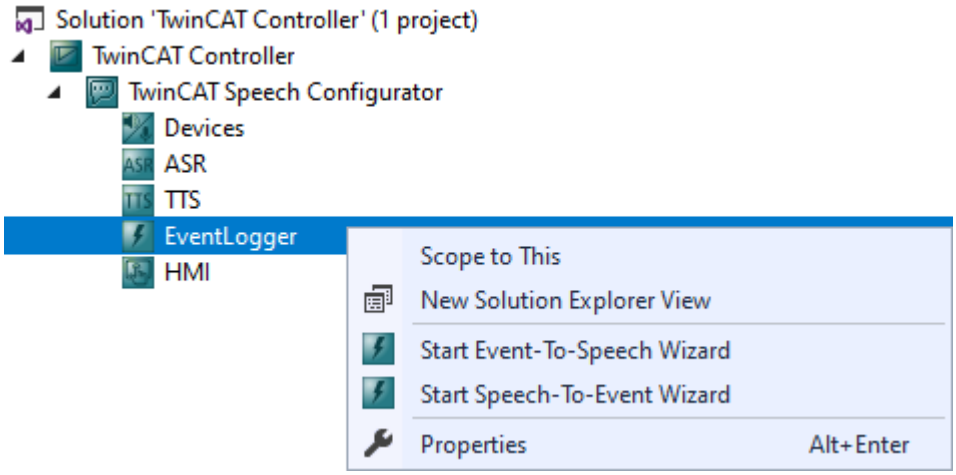
⇨ On occurrence of an event, the text of the event will be output via the selected speech output service using the default language stored there.

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

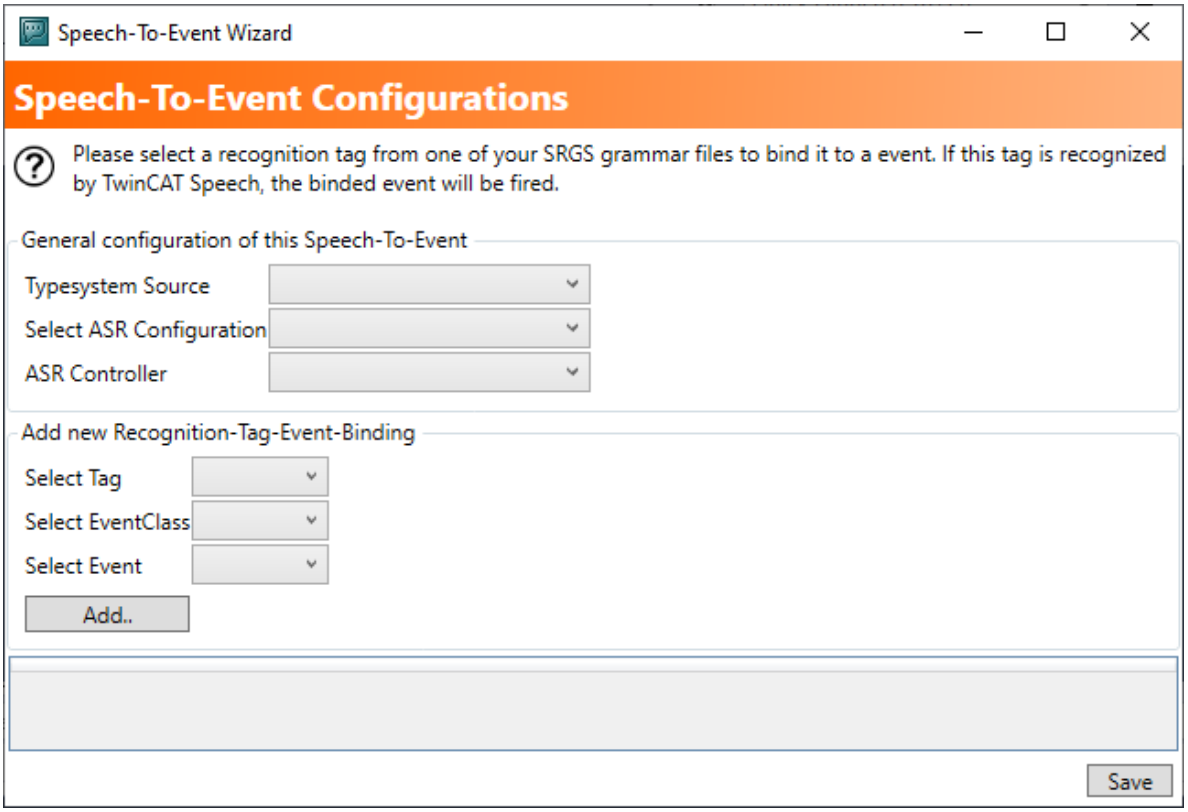
7.3 Configuring Speech-To-Event

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.

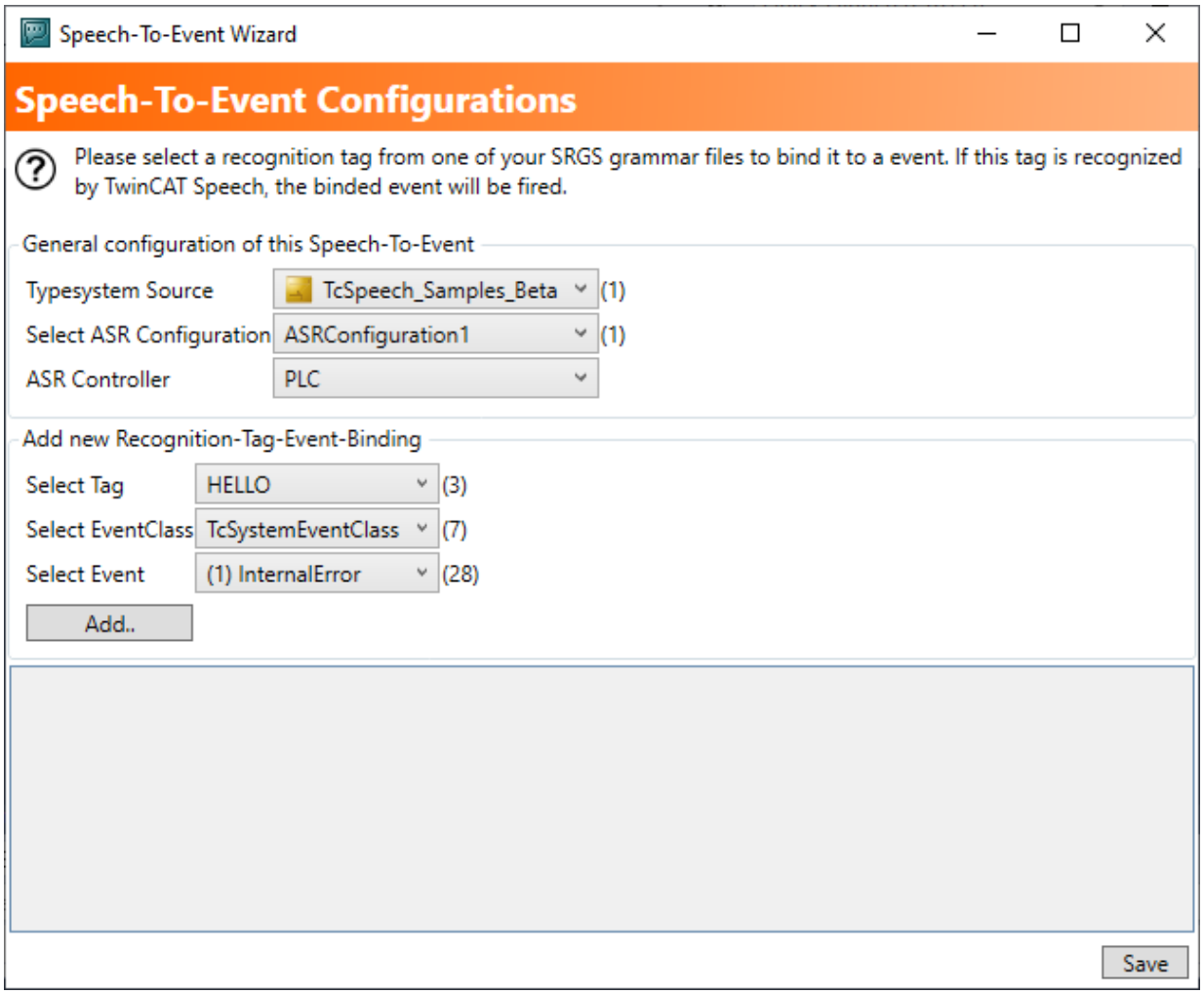


2. Select **Start Speech-To-Event Wizard** from the context menu.
   ⇨ The Speech-To-Event wizard starts.

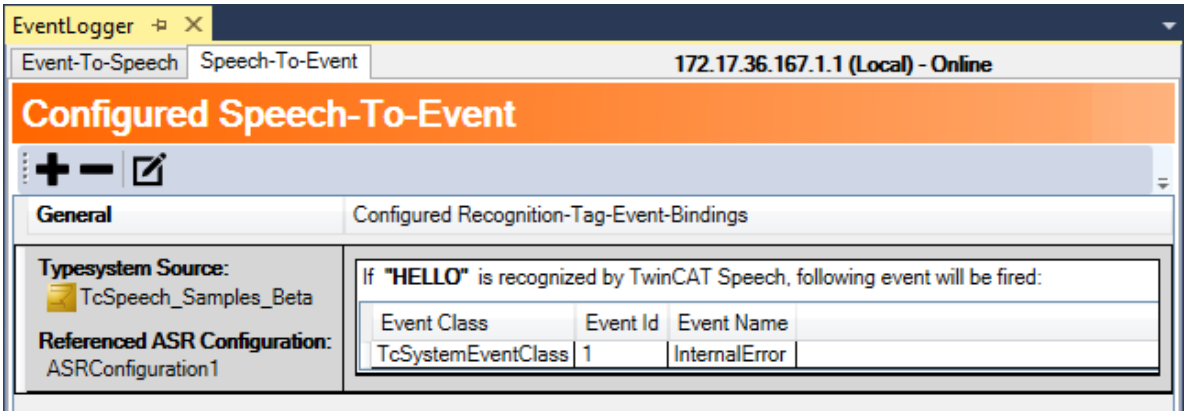


3. Select the respective TwinCAT project under Main Configuration.

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.

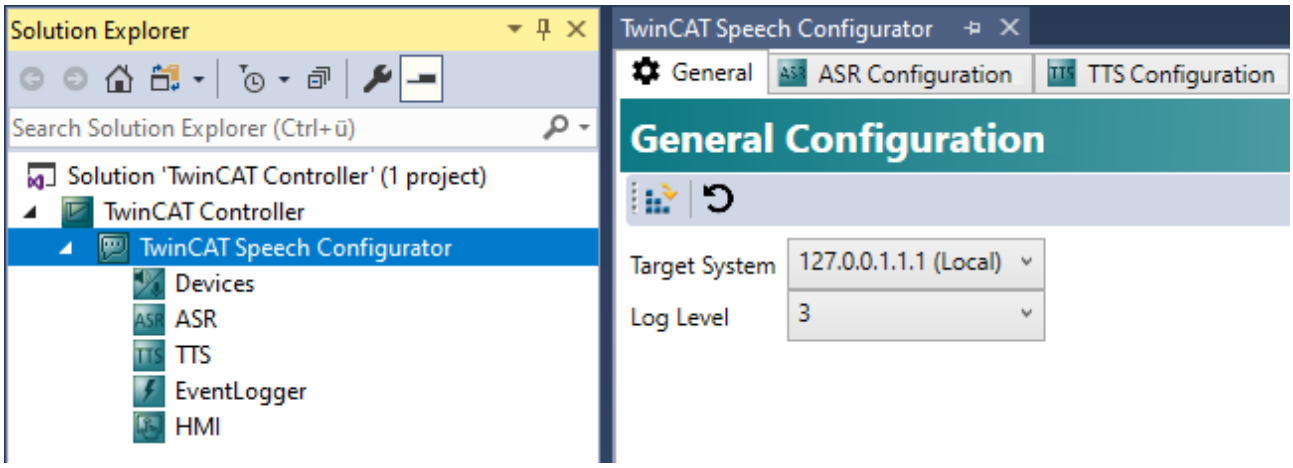
The associated sample [▶ 67] provides a PLC project for triggering events in a simple way.

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

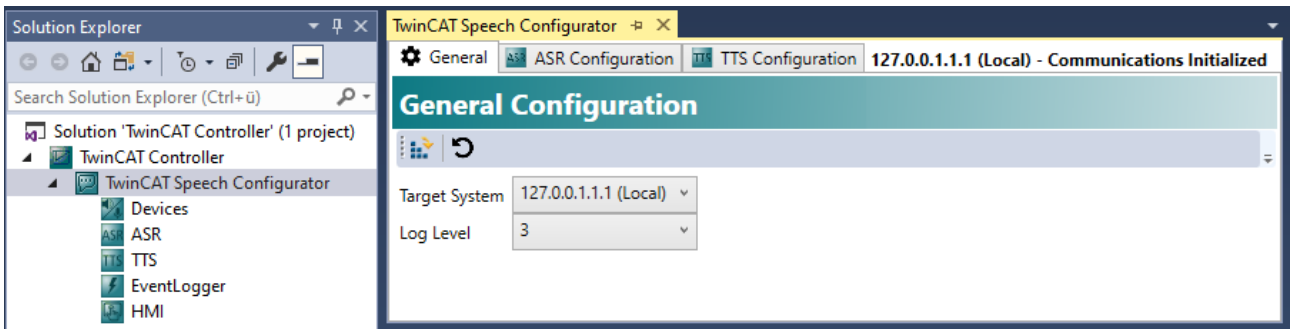
If you have created the TwinCAT Speech project, the window looks like this:



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

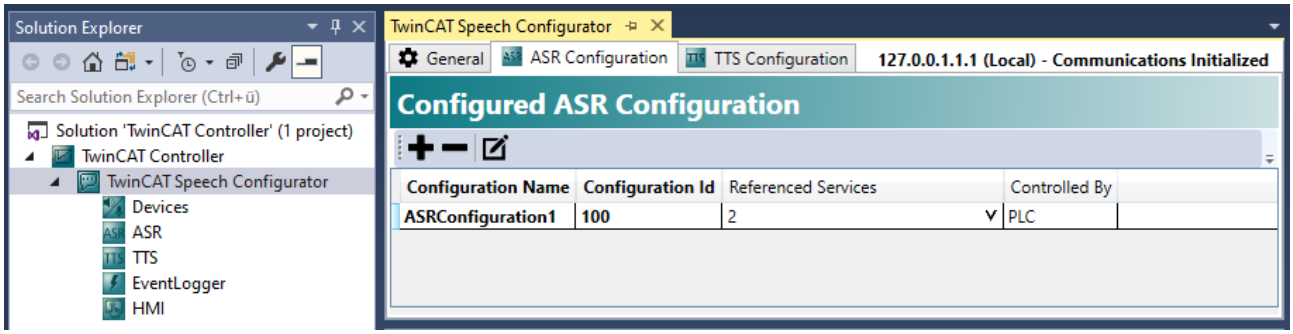
8.2 General Configurations

The **General Configurations** window provides the possibility to make various settings.



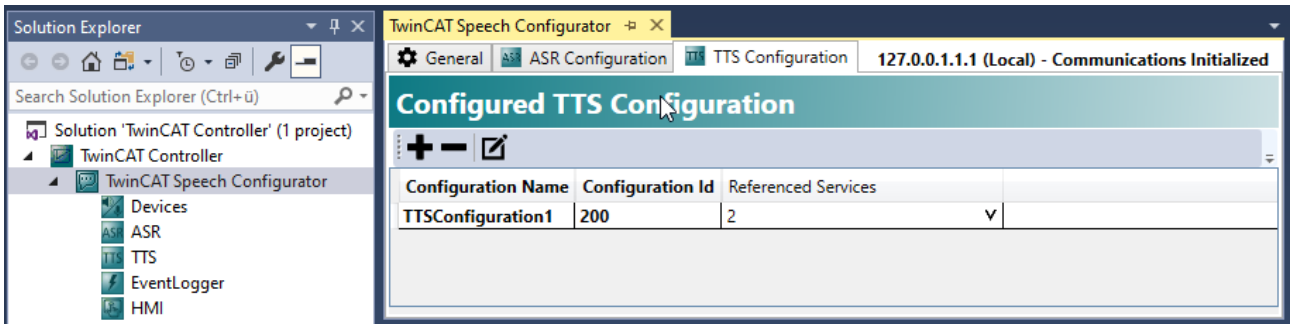
	"Activate Configuration": This transfers the configuration to the target system and restarts it for activation upon request. **Note Every change in the TwinCAT Speech Configurator has to be activated first!**
	Reload available targets.
Target System	Target system for which the configuration is intended. 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



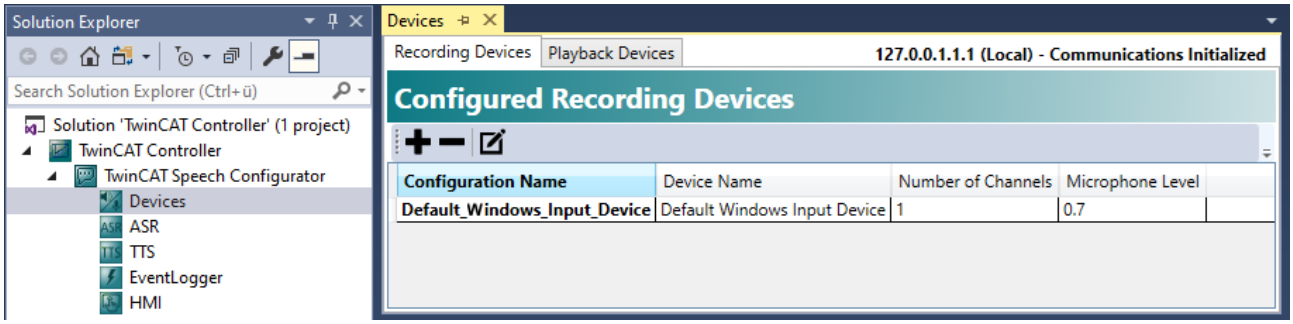
	Add a microphone.
	Delete a microphone.
	Edit the configuration of a microphone.
List	The list of currently configured speech recognitions: this combines an ASR service with a microphone and is used, for example, by the PLC to perform speech recognition

TTS Configurations

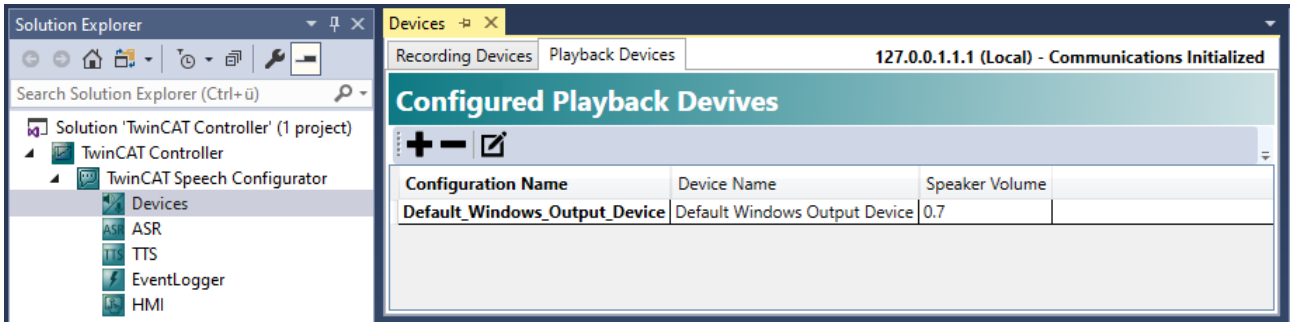


	Add a playback device.
	Delete a playback device.
	Edit the configuration of a playback device.
List	The list of currently configured speech outputs: this combines a TTS service with a playback device and is used, for example, by the PLC to perform a speech output.

Devices

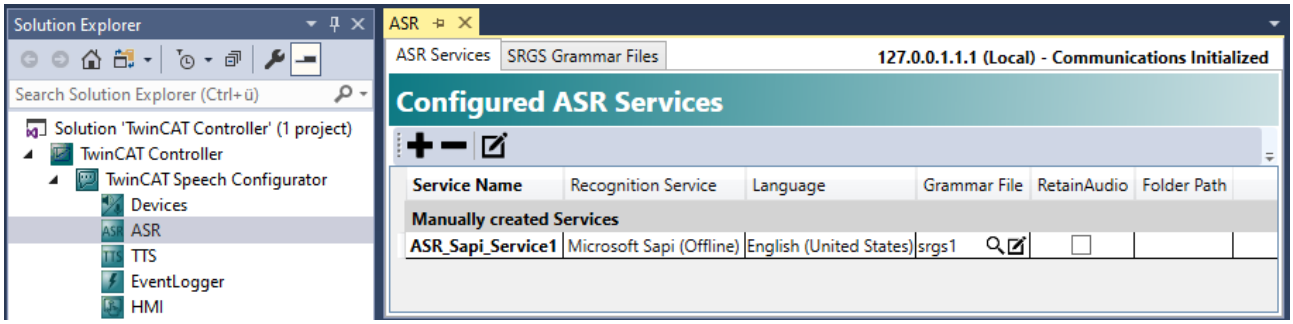


	Add a recording device.
	Delete a recording device.
	Edit the configuration of a recording device.
List	List of currently configured microphones (recording devices): these are usually created by the ASR wizard, but can also be created directly. They are combined with ASR services in the ASR configuration.



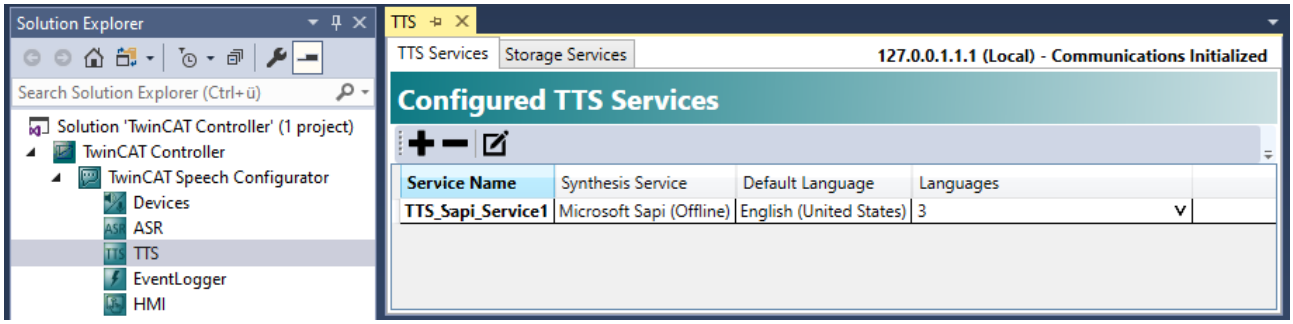
	Add a playback device.
	Delete a playback device.
	Edit the configuration of a playback device.
List	List of currently configured playback devices: these are usually created by the TTS wizard, but can also be created directly. They are combined with TTS services in the TTS configuration

ASR



	Add a speech recognition service.
	Delete a speech recognition service.
	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

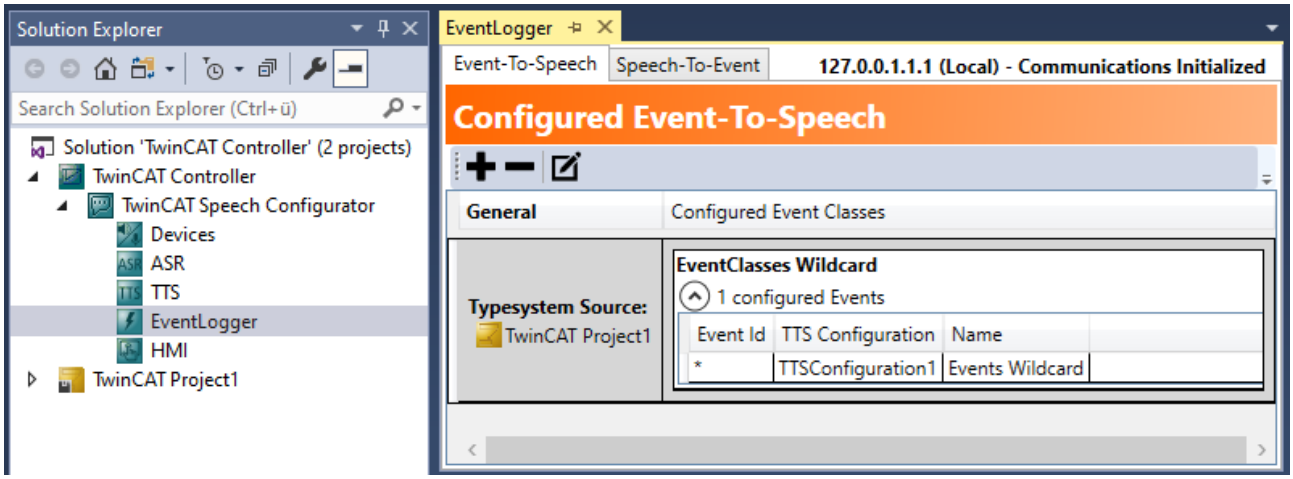
TTS



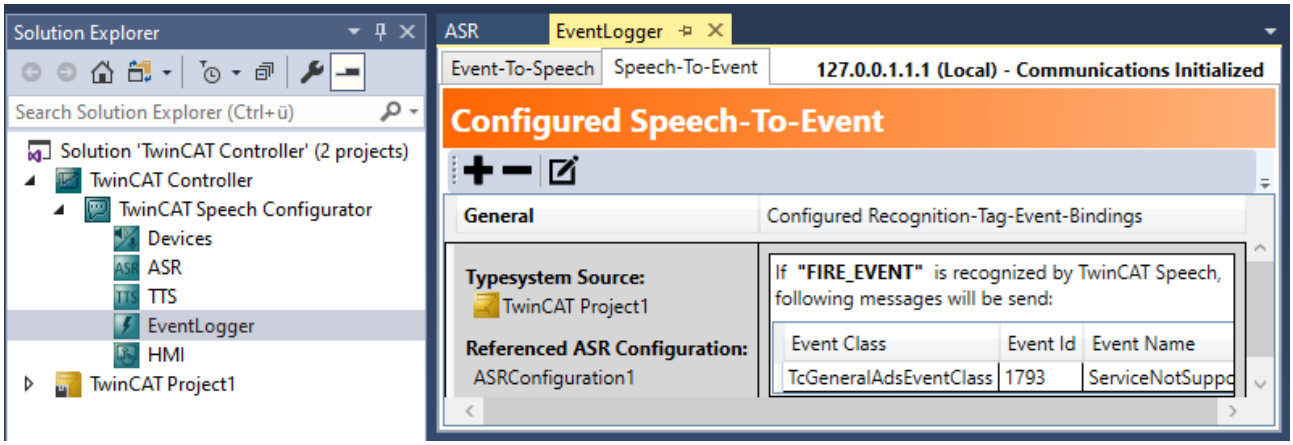
	Add a playback service.
	Delete a playback service.
	Edit the configuration of a playback service.
List	List of currently configured playback services: these are usually created by the TTS wizard, but can also be created directly. They are combined with playback devices in the TTS configuration.

EventLogger

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

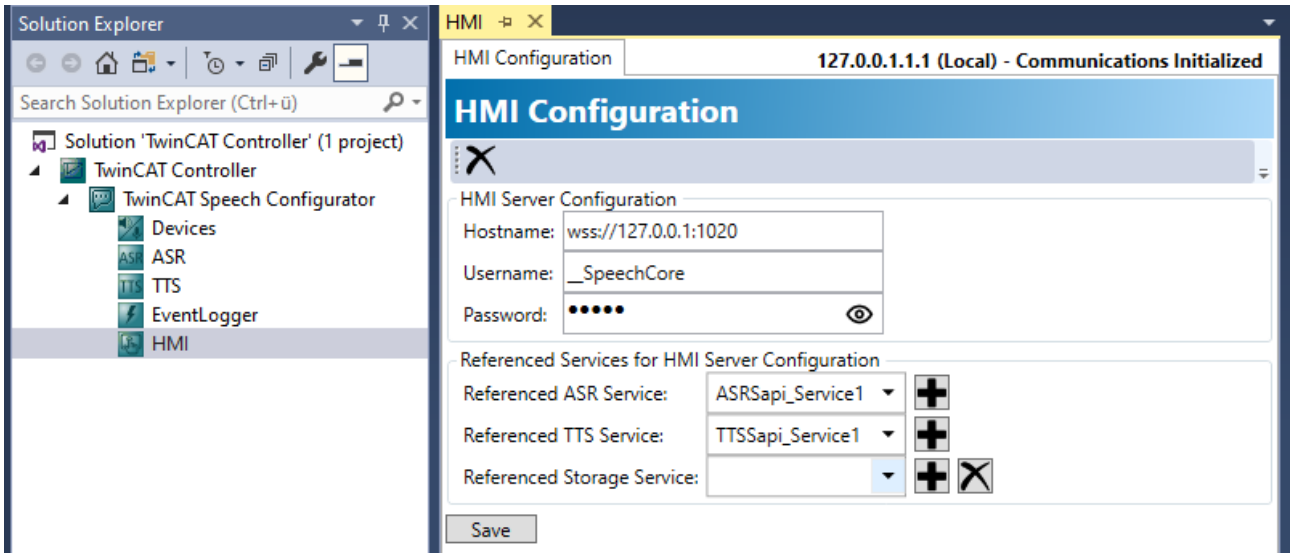


	Add an event.
	Delete an event.
	Edit the configuration of an event.
Event-To-Speech	The events that should be output via speech output on occurrence are defined on the "Event-to-Speech" tab. A TTS configuration is used to reference the TTS service and playback device to be used . The default language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



	Add an event.
	Delete an event.
	Edit the configuration of an event.
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.

HMI



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 [▶ 52]. Note that only SRGS files created by the Simple SRGS Editor can be used for language switching.

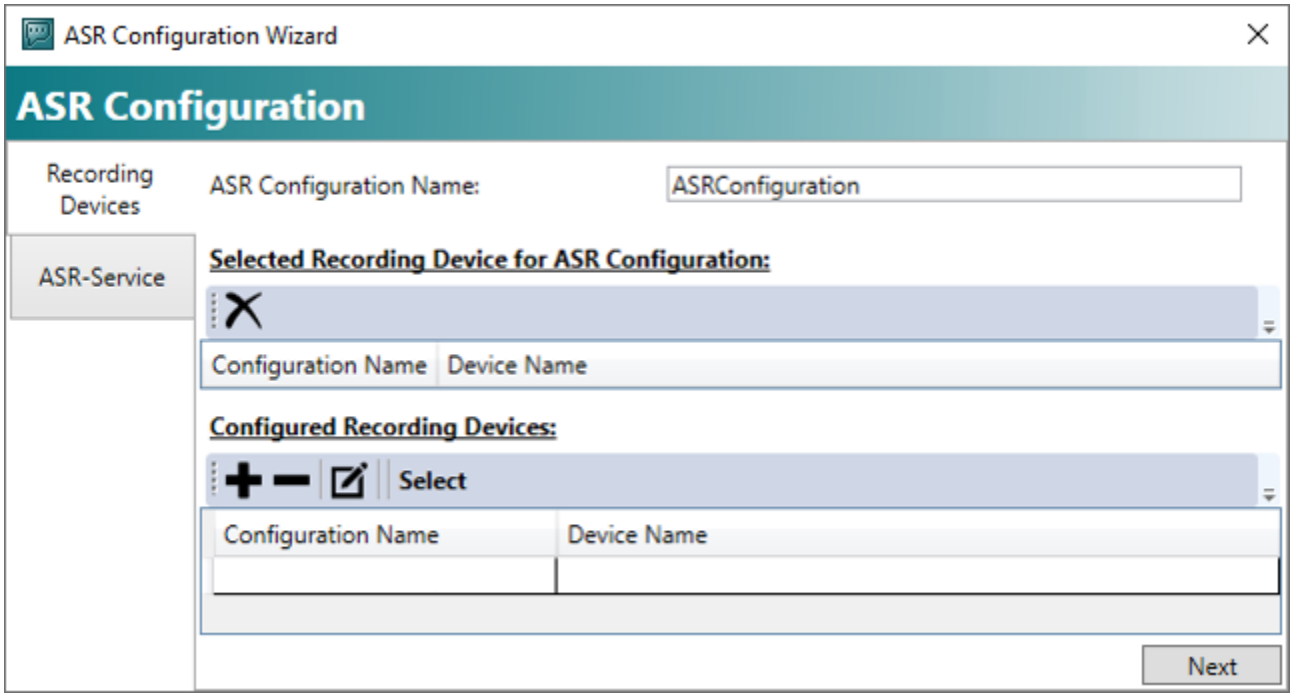
The **Recognition Tag**s defined in the SRGS file can trigger any actions in the HMI. See the TwinCAT HMI documentation.

8.3 Testing the sound card

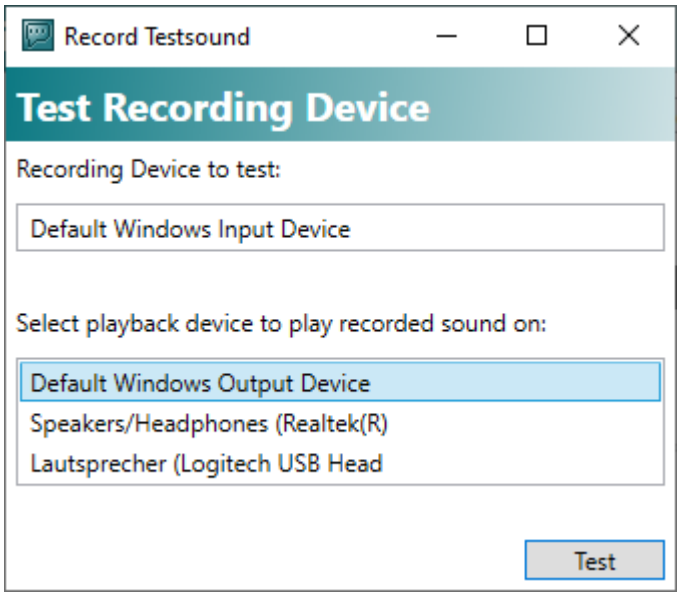
The TwinCAT Speech Configurator allows you to test and identify sound cards from the target system for later use as a microphone for ASR or as a playback device for TTS.
The devices are detected and also identified via the Windows operating system, so the configuration must normally be specific to a 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.



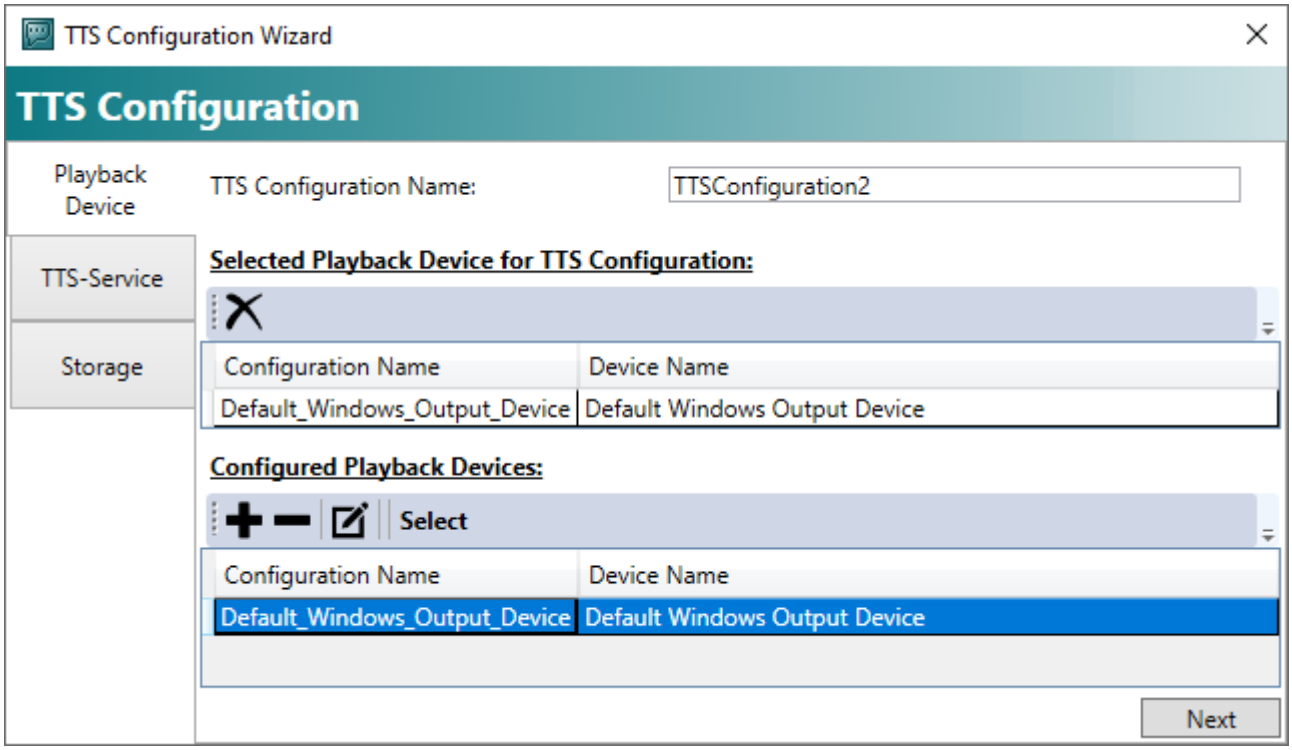
3. Click the **+** button or use the  button for an existing device.
   ⇨ The device wizard opens.
4. In the drop-down box, select the microphone that you want to test.
5. To test the microphone, click the  button.



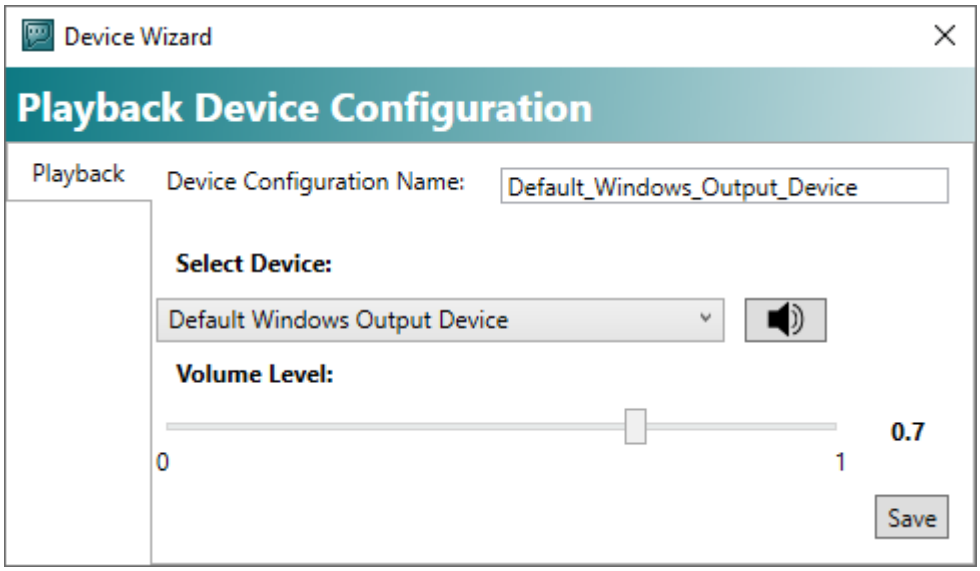
⇨ On clicking the  button, the recording is started and played back immediately afterwards.

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.

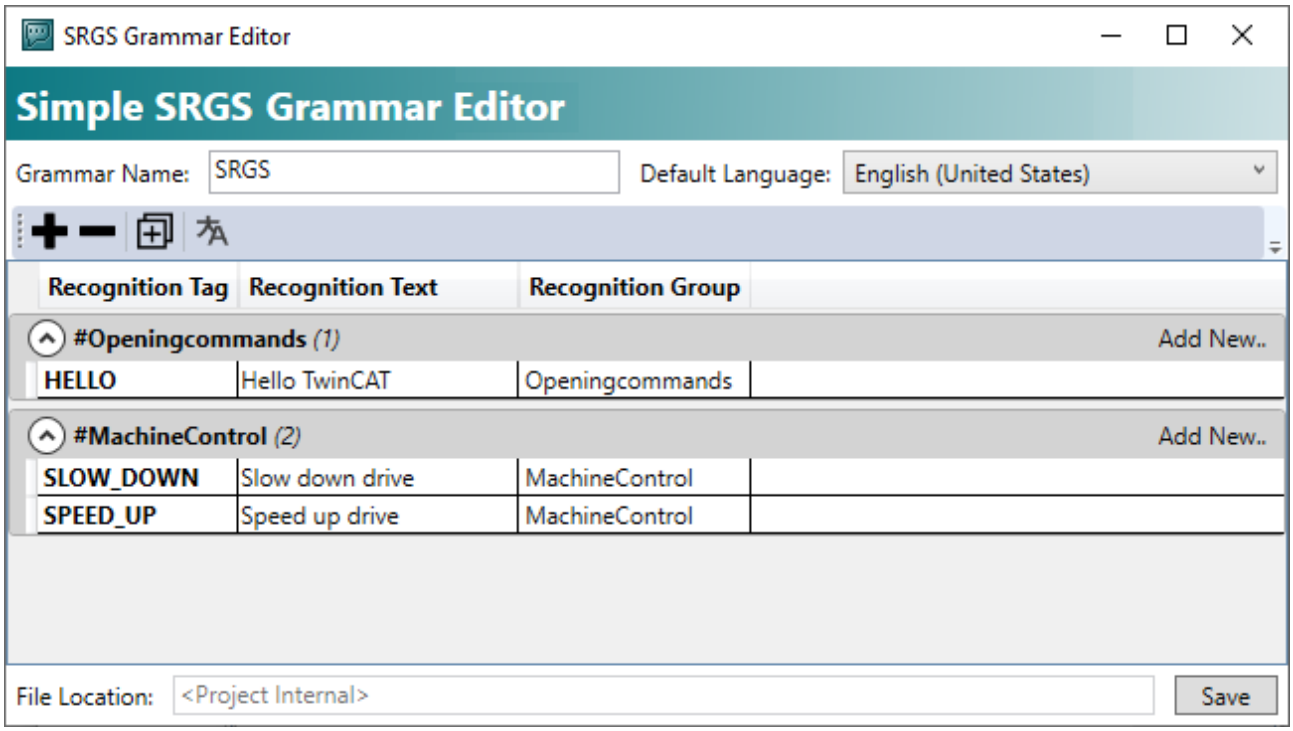
On the other hand, external SRGSs, which are created with other editors and are more complex, can also be imported. An example of this use is Sample 03: EventLogger [▶ 66].

Some functions, such as FB_SetRuleState [▶ 60] 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

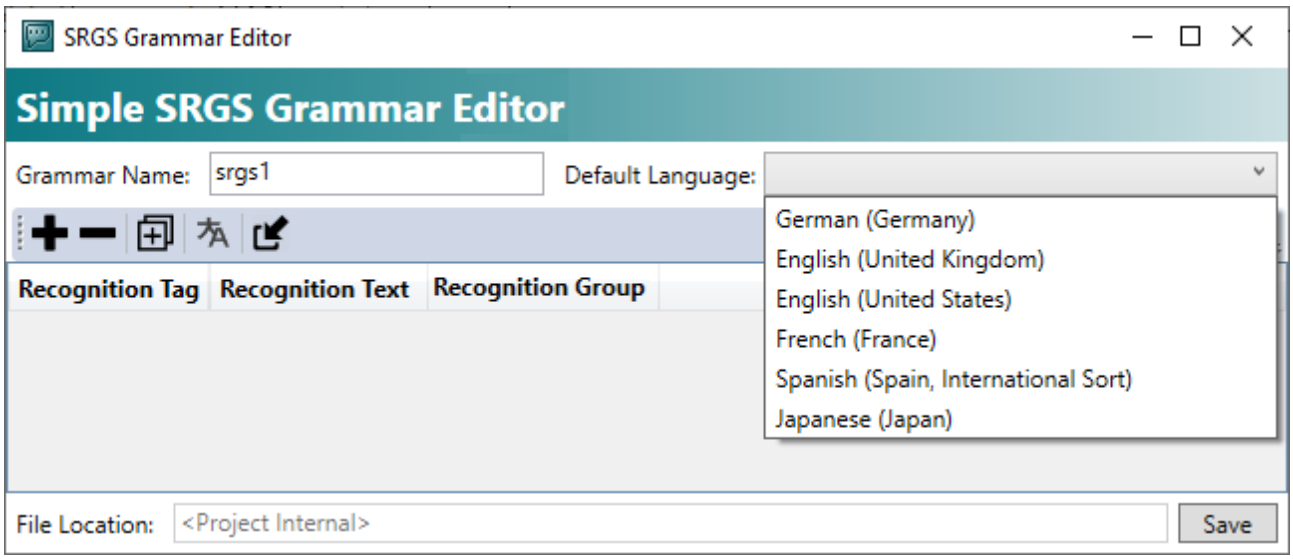
The SRGS file is provided with the ASR service, which is typically created by the ASR wizard. Here you can either use the Simple SRGS Editor to edit a local SRGS file or reference an externally provided SRGS.

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 [▶ 60].

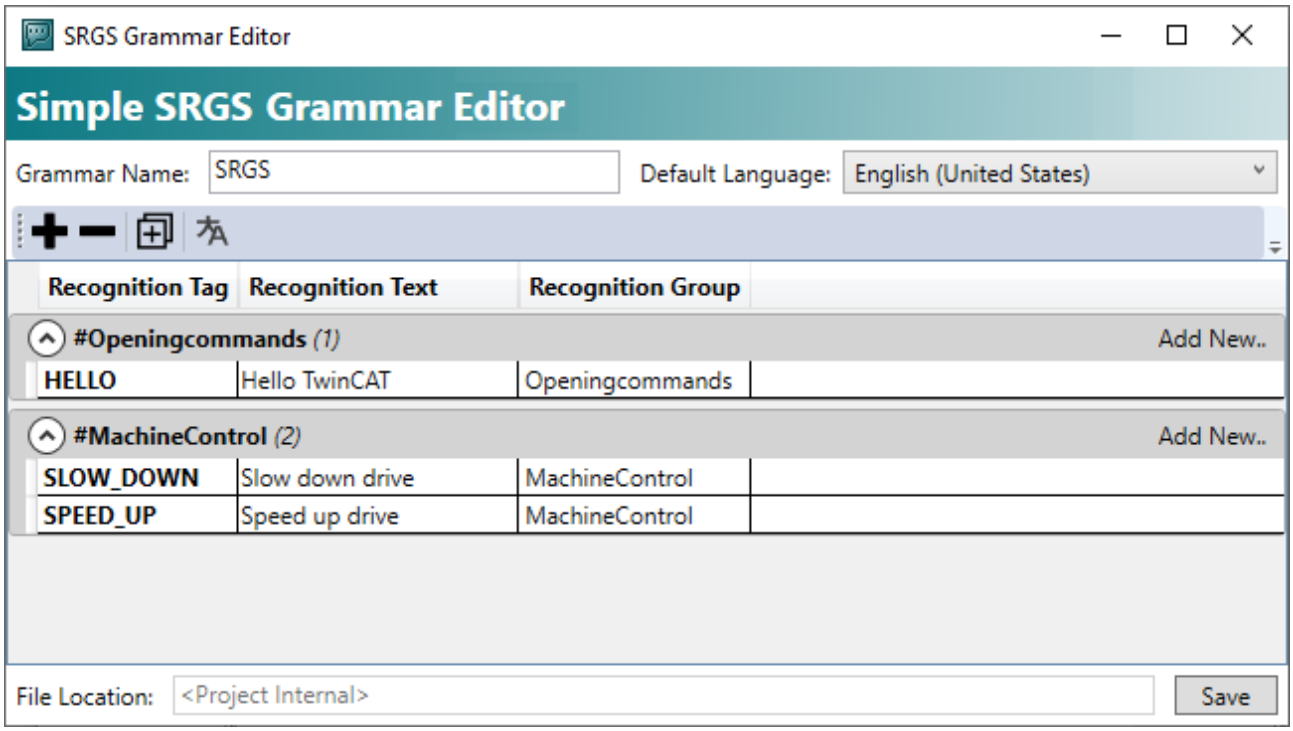


Procedure with the Simple SRGS Editor

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 [▶ 38] 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.

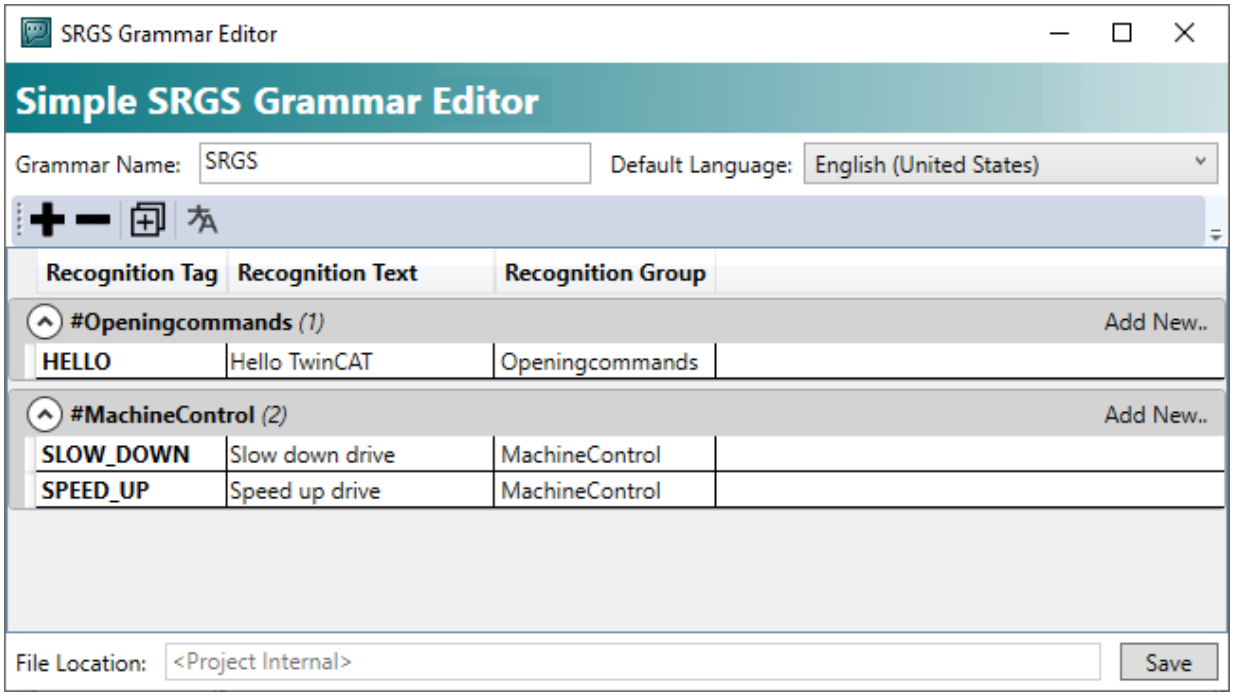
⇨ The SRGS file is transferred to the target by activating the TwinCAT Speech Configurator project.

Entering voice commands in different languages

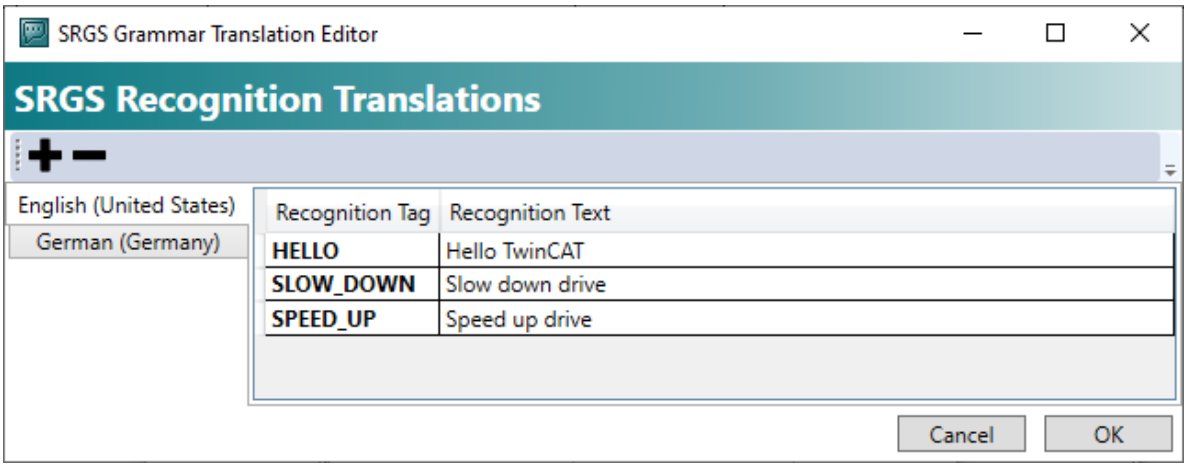
The Simple SRGS Editor can be used to recognize multilingual texts.

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.



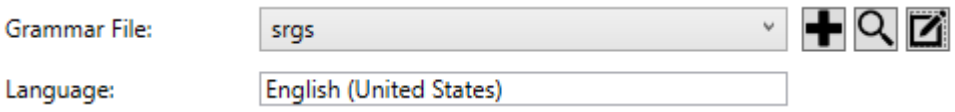
⇨ When the editor is opened, all Recognition Tags already defined are displayed.



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.



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.

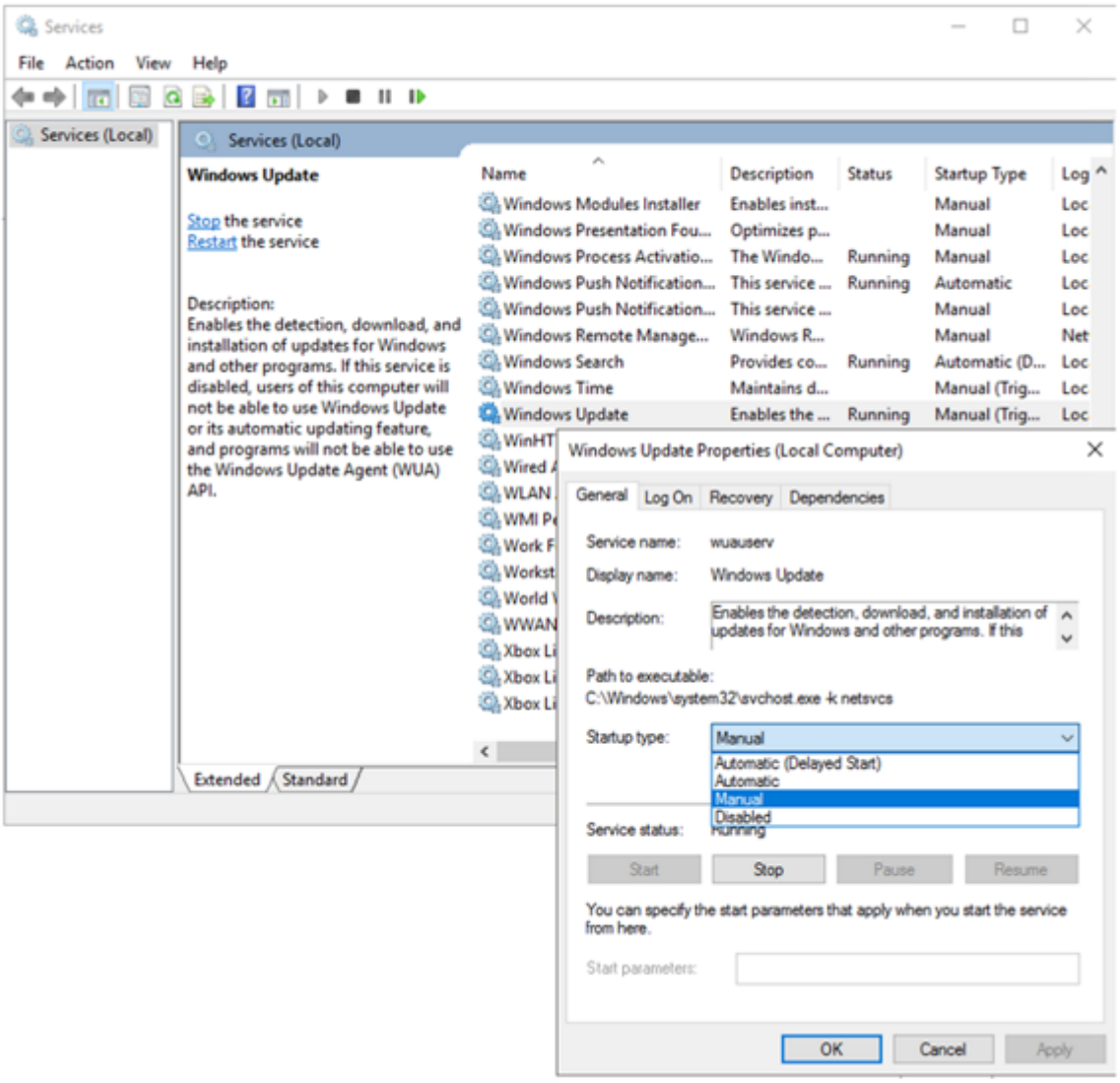
Proceed as follows to install a language:

✓ The controller must be able to communicate with the Microsoft server via the network.

1. Open **Windows Settings>Time and Language>Language**.
2. Click the **Add Preferred Language** button.
3. Select a language and click **Next**.
4. Select whether you only want to install the language pack or also set it as the display language.
5. Click **Install**.
   ⇨ The language pack is installed.
6. Click the installed language.
7. Click on **Settings**.

⇨ 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:



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

Name	Description
TCS_ENGINE_UNKNOWN	ASR-Engine in unknown state.
TCS_ENGINE_STARTING	ASR-Engine is starting up.
TCS_ENGINE_STARTED	ASR-Engine is started.
TCS_ENGINE_STOPPING	ASR-Engine is stopping.
TCS_ENGINE_STOPPED	ASR-Engine is stopped.
TCS_ENGINE_FAILURE	ASR-Engine is in failure state. See ErrorId for details.
TCS_ENGINE_PAUSED	ASR-Engine is paused.
TCS_ENGINE_INACTIVE	ASR-Engine is inactive.

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,
```

```
    TCS_CMD_TTS_SYNTH_ERROR                                          := 30030,
    TCS_CMD_TTS_SYNTH_ERROR_CONFIGURATION_DETAILS                    := 30040,
    TCS_CMD_TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_LANGUAGEID_MISSING  := 30050,
    TCS_CMD_TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_LANGUAGEID_NOTSUPPORTED := 30060,
    TCS_CMD_TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_INCOMPLETE          := 30070,
    TCS_CMD_TTS_SYNTH_ERROR_UTTERANCE_EMPTY                          := 30080,
    TCS_CMD_TTS_SYNTH_ERROR_UTTERANCE_AND_PATH_EMPTY                 := 30090,
    TCS_CMD_TTS_SYNTH_ERROR_SAPI_                                    := 31000,
    TCS_CMD_TTS_SYNTH_ERROR_AWSPOLLY_                                := 32000,
    TCS_CMD_AUDIO_PLAYBACK_START_FAILURE                             := 40010,
    TCS_CMD_AUDIO_PLAYBACK_START_FAILURE_INVALIDFORMAT               := 40011,
    TCS_CMD_AUDIO_PLAYBACK_START_FAILURE_FILENOTFOUND                := 40012
)UDINT;
END_TYPE
```

Values

Name	Description
TCS_CMD_SUCCEEDED	Speech command succeeded.
TCS_CMD_REQUIRED_SERVICE_NOT_FOUND	Referenced service not found. Wrong Id configured?
TCS_CMD_REQUIRED_SERVICE_NOT_INITIALIZED	Referenced service not initialized.
TCS_CMD_INSUFFICIENT_PARAMETERS	Speech command with insufficient parameters.
TCS_CMD_ASR_RULES_SETFAILED	Speech command SetRuleState failed.
TCS_CMD_ASR_INSTANTIATION_FAILURE	Speech engine could not be instantiated.
TCS_CMD_ASR_STOP_FAILURE	Speech engine could not be stopped.
TCS_CMD_ASR_START_FAILURE	Speech engine could not be started.
TCS_CMD_ASR_PAUSE_FAILURE	Speech engine could not be paused.
TCS_CMD_TTS_STATUS_REQUEST_FAILURE	Speech engine's status request failed.
TCS_CMD_TTS_STOP_REQUEST_FAILURE	Speech engine's stop request failed.
TCS_CMD_TTS_SYNTH_ERROR	Generic speech synthesis related error.
TCS_CMD_TTS_SYNTH_ERROR_CONFIGURATION_DETAILS	Speech synthesis configuration related error.
TCS_CMD_TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_LANGUAGEID_MISSING	LanguageId is missing.
TCS_CMD_TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_LANGUAGEID_NOTSUPPORTED	LanguageId is not supported.
TCS_CMD_TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_INCOMPLETE	Speech synthesis configuration is incomplete.
TCS_CMD_TTS_SYNTH_ERROR_UTTERANCE_EMPTY	No utterance is specified.
TCS_CMD_TTS_SYNTH_ERROR_UTTERANCE_AND_PATH_EMPTY	No utterance or path is specified.
TCS_CMD_TTS_SYNTH_ERROR_SAPI_	Speech synthesis with error of underlying SAPI engine.
TCS_CMD_TTS_SYNTH_ERROR_AWSPOLLY_	Speech synthesis with error of underlying Amazon Polly engine.
TCS_CMD_AUDIO_PLAYBACK_START_FAILURE	Audio playback could not be started.
TCS_CMD_AUDIO_PLAYBACK_START_FAILURE_INVALIDFORMAT	Audio playback could not be started. Invalid audio format.
TCS_CMD_AUDIO_PLAYBACK_START_FAILURE_FILENOTFOUND	Audio playback could not be started.File not found on target system.

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

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

Name	Type	Description
bPlay	BOOL	Rising flag triggers play command.
sPath	STRING	Path to audio file on target system. 16khz sample rate, 16 bit sample size, 1 channel expected.
nConfigurationId	UINT	ConfigurationId of configuration to be used in play request.

Outputs

Name	Type	Description
bBusy	BOOL	Function block is busy.
bError	BOOL	Function block is in error state.
nErrorId	ETcsSpeechCommandExitCode [▶ 56]	ErrorId providing details of the occurred error.
nLastCommandExitCode	UINT	Returncode of last executed command.
nPlaybackPosition	ULINT	Current playback position in file playback (milliseconds).
nPlaybackTotal	ULINT	Total length of file playback (milliseconds).

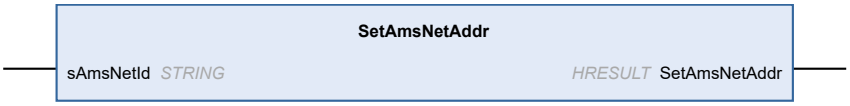
Methods

Name	Description
SetAmsNetAddr [▶ 59]	Method to be called once to specifiy the AmsNetId of destination TcSpeech service.

Required License

TC3 Speech

9.2.2.1 SetAmsNetAddr



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

Syntax

Definition:

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

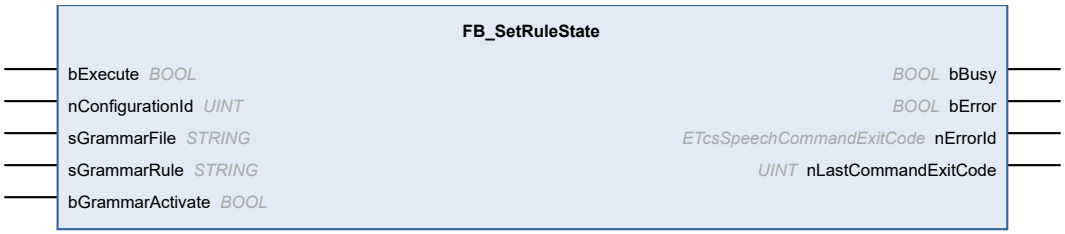
Inputs

Name	Type	Description
sAmsNetId	STRING	AmsNetId of destination TcSpeech service.

Return value

HRESULT

9.2.3 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

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

Outputs

Name	Type	Description
bBusy	BOOL	Function block is busy.
bError	BOOL	Function block is in error state.
nErrorId	ETcsSpeechCommandExitCode [▶ 56]	ErrorId providing details of the occurred error.
nLastCommandExitCode	UINT	Returncode of last executed command.

Methods

Name	Description
SetAmsNetAddr [▶ 61]	Method to be called once to specifiy the AmsNetId of destination TcSpeech service.

Required License

TC3 Speech

9.2.3.1 SetAmsNetAddr

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

Syntax

Definition:

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

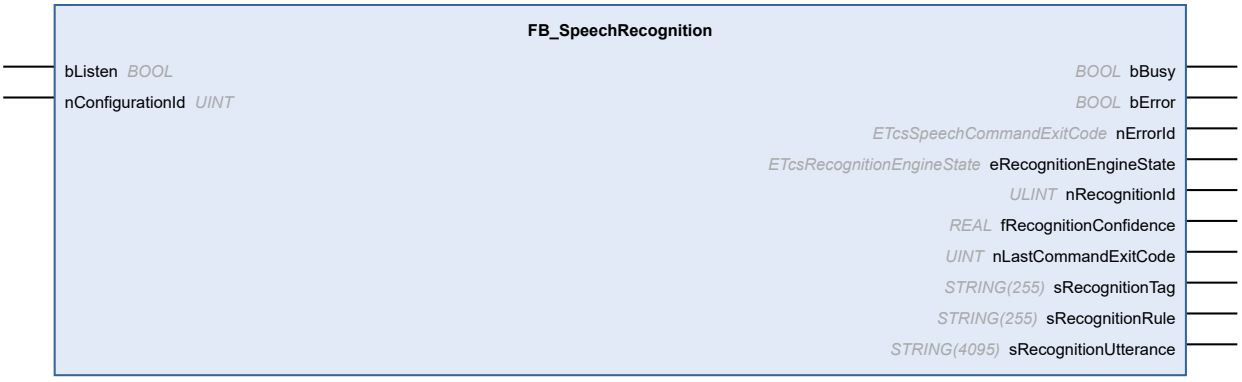
Inputs

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;
```

```
    nLastCommandExitCode    : UINT;
    sRecognitionTag         : STRING(255);
    sRecognitionRule        : STRING(255);
    sRecognitionUtterance   : STRING(4095);
END_VAR
```

Inputs

Name	Type	Description
bListen	BOOL	Rising edge triggers listening.
nConfigurationId	UINT	ConfigurationId of configuration to be used in listen request.

Outputs

Name	Type	Description
bBusy	BOOL	Function block is busy.
bError	BOOL	Function block is in error state.
nErrorId	ETcsSpeechCommandExitCode [▶ 56]	ErrorId providing details of the occurred error.
eRecognitionEngineState	ETcsRecognitionEngineState [▶ 56]	Current state of the recognition engine.
nRecognitionId	ULINT	Last speech recognition (RecognitionId)
fRecognitionConfidence	REAL	Last recognition's confidence level.
nLastCommandExitCode	UINT	Returncode of last executed command.
sRecognitionTag	STRING(255)	Last recognition's tag.
sRecognitionRule	STRING(255)	Last recognition's triggered rule.
sRecognitionUtterance	STRING(4095)	Last recognition's utterance.

Methods

Name	Description
SetAmsNetAddr [▶ 62]	Method to be called once to specifiy the AmsNetId of destination TcSpeech service.

Required License

TC3 Speech

9.2.4.1 SetAmsNetAddr

Method to be called once to specifiy 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

9.2.5 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    : ULINT;
    nPlaybackTotal       : ULINT;
END_VAR
```

Inputs

Name	Type	Description
bSpeak	BOOL	Rising edge triggers speak command.
sUtterance	STRING	Utterance to speak.
nConfigurationId	UINT	ConfigurationId to be used (cmp. TcSpeech Configurator).
nLanguageId	UINT	LanguageId to be used in speech synthesis request. '0' defaults to DefaultLanguageId specified in configuration.

Outputs

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

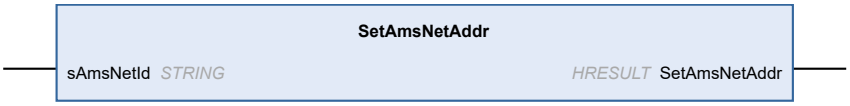
Methods

Name	Description
SetAmsNetAddr [▶ 64]	Method to be called once to specifiy the AmsNetId of destination TcSpeech service.

Required License

TC3 Speech

9.2.5.1 SetAmsNetAddr



Method to be called once to specifiy 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

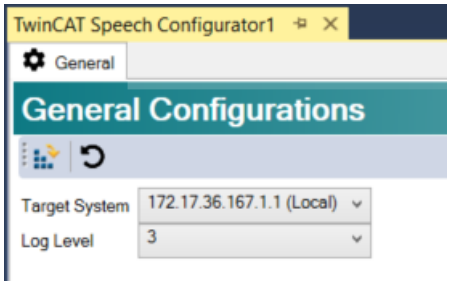
10 Samples

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) [▶ 27] Quick start: speech input (ASR) [▶ 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. ***Note*** **Activate the TwinCAT Speech configuration.**
Sample project	https://infosys.beckhoff.com/content/1033/TF4500_TC3_TwinCAT_Speech/Resources/zip/10037206155.zip
Further information	Configuring a grammar file [▶ 52]

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. ***Note*** **Activate the TwinCAT Speech configuration.**
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
Further information	FB_Play

Also see about this

FB_Play [▶ 58]

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
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. ***Note*** **Activate the TwinCAT Speech configuration.**
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.

Hex	Dec	Description	Reason
	0	SUCCEEDED	The call was successful
	1	GENERIC_ERROR	The call ended in a generic error that is not specified in more detail. Details in the LogFile.
	100	ASR_NO_RECOGNITION	The ASR did not find any recognition.
	200	TTS_SYNTH_WARNING_VOICENOTFOUND	The configured language could not be used.
	10000	REQUIRED_SERVICE_NOT_FOUND	TwinCAT Speech internal: A required service could not be found.
	10001	REQUIRED_SERVICE_NOT_INITIALIZED	TwinCAT Speech internal: A required service is not initialized.
	10100	INSUFFICIENT_PARAMETERS	TwinCAT Speech internal: A call was made with incorrect parameters
	20000	ASR_RULES_SETFAILED	The ASR was unable to activate the rules created in the SRGS.
	20010	ASR_INSTANTIATION_FAILURE	An ASR could not be initialized. There may be a configuration error.
	20020	ASR_STOP_FAILURE	An ASR could not be stopped.
	20030	ASR_START_FAILURE	An ASR could not be started.
	30010	TTS_STATUS_REQUEST_FAILURE	A TTS call failed.
	30020	TTS_STOP_REQUEST_FAILURE	A TTS call could not be interrupted.
	30030	TTS_SYNTH_ERROR	A TTS call could not be synthesized.
	30040	TTS_SYNTH_ERROR_CONFIGURATION_DETAILS	A TTS call could not be synthesized because a configuration is faulty.
	30050	TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_LANGUAGEID_MISSING	A TTS call could not be synthesized because the configuration includes an incorrect language ID.
	30060	TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_LANGUAGEID_NOTSUPPORTED	A TTS call could not be synthesized because the configuration includes an unsupported language ID.
	30070	TTS_SYNTH_ERROR_CONFIGURATION_DETAILS_INCOMPLETE	A TTS call could not be synthesized because the configuration is incomplete.
	30080	TTS_SYNTH_ERROR_UTTERANCE_EMPTY	A TTS call could not be synthesized because the input is empty.
	31000	TTS_SYNTH_ERROR_SAPI	A TTS call could not be synthesized because the underlying Microsoft SAPI reported an error. Details in the LogFile
	32000	TTS_SYNTH_ERROR_AWSPOLLY	A TTS call could not be synthesized because the underlying AWS Polly interface reported an error. Details in the LogFile
	40010	AUDIO_PLAYBACK_START_FAILURE	Playback could not be started. Another playback may have just started.

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

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com