

0 - Introduction	2
1 - Preparations	5
2 - Setup GSE Action	7
3.1 - Usage	14
3.2 - Trouble Shooting	20
A.1 - Example - Announce Text	24
A.2 - Example - Translate & Announce Text	27
A.3 - More examples	31

AzureTTS (text-to-speech)

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

3.1 - Usage

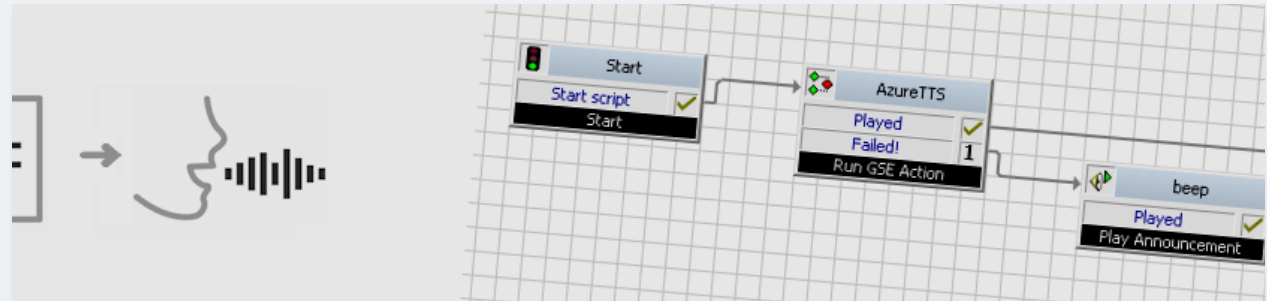
3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Announce Text

A.2 - Example: Translate & Announce Text

A.3 - More examples...



AzureTTS - 0 - Introduction

Followers

0

VBScript

Lua

This extension provides **text to speech** functionality for the SwyxWare call routing. It

- uses the Microsoft Azure Cognitive Speech services
- comes for **VBscript based** and **Lua based** call routing
- requires **SwyxWare 14.10** (or newer)

To handle the text to speech functionality of the Cognitive Speech services it uses the [Text to speech REST API](#).



Please refer to the [Forums](#) to discuss the AzureTTS extension or for support requests.



Please find the download for this project [here](#).



For the complete documentation explaining the setup, usage and all included examples just read the following chapters from the menu on the left.

As with all other Swyx Forum Open Source Projects, Support is **EXCLUSIVELY** provided in the Project Froum (see link above).

Lua based call routing has been introduced to SwyxWare in version **13.10** on. As of the time of the release of the AzureTTS extension the Lua based call routing is still in **BETA state** and should not be used in productive environments. However, this project already comes also for **Lua based** call routing as an example of its ease of use.

License

AzureTTS
v1.0.0

This is a Swyx Forum Open Source Project.

<https://www.swyxforum.com/projects/>

<https://www.swyxforum.com/azuretts/introduction/0-introduction-r1/>

The MIT License (MIT)

Copyright (c) 2024 by Swyx Forum

Copyright (c) 2024 by Tom Wellige

All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice must be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



By Tom Wellige

June 17, 2023

 Share

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige
Powered by Invision Community

AzureTTS (text-to-speech)

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

3.1 - Usage

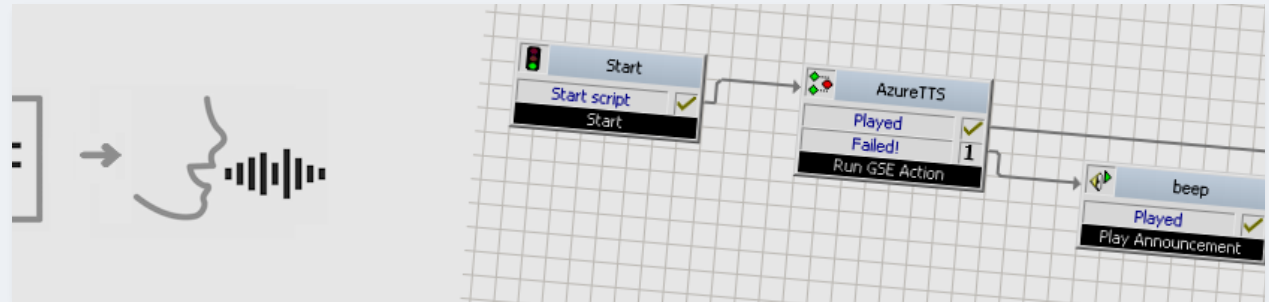
3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Announce Text

A.2 - Example: Translate & Announce Text

A.3 - More examples...



AzureTTS - 1 - Preparations

Followers

0

VBScript

Lua

[Download](#) the latest version of this project.

Additionally you need an account on the [Microsoft Azure Portal](#) and a **Speech subscription** (a free of charge pricing tier is available - Free F0). The following link explains in detail how to create such a subscription:

- [How to get Microsoft Azure Text-to-Speech API key.](#)

In **Step 3** in this description you see a "**View full pricing details**" link. There you will get an overview of all available options. At the time the AzureTTS extension was developed the "**Free F0**" tier included 5000 free text-to-speech conversions per months. Microsoft might

change the pricing scheme in the future. Make sure to select the proper scheme/tier to your needs.

Step 6 shows how to obtain your **KEY 1**. This is the **subscription key** which is needed for the AzureTTS extension.



By Tom Wellige

June 17, 2023

 Share

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

AzureTTS (text-to-speech)

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

3.1 - Usage

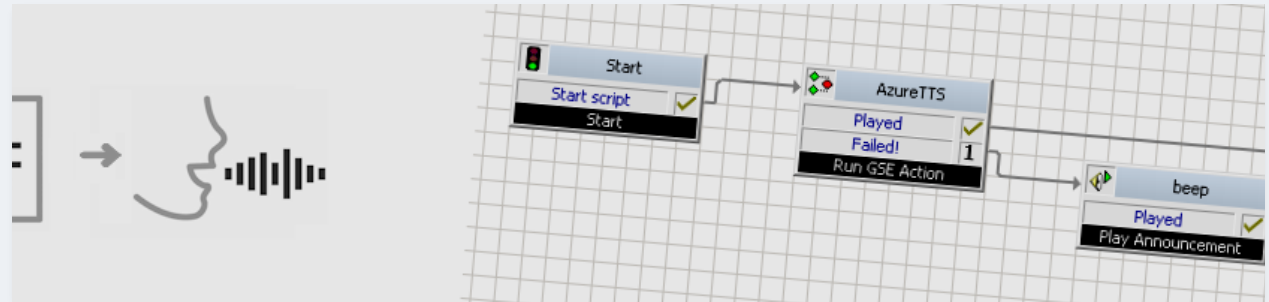
3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Announce Text

A.2 - Example: Translate & Announce Text

A.3 - More examples...



AzureTTS - 2 - Setup GSE Action

Followers

0

VBScript

Lua

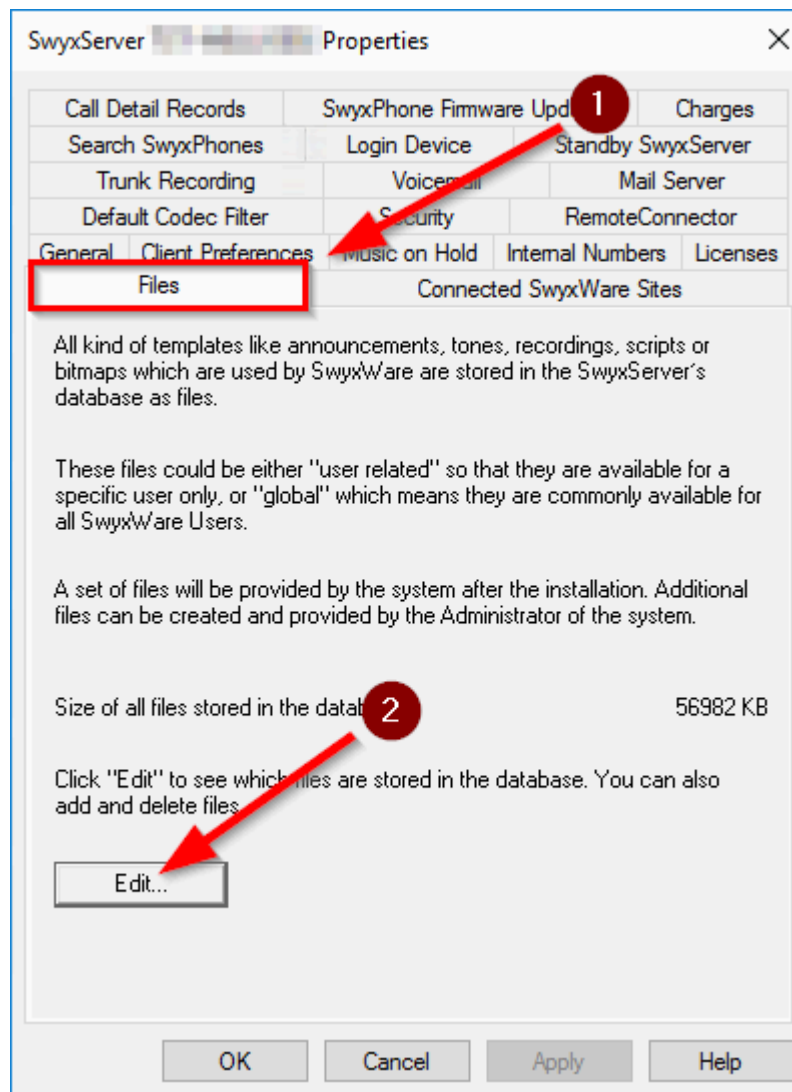
The **AzureTTS** extension is designed as GSE action. To install the GSE action you need to use the **SwyxWare Administration**.

Step 1

Open the SwyxWare Administration and open the properties of your Swyx Server.

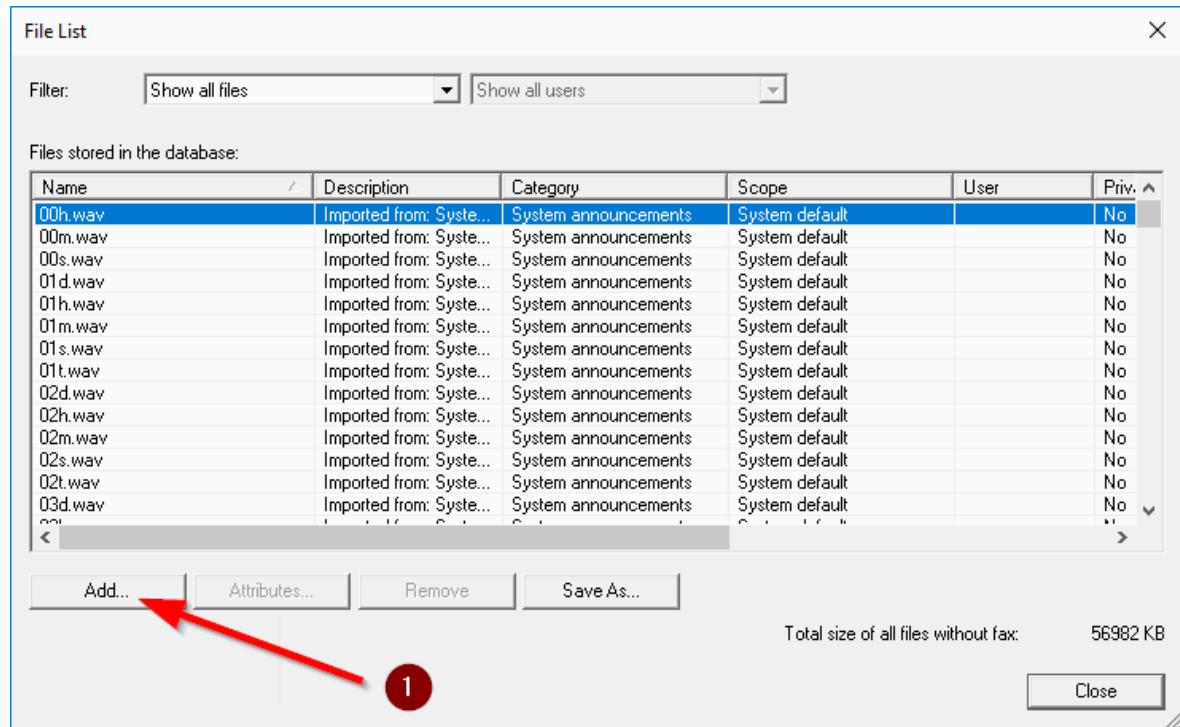
Step 2

Switch to the **Files** page and click on **Edit....**



Step 3

Click on **Add...**



Step 4.1 - VBScript usage

1. Select all files from the **VBScript based\ase** folder from the download package.
2. Select **Global** as **Scope**. (you can also select User to load the files into the user scope a your script user)
3. Select **Call Routing VBS scripts** as **Category**.
4. It is recommended to enter **AzureTTS** into the **Description** field, but not necessary.

The screenshot shows a Windows-style dialog box titled "Add File to Database" with a close button (X) in the top right corner. The dialog contains several input fields and dropdown menus, with four red circular callouts numbered 1 through 4 pointing to specific elements:

- Callout 1:** Points to the "File to add:" text box, which contains the path "C:\Projects\AzureTTS\VBScript based\ase\action..." and a browse button (...).
- Callout 2:** Points to the "Scope:" dropdown menu, which is set to "Global".
- Callout 3:** Points to the "Category:" dropdown menu, which is set to "Call Routing VBS scripts".
- Callout 4:** Points to the "Description:" text box, which contains the text "AzureTTS".

Other fields in the dialog include:

- Name:** A text box containing "actionAzureTTS.ase; actionAzureTT:".
- User:** A dropdown menu set to "Conference".
- Group:** A dropdown menu set to "Everyone".
- File Properties:** A section with three unchecked checkboxes: "Private", "Hidden", and "System".
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

Step 4.2 - Lua usage

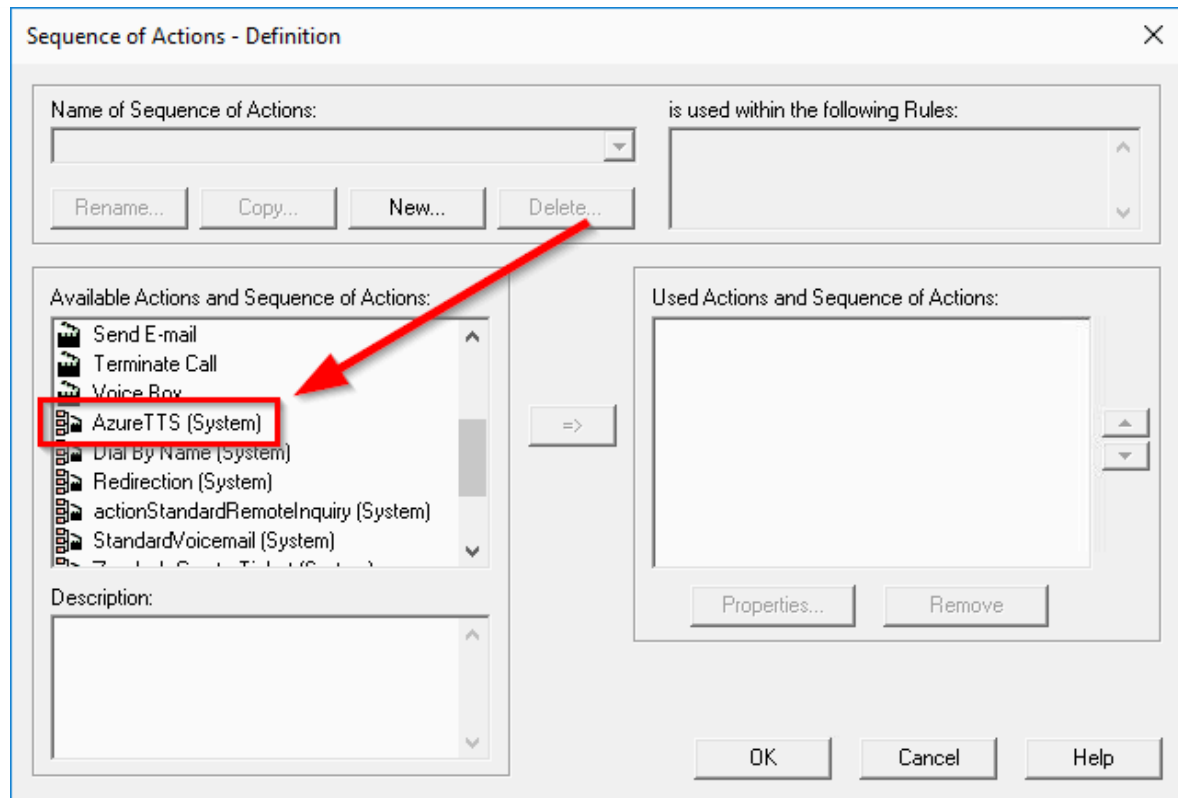
1. Select all files from the **Lua based\ase** folder from the download package.
2. Select **Global** as **Scope**. (you can also select User to load the files into the user scope a your script user)
3. Select **Call Routing Lua scripts** as **Category**.
4. It is recommended to enter **AzureTTS** into the **Description** field, but not necessary.

The screenshot shows a 'Add File to Database' dialog box with the following fields and callouts:

- File to add:** A text field containing the path 'C:\Projects\AzureTTS\Lua based\ase\actionAzur' with a red circle '1' next to the file explorer button.
- Name:** A text field containing 'actionAzureTTS.ase; actionAzureTT:'.
- Scope:** A dropdown menu set to 'Global' with a red circle '2' next to it.
- Category:** A dropdown menu set to 'Call Routing Lua scripts' with a red circle '3' next to it.
- User:** A dropdown menu set to 'Conference'.
- Group:** A dropdown menu set to 'Everyone'.
- File Properties:** A section with three unchecked checkboxes: 'Private', 'Hidden', and 'System'.
- Description:** A text field containing 'AzureTTS' with a red circle '4' next to it.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

Step 5

To check if the GSE action is available within call routing scripts open the **Call Routing Manager** of any user and click the button **Sequence of Actions**. Scroll the list on the left side down until you reach the **AzureTTS** action. The **(System)** behind the GSE action name shows that this is a global action being available for every SwyxWare user.



The AzureTTS action is now installed and can be fully used within GSE call routing scripts.



By Tom Wellige
June 17, 2023

 Share

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

AzureTTS (text-to-speech)

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

3.1 - Usage

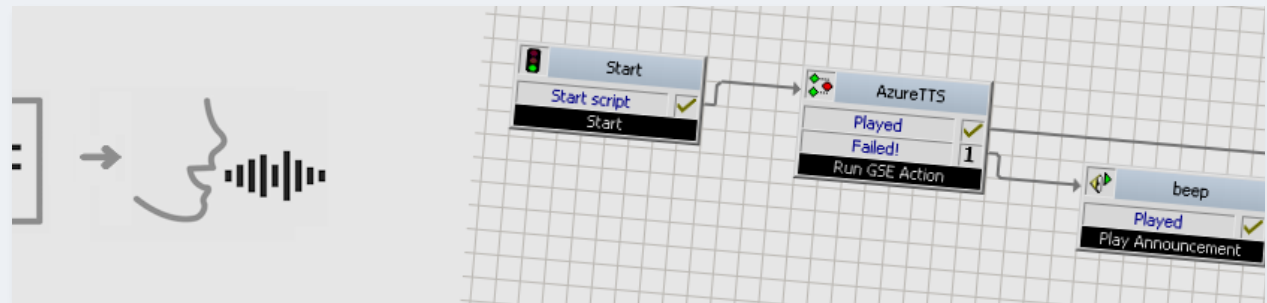
3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Announce Text

A.2 - Example: Translate & Announce Text

A.3 - More examples...



AzureTTS - 3.1 - Usage

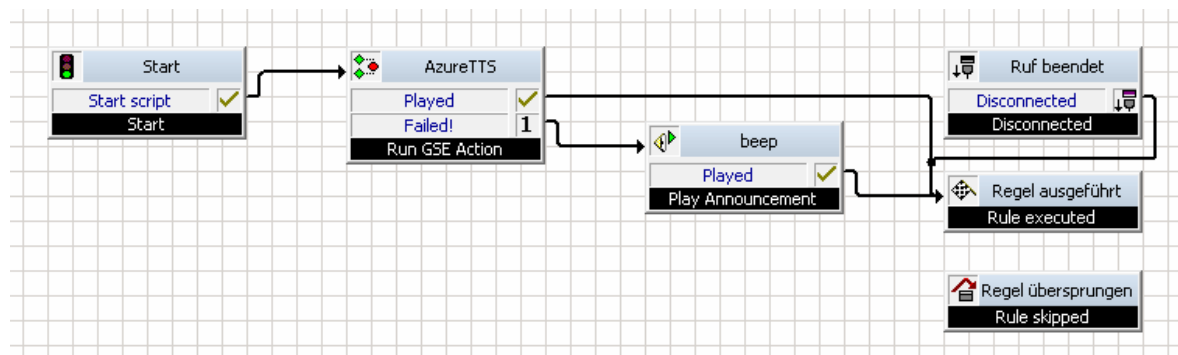
Followers

0

VBScript

Lua

An example call routing script can be found in [A.1 - Example: Announce Text](#).



Configure action parameters

Run GSE Action Properties

General Parameters Links

Select GSE action: Azure TTS (System)

Set action parameters:

Name	Value	Default Value
Region	westeurope	westeurope
SubscriptionKey	= ""	= ""
Text	= ""	= ""
Language	de-DE	de-DE
Voice	de-DE-ConradNeural	de-DE-ConradNeural
Gender	Male	Male

Edit Parameter...

Description:

AzureTSS - Announces a given text (v1.0.0)

Return Values:

0 - Played
1 - Failed!

OK Cancel Help

By double clicking a parameter in the list, you can edit it.

Region

This is the configured region of your speech subscription. A list of available values can be found at the [Text to speech REST API](#) page under "Prebuilt neural voices".

SubscriptionKey

This is the subscription key of your speech subscription. Please read under [1 - Preparations](#) how to obtain it.

Text

This is the text you want to get announced. Make sure that the language of the text matches the following "Language" parameter.

Language

This is the language of the above given "Text". You need to enter a BCP 47 language tag, like "de-DE", "en-GB", "fr-FR". This value can also be taken from the beginning of the "Voice" parameter.

Voice

This is the voice your text should be announced in. A list of all available voices can be found in the Azure Speech Voice Gallery. There you a huge selection of different voices per language and gender.

Once you have chosen a voice, you need to switch to "Samplecode" in the right box and grab the "SpeechSynthesisVoiceName" from there, e.g. "en-GB-SoniaNeural" or "de-DE-ConradNeural".

Gender

This is the gender of the above chosen voice. Valid values are "Male" or "Female".

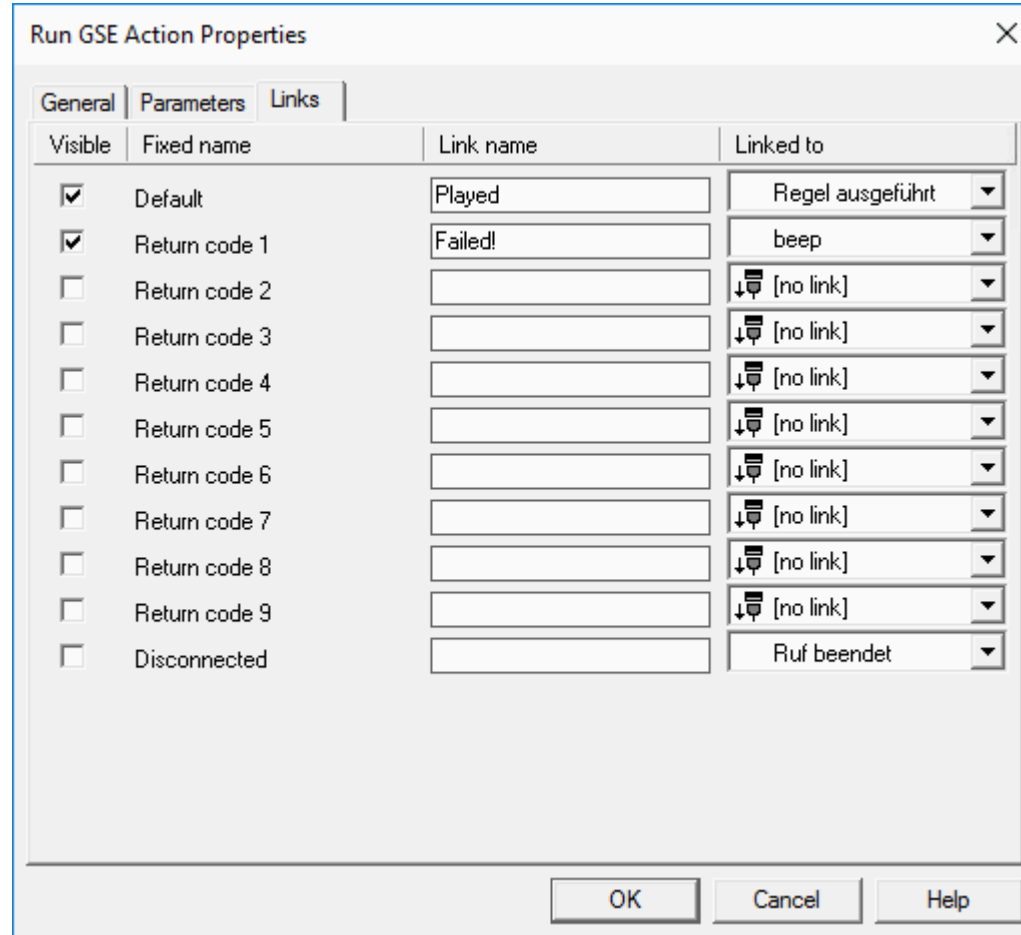
OwnSSML

The AzureTTS extension uses the above parameters (Text, Language, Voice, Gender) to generate some basic so called "Speech Synthesis Markup Language (SSML)," which is passed the the Azure Speech system.

SSML specifies in detail how the speech is getting generated. If you want to use your own SSML code instead of the very basic one the AzureTTS extension generates you can use this parameter.

Your own SSML code must comply to the Azure Speech SSML document structure and events. Once you use your own SSML code the above parameters (Text, Language, Voice, Gender) will be completely ignored and no SSML code will be generated anymore.

Configure action exits



Visible	Fixed name	Link name	Linked to
☒	Default	Played	Regel ausgeführt
☒	Return code 1	Failed!	beep
☐	Return code 2		↓ [no link]
☐	Return code 3		↓ [no link]
☐	Return code 4		↓ [no link]
☐	Return code 5		↓ [no link]
☐	Return code 6		↓ [no link]
☐	Return code 7		↓ [no link]
☐	Return code 8		↓ [no link]
☐	Return code 9		↓ [no link]
☐	Disconnected		Ruf beendet

OK Cancel Help

Exit 0 (Default)

This exit will be reached when everything worked fine and the given text has been announced. It is recommended to name this exit "**Played**".

Exit 1

This exit will be reached when there was any kind of problem and no text has been announced. It is recommended to name this exit "**Failed**".

If you reach this exit you can refer to [3.2 - Trouble Shooting](#) to figure what went wrong.

Additional return value (as global variable)

g_sAzureTTS_LatestWavFile

This global variable holds the name of the generated and played .wav file (path + filename).

From a Swyx Server perspective this is a temporary file which will be automatically deleted once the call ends (i.e. the call routing script finishes).

If you want to keep the .wav file for your own use, you need to copy it to another location **before** the call ends (just like you would have to do with announcements you recorded using the "Record Announcement" GSE block).

To do so, just use an **Insert Script Code** block right behind the **Played** exit to copy the generated temporary wav file to a permanent location.

This only works with the **VBScript** version of the extension.

```
Dim fso
Set fso = CreateObject("Scripting.FileSystemObject")
fso.CopyFile g_sAzureTTS_LatestWavFile, "C:\MyFiles\"
Set fso = Nothing
UseExit = 0
```



By Tom Wellige

June 17, 2023

 Share

Next Page >
3.2 - Trouble Shooting

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige
Powered by Invision Community

AzureTTS (text-to-speech)

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

3.1 - Usage

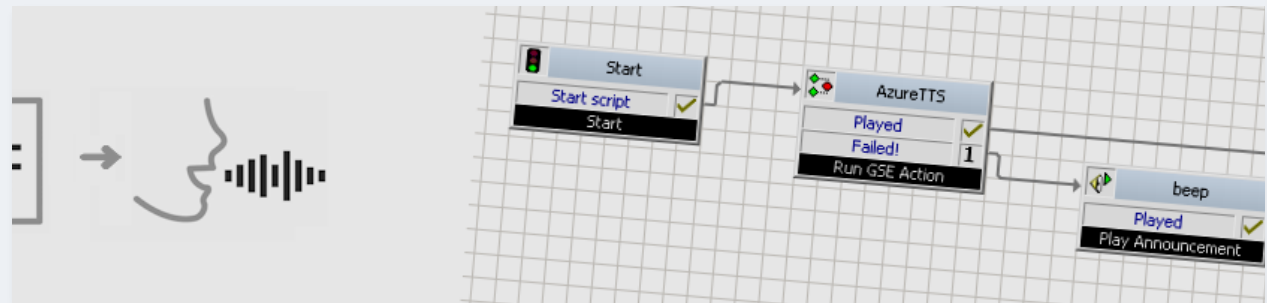
3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Announce Text

A.2 - Example: Translate & Announce Text

A.3 - More examples...



AzureTTS - 3.2 - Trouble Shooting

Followers

0

VBScript

Lua

In case the text to speech fails, for what reason ever, you will reach the **Failed** (1) exit of the GSE action.

The AzureTTS extension writes meaningful trace information into the server trace file, according to the trace recommendation from this blog post:

- [Don't be shy, be chatty!](#)

So, to figure the exact reason for the problem you need to take a look into that file. The following link explains where to find it and how to filter its content down to the call in question and call routing output only:

- [How to filter SwyxWare traces for call routing output of single call](#)

Once you have opened and filtered the trace file you need to search for the first trace lines written by the GSE action. Just look for these lines:

For **VBScript** based call routing:

```
-----> AzureTTS.Class_Initialize
```

For **Lua** based call routing:

```
-----> AzureTTS.Create
```

The following lines will contain information of all things happened, like setting the parameters, calling the Azure REST API to obtain an access token, calling the Azure REST API to generate a wav file, playing the wav file.

If at some point an error occurs, you will see an error message in the trace.

For example, if you forgot to enter your subscription key, you will see the following line:

```
Invalid sSubscriptionKey parameter! Must not be empty.
```

If the REST API call fails, for example because you entered an invalid subscription key, you will see the following line:

```
Azure web request failed! ({"error":{"code":"401","message":"Access denied
```

Make sure to provide a valid key for an active subscription and use a corr

In addition the the error message you will also find the original response code of the web request into the REST API. If everything went fine, the response code is **200**, otherwise it any other number (most likely ≥ 400), e.g.

```
nResponseCode = 401
```

By taking the detailed error information from the server trace file you should be able to fix your problem.

In case you can't get your problem fixed or have specific questions, please open your own topic in the [project forum](#).

Please understand that if you are using own SSML code and can't get it working, the project forum as also the project author aren't really available for help. You would have to check other online sources which are dedicated to creating SSML, preferably Azure SSML.



By Tom Wellige
June 17, 2023

 Share

[Previous Page](#)
[3.1 - Usage](#)

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige
Powered by Invision Community

AzureTTS (text-to-speech)

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

3.1 - Usage

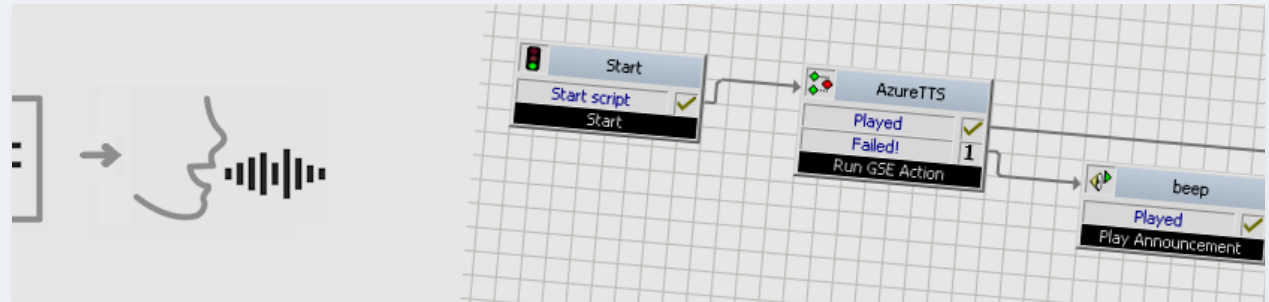
3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Announce Text

A.2 - Example: Translate & Announce Text

A.3 - More examples...



AzureTTS - A.1 - Example: Announce Text

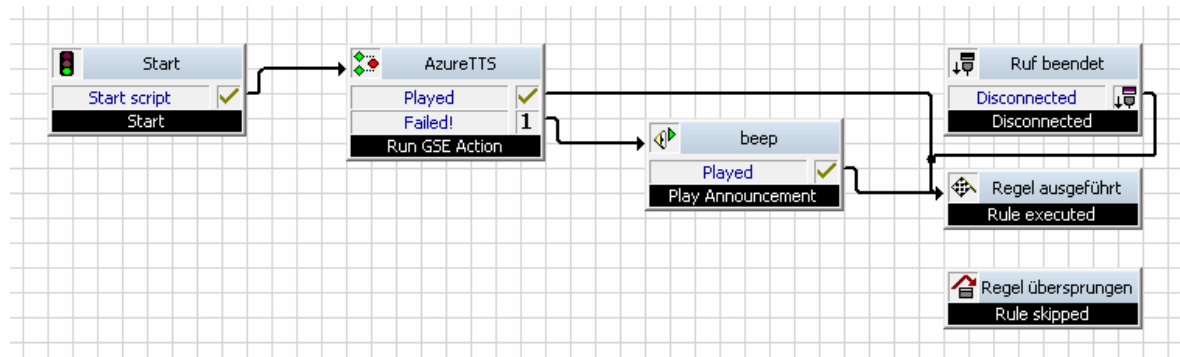
Followers 0

VBScript

Lua

This example demonstrates the usage of the **AzureTTS** GSE action.

It announces a given text. In case of an error (in the usage or execution of the AzureTTS action) it simply plays a beep.



To install it:

1. Open the **Call Routing Manager** of your desired SwyxWare user.
2. Click the "New Rule..." button.
3. Select "**Graphical Script Editor**" and click **Ok**.
4. Within the GSE open the **File | Import...** menu and click **No**.
5. From the download package select the following file:

For **VBScript** usage:

VBScript based\rse\Announce Text.rse

For **Lua** usage:

Lua based\rse\Announce Text.rse

6. You need to make some modifications in the **AzureTTS** block (at least your subscription key and of course the text you want to announce). They are explained in detail in chapter [3.1 - Usage](#).



By Tom Wellige

June 19, 2023

 Share

Next Page

[A.2 - Example: Translate & Announce Text](#)



Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

AzureTTS (text-to-speech)

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

3.1 - Usage

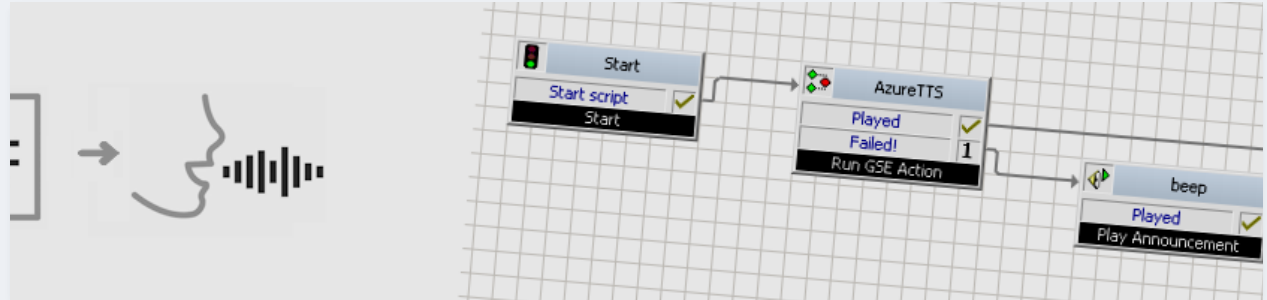
3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Announce Text

A.2 - Example: Translate & Announce Text

A.3 - More examples...



AzureTTS - A.2 - Example: Translate & Announce Text

Followers

0

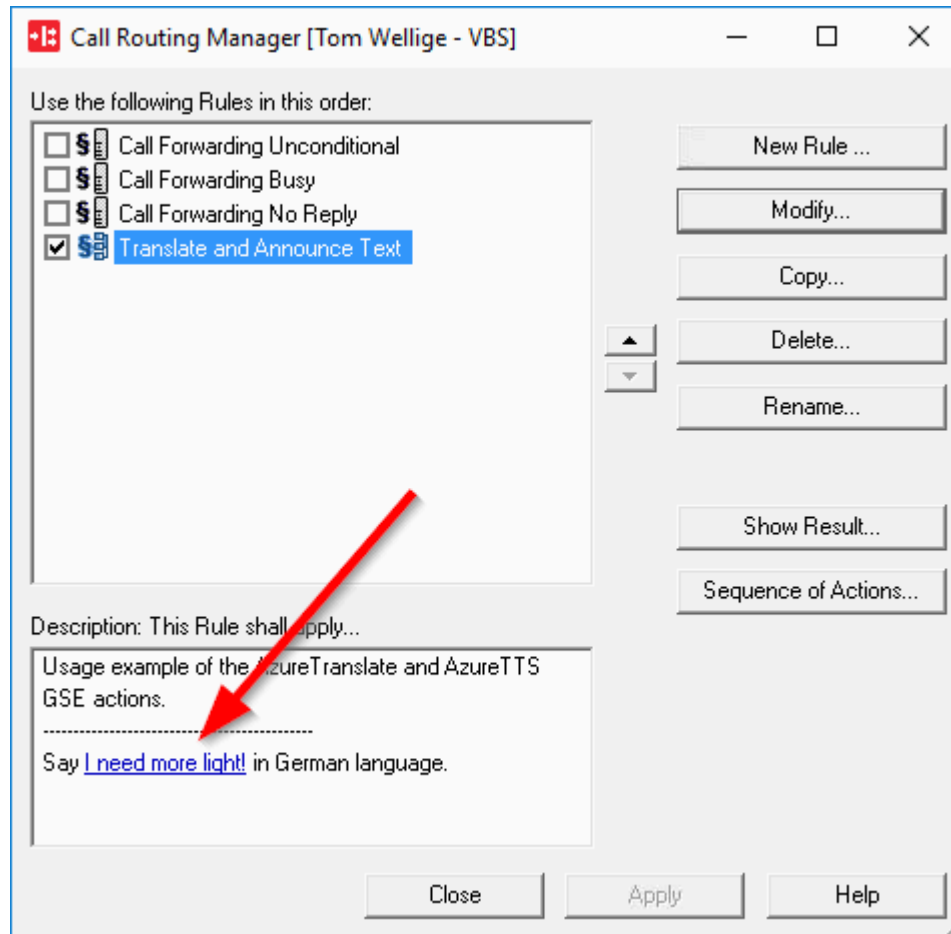
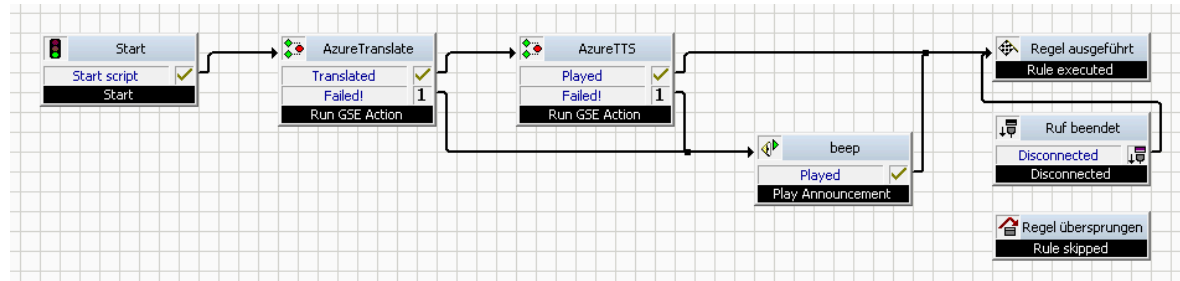
VBScript

Lua

This example demonstrates the usage of the **AzureTranslate** and **AzureTTS** GSE actions.

Please make sure that your SwyxWare matches the version requirements for both GSE actions.

It translates any given text in any language to German and announces it. In case of an error (in the usage or execution of the AzureTranslate and/or AzureTTS actions) it simply plays a beep. The text to translate and announce is passed as GSE rule parameter into the GSE rule.



To install it:

1. Follow the instructions [AzureTranslate - 2 - Setup GSE Action](#) to also install the **AzureTranslate** GSE action.
2. Open the **Call Routing Manager** of your desired SwyxWare user.
3. Click the "New Rule..." button.
4. Select "**Graphical Script Editor**" and click **Ok**.
5. Within the GSE open the **File | Import...** menu and click **No**.
6. From the download package select the following file:

For **VBScript** usage:

```
VBScript based\rse\Translate and Announce Text.rse
```

For **Lua** usage:

```
Lua based\rse\Translate and Announce Text.rse
```

7. You need to make some modifications in the **AzureTranslate** and **AzureTTS** blocks (e.g. your subscription keys for both services). They are explained in detail in the chapters [AzureTranslate - 3.1 - Usage](#) and [AzureTTS - 3.1 - Usage](#).



By Tom Wellige
June 23, 2023

 Share

< Previous Page
A.1 - Example: Announce Text

Next Page >
A.3 - More examples...

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige
Powered by Invision Community

AzureTTS (text-to-speech)

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

3.1 - Usage

3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Announce Text

A.2 - Example: Translate & Announce Text

A.3 - More examples...

AzureTTS - A.3 - More examples...

Followers

0

VBScript

Lua

There are more examples available showcasing the usage of **AzureTTS** extension. They are part of other **Open ECR Extensions** and are listed below:

- **Freshdesk Integration** A.5 - Example: Check and Announce Status
- **Freshservice Integration** A.5 - Example: Check and Announce Status
- **Jira Service Integration** A.5 - Example: Check and Announce Status
- **Zendesk Integration** A.6 - Example: Check and Announce Status



By Tom Wellige

[Share](#)

[Previous Page](#)
A.2 - Example: Translate & Announce Text

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige
Powered by Invision Community