

1 - Preparations	2
2 - Setup GSE Action	5
3.1 - Usage	12
3.2 - Trouble Shooting	17
A.1 - Example - Translate Text	20
A.2 - Example - Translate & Announce Text	23
0 - Introduction	27

AzureTranslate

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

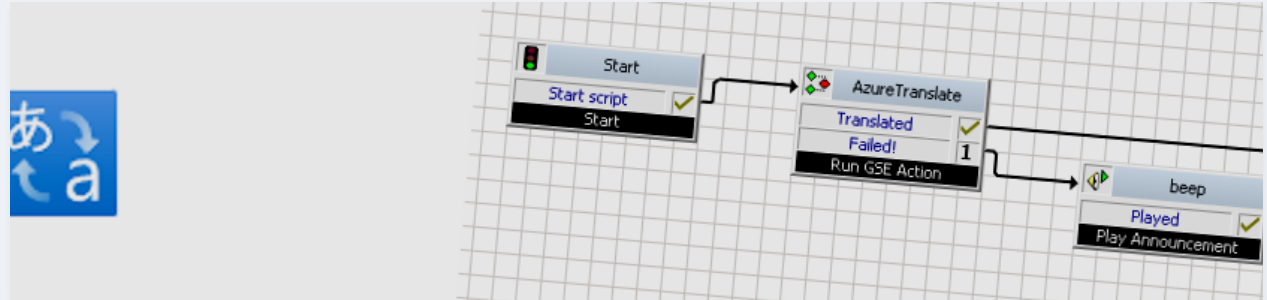
3.1 - Usage

3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Translate Text

A.2 - Example: Translate & Announce Text



AzureTranslate - 1 - Preparations

Followers

0

VBScript

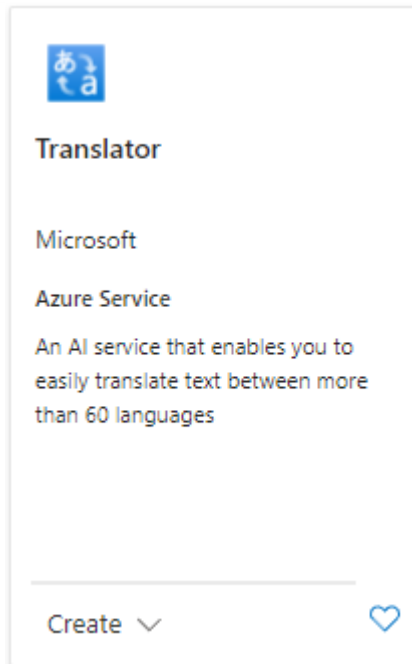
Lua

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

Additionally you need an account on the [Microsoft Azure Portal](#) and a **Translate 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.](#)

Although the above link explains how to create a subscription for the Speech service, the steps are more or less the same for the Translator service. In **Step 2** you just select the **Translator** service instead of the Speech service.



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 AzureTranslate extension was developed the "**Free F0**" tier included up to 2M characters translated per month. 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 AzureTranslate extension.



By Tom Wellige
June 23, 2023

 Share

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

AzureTranslate

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

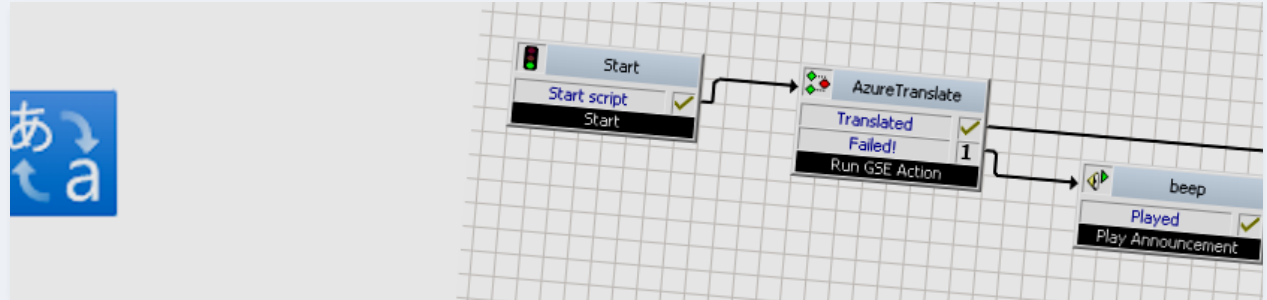
3.1 - Usage

3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Translate Text

A.2 - Example: Translate & Announce Text



AzureTranslate - 2 - Setup GSE Action

Followers

0

VBScript

Lua

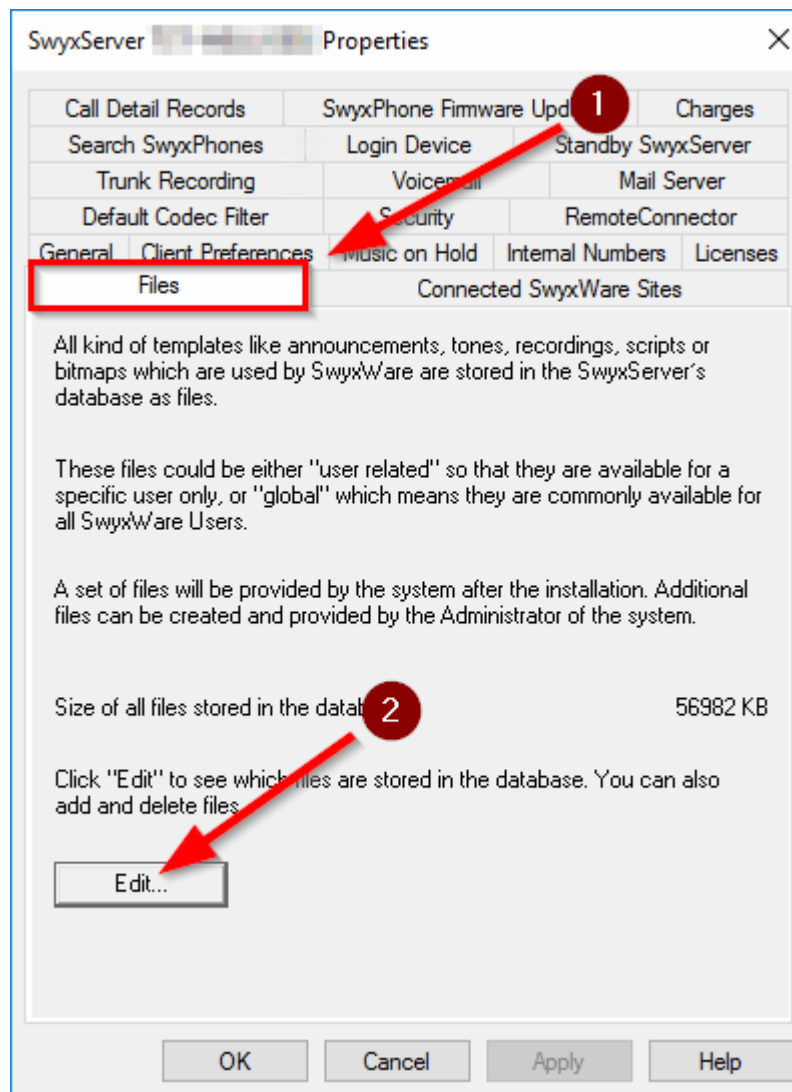
The **AzureTranslate** extension is designed as GSE action. To install the GSE action you need to use the **SwyxWare Administration**.

Step 1

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

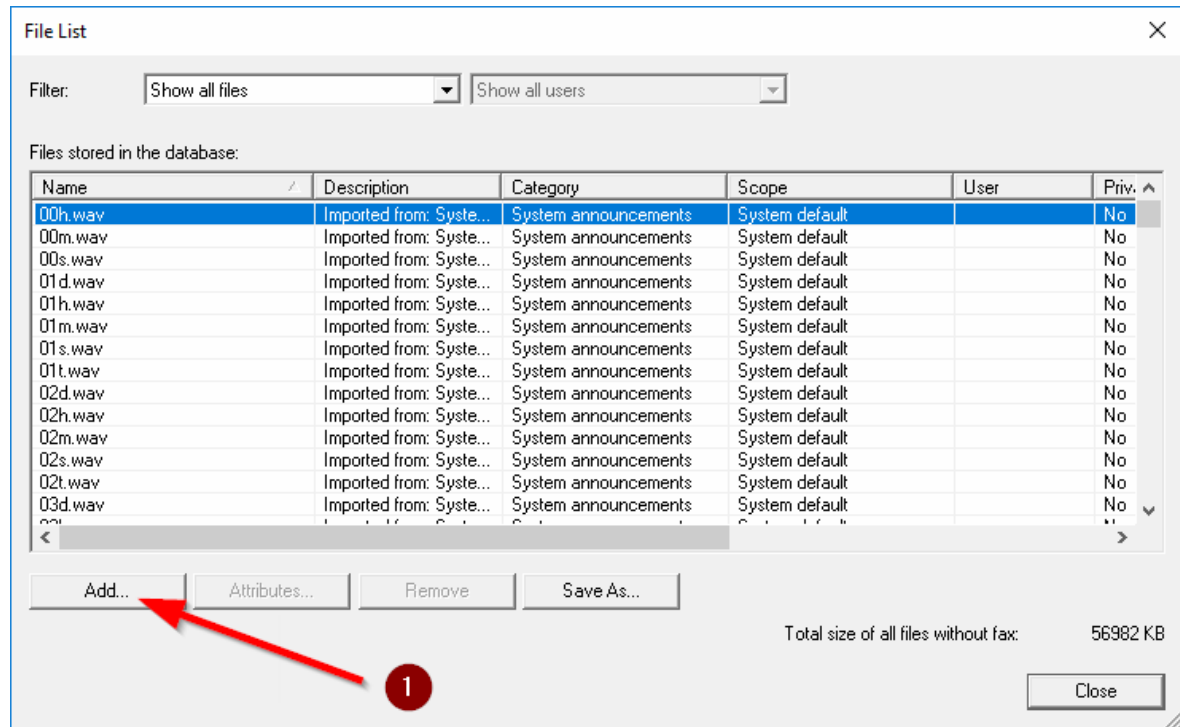
Step 2

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 **AzureTranslate** 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 circles and numbers (1, 2, 3, 4) highlighting specific areas:

- 1** points to the "File to add:" text box, which contains the path "C:\Projects\temp_azuretranslate\actionAzureTra" and a browse button "...".
- 2** points to the "Scope:" dropdown menu, which is set to "Global".
- 3** points to the "Category:" dropdown menu, which is set to "Call Routing VBS scripts".
- 4** points to the "Description:" text box, which contains the text "AzureTranslate".

Other fields in the dialog include:

- "Name:" with the text "actionAzureTranslate.ase; actionAzi".
- "User:" dropdown set to "Conference".
- "Group:" dropdown set to "Everyone".
- A "File Properties" section with three unchecked checkboxes: "Private", "Hidden", and "System".
- "OK" and "Cancel" buttons at the bottom right.

Step 4.2 - Lua usage

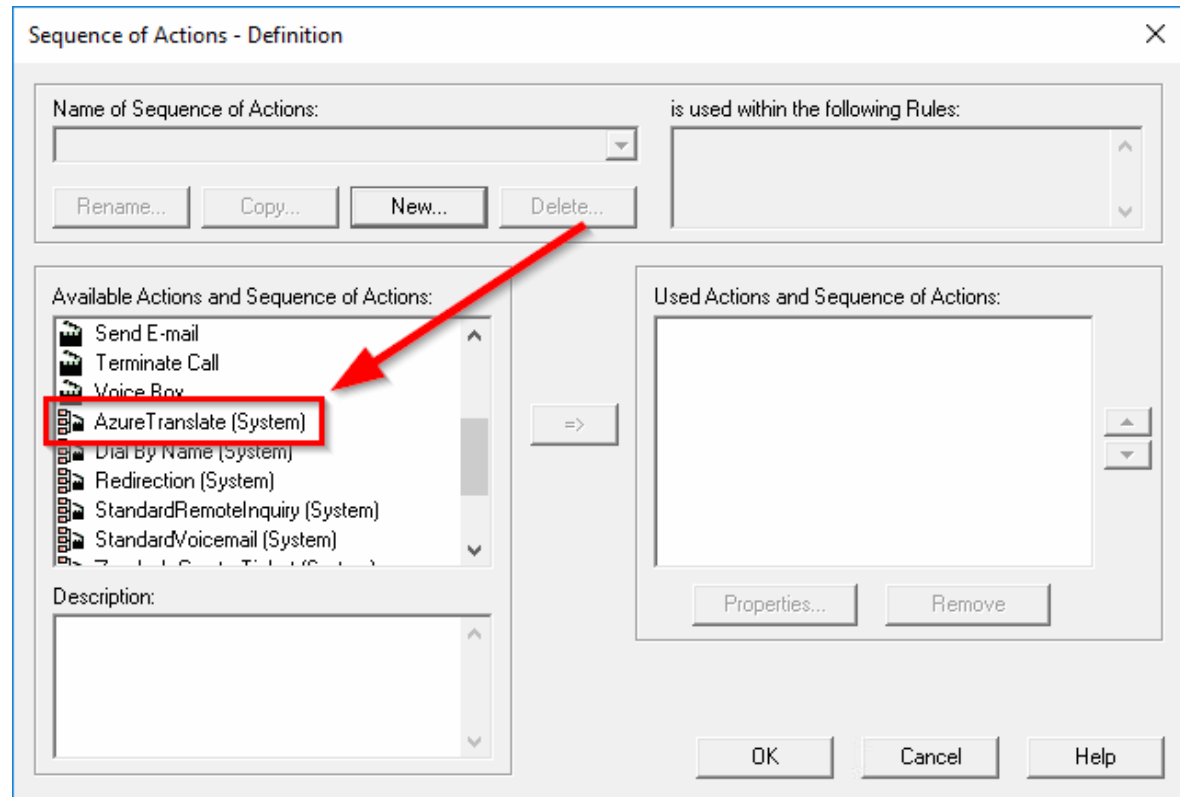
1. Select all files from the **Lua based\ase** folder from the download package.
2. Select **Global** as **Scope**. (you can also select User to load the files into the user scope a your script user)
3. Select **Call Routing Lua scripts** as **Category**.
4. It is recommended to enter **AzureTranslate**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

The screenshot shows a dialog box titled "Add File to Database" with a close button (X) in the top right corner. The dialog contains the following fields and options:

- File to add:** A text field containing the path "C:\Projects\AzureTranslate\Lua based\ase\action" followed by a red circle with the number "1" and a browse button "...".
- Name:** A text field containing "actionAzureTranslate.ase; actionAzi".
- Scope:** A dropdown menu with "Global" selected, preceded by a red circle with the number "2".
- Category:** A dropdown menu with "Call Routing Lua scripts" selected, preceded by a red circle with the number "3".
- User:** A dropdown menu with "Conference" selected.
- Group:** A dropdown menu with "Everyone" selected.
- File Properties:** A section with three unchecked checkboxes: "Private", "Hidden", and "System".
- Description:** A text field containing "AzureTranslate" preceded by a red circle with the number "4".
- 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 **AzureTranslate** action. The **(System)** behind the GSE action name shows that this is a global action being available for every SwyxWare user.



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



By Tom Wellige
June 23, 2023

 Share

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

AzureTranslate

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

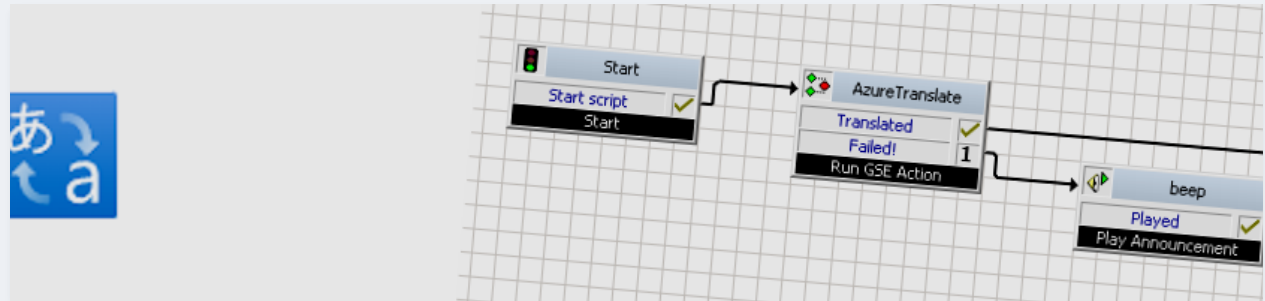
3.1 - Usage

3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Translate Text

A.2 - Example: Translate & Announce Text



AzureTranslate - 3.1 - Usage

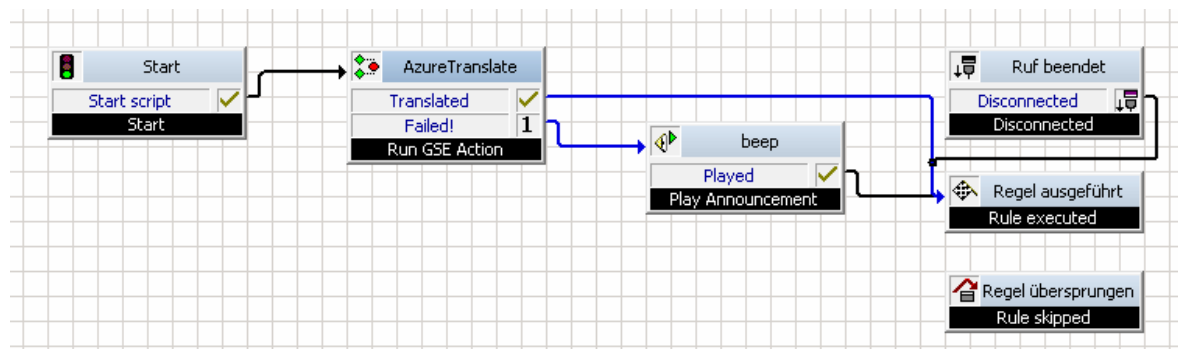
Followers

0

VBScript

Lua

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



Configure action parameters

Run GSE Action Properties

General Parameters Links

Select GSE action: Azure Translate (System)

Set action parameters:

Name	Value	Default Value
Region	westeurope	westeurope
SubscriptionKey	= ""	= ""
SourceText	Good morning my dear!	= ""
SourceLanguage	= ""	= ""
TargetLanguage	de	= ""

Edit Parameter...

Description:

Azure Translate - Translates a given text (v1.0.0)

Return Values:

0 - Translated
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.

SourceText

This is the text to get translated.

SourceLanguage

This is the language the "SourceText" is in. If you omit this parameter, the language will be detected automatically. A list of valid language values can be found at the [Translate REST API](#) page under [supported Languages](#). [Here](#) is the current complete list of languages.

TargetLanguage

This is the language the "SourceText" is in. A list of valid language values can be found at the [Translate REST API](#) page under [supported Languages](#). [Here](#) is the current complete list of languages.

Configure action exits

Run GSE Action Properties

General Parameters Links

Visible	Fixed name	Link name	Linked to
☒	Default	Translated	Run GSE Action
☒	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 "**Translated**".

Exit 1

This exit will be reached when there was any kind of problem and no text has been translated. 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 values (as global variables)

g_sAzureTranslate_LatestTranslation (string)

This global variable holds the translated text after the **Translated** (0) exit has been reached.

g_sAzureTranslate_LatestDetectedLanguage (string)

This global variable holds the detected language of the **SourceText** (in case the **SourceLanguage** parameter was omitted) after the **Translated** (0) exit has been reached.

g_sAzureTranslate_LatestDetectedLanguageConfidence (string)

This global variable holds a confidence value of the language detection of the **SourceText** (in case the **SourceLanguage** parameter was omitted) after the **Translated** (0) exit has been reached.



By Tom Wellige

June 23, 2023

 Share

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

AzureTranslate

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

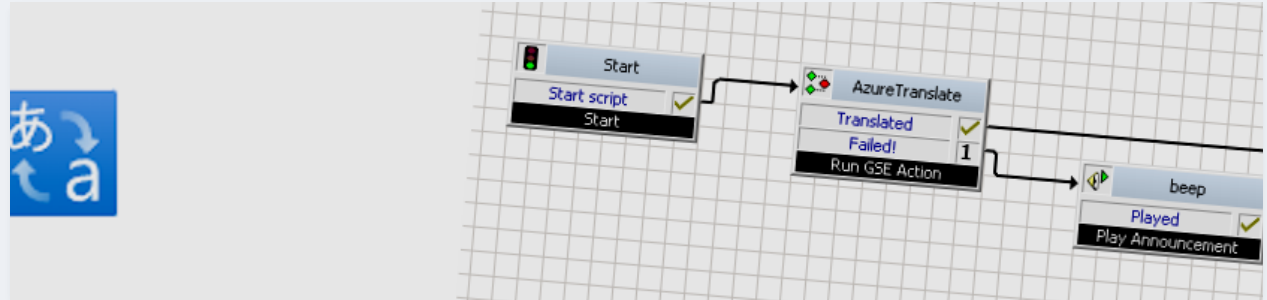
3.1 - Usage

3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Translate Text

A.2 - Example: Translate & Announce Text



AzureTranslate - 3.2 - Trouble Shooting

Followers

0

VBScript

Lua

In case the translation 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:

```
-----> AzureTranslate.Class_Initialize
```

For **Lua** based call routing:

```
-----> AzureTranslate.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 and calling the Azure REST API to translate the given text.

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](#).



By Tom Wellige

June 23, 2023

Share

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

AzureTranslate

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

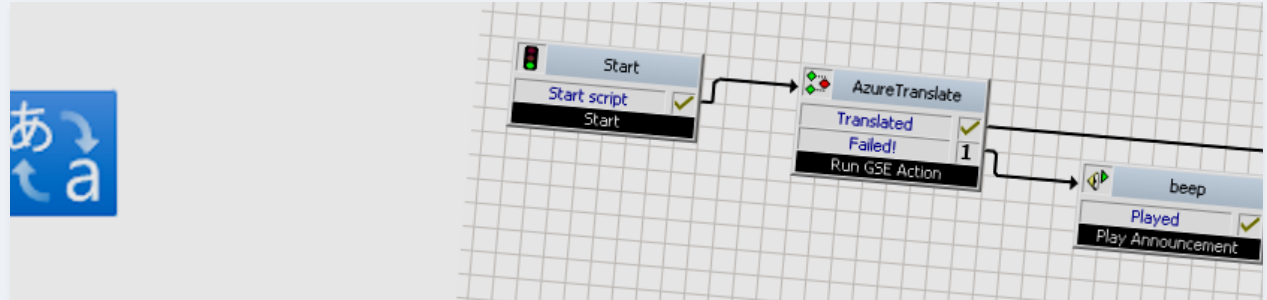
3.1 - Usage

3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Translate Text

A.2 - Example: Translate & Announce Text



AzureTranslate - A.1 - Example: Translate Text

Followers

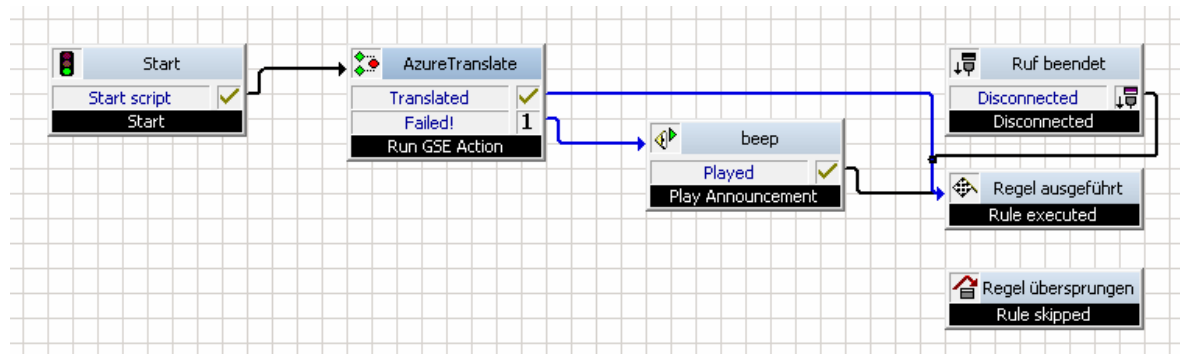
0

VBScript

Lua

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

It translates any given text written in any language into any other language. In case of an error (in the usage or execution of the AzureTranslate 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\Translate Text.rse
```

For **Lua** usage:

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

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



By Tom Wellige

June 23, 2023

 Share

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

AzureTranslate

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

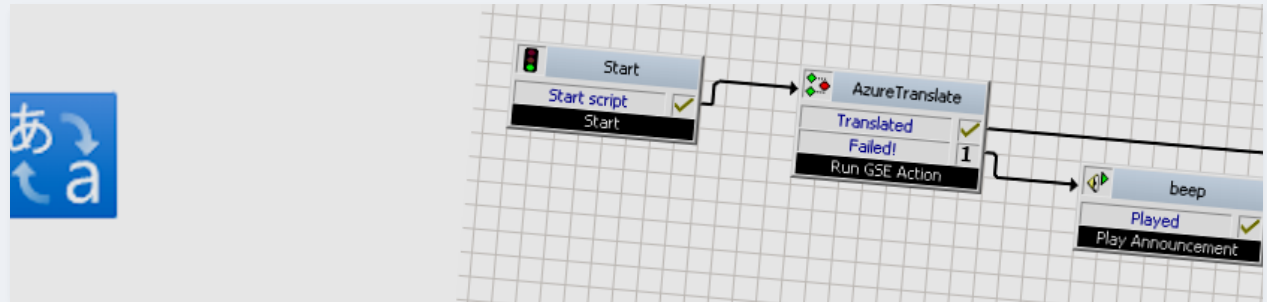
3.1 - Usage

3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Translate Text

A.2 - Example: Translate & Announce Text



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

Followers

0

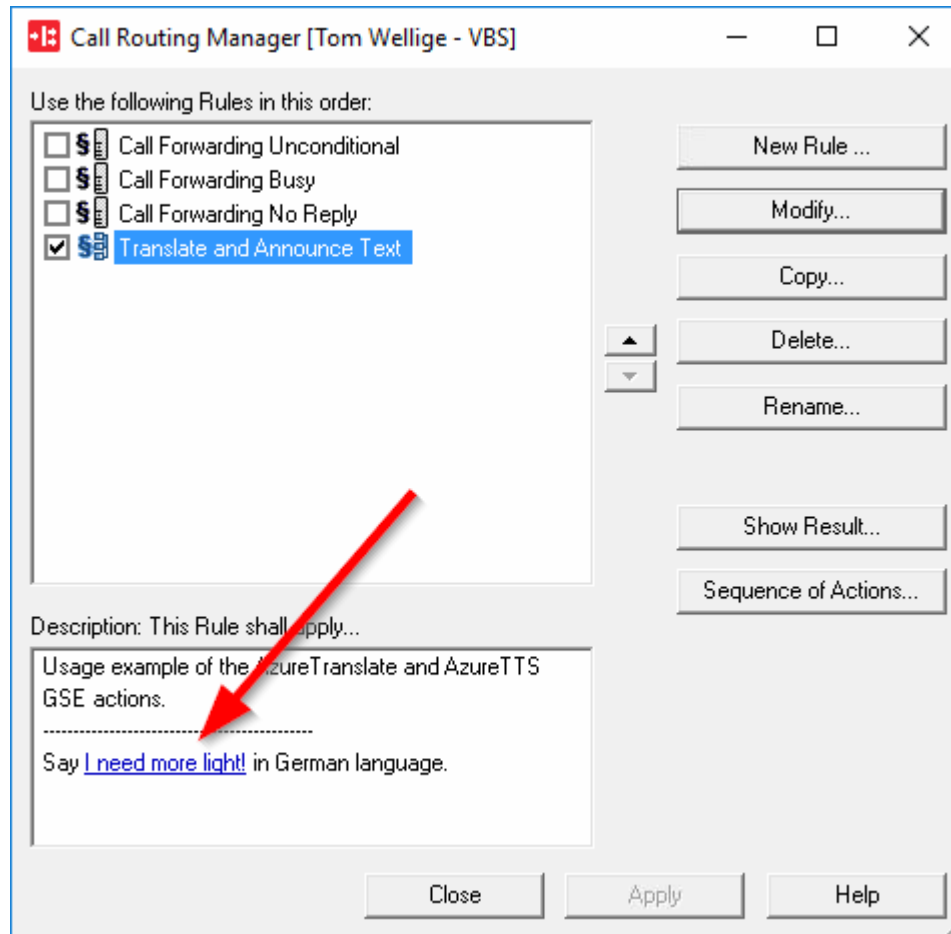
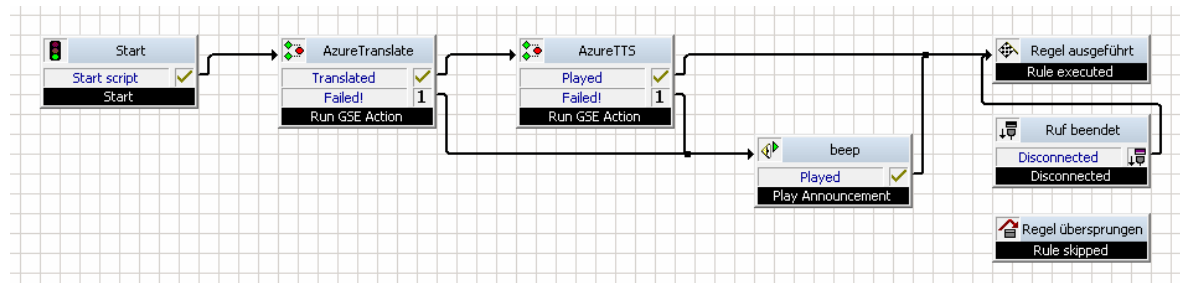
VBScript

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 [AzureTTS - 2 - Setup GSE Action](#) to also install the **AzureTTS** 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

Theme ▼

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

AzureTranslate

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2 - Setup GSE Action

USAGE

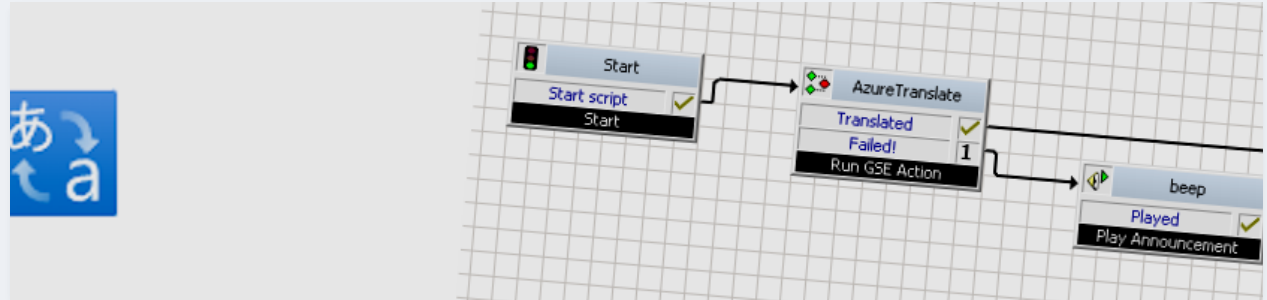
3.1 - Usage

3.2 - Trouble Shooting

APPENDIX A

A.1 - Example: Translate Text

A.2 - Example: Translate & Announce Text



AzureTranslate - 0 - Introduction

Followers

0

VBScript

Lua

This extension provides **text translation** 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 12.40** (or newer) for **VBScript based** call routing
- requires **SwyxWare 13.10** (or newer) for **Lua based** call routing

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



Please refer to the [Forums](#) to discuss the AzureTranslate 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 AzureTranslate 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

AzureTranslate
v1.0.0

This is a Swyx Forum Open Source Project.

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

<https://www.swyxforum.com/azuretranslate/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.

For the VBScript based integration this project also includes the following Open Source project:

JSON object class 3.5.4 - May, 29th - 2016

Licence:

The MIT License (MIT)

Copyright (c) 2016 RCDMK - rcdmk[at]hotmail[dot]com

<https://github.com/rcdmk/aspJSON>

<https://github.com/rcdmk/aspJSON/blob/master/README.md>

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 shall 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.

Modifications needed for SwyxWare Call Routing
Tom Wellige, 03.05.2017

For the Lua based integration this project also includes the following Open Source project:

Simple JSON encoding and decoding in pure Lua.
Copyright 2010-2016 Jeffrey Friedl

<http://regex.info/blog/>

Latest version: <http://regex.info/blog/lua/json>

This code is released under a Creative Commons CC-BY "Attribution" License:
http://creativecommons.org/licenses/by/3.0/deed.en_US

It can be used for any purpose so long as:

- 1) the copyright notice above is maintained
- 2) the web-page links above are maintained
- 3) the 'AUTHOR_NOTE' string below is maintained



By Tom Wellige
June 23, 2023

 Share

Theme ▼

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