

0 - Introduction	2
1 - Preparations	7
2.1 - Setup GSE Actions	9
2.2 - Setup Zendesk SwyxIt! button	16
3.1 - Usage - Create Ticket	21
3.2 - Usage - Open Ticket	26
3.3 - Usage - Ticket Status	32
3.4 - Usage - Update Ticket	42
3.5 - Usage - Zendesk SwyxIt! button	47
3.6 - Usage - Trouble Shooting	51
A.1 - Example - Check Status	54
A.2 - Example - Check Status and Update Ticket	57
A.3 - Example - Create Ticket	61
A.4 - Example - Update Ticket	64
A.5 - Example - Open Ticket	67
A.6 - Example - Check and Announce Status	70
B.1 - [Obsolete] Ticket Validation against Zendesk	75
B.2 - Access modify the code behind	77
B.3 - Version History	86

## Zendesk Integration

### INTRODUCTION

#### 0 - Introduction

### PREPARATIONS

#### 1 - Preparations

### SETUP

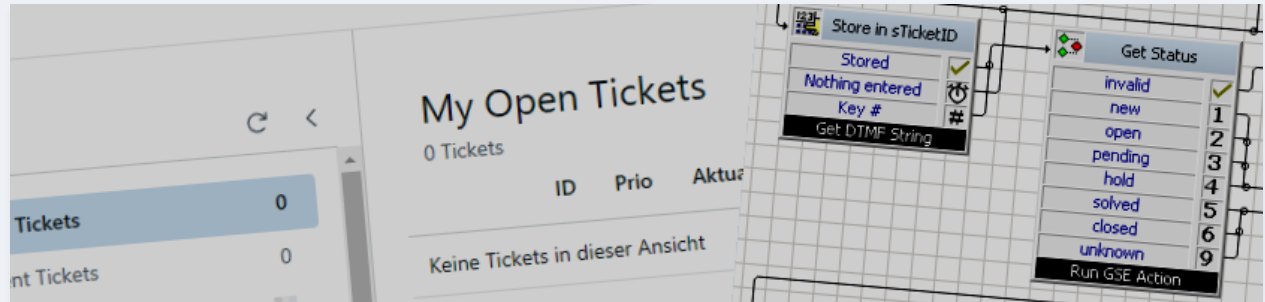
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket
- 3.3 - GSE Action Zendesk Ticket Status
- 3.4 - GSE Action Zendesk Update Ticket
- 3.5 - Zendesk SwyxIt! button
- 3.6 - Trouble Shooting

### APPENDIX A

- A.1 - Example: Check Status



## Zendesk Integration - 0 - Introduction

Followers

0

VBScript

Lua

This extension provides Zendesk ticket functionality for the SwyxWare call routing:

- **Check** the current **status** of a Zendesk ticket
- **Create** a new Zendesk ticket (for example to register a call back request)
- **Update** an existing Zendesk ticket
- **Open** an existing Zendesk ticket in an agent's web browser window
- **Zendesk SwyxIt! button** to open the Zendesk ticket directly with a mouse click within SwyxIt!
- 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

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History

To handle Zendesk tickets it makes use of the Zendesk Core API (REST API) and the Zendesk Talk Partner Edition API (REST API).



Please refer to the Forums to discuss the Zendesk Integration 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 from **13.10** on. As of the release of the Zendesk Integration 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

Zendesk SwyxWare Integration  
v1.5.0

This is a Swyx Forum Open Source Project.

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

<https://www.swyxforum.com/zendesk-integration/introduction/0-introduction-r6/>

The MIT License (MIT)

Copyright (c) 2017-22 by Swyx Forum

Copyright (c) 2017-22 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](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  
May 14, 2022

 Share

Theme ▼

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

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

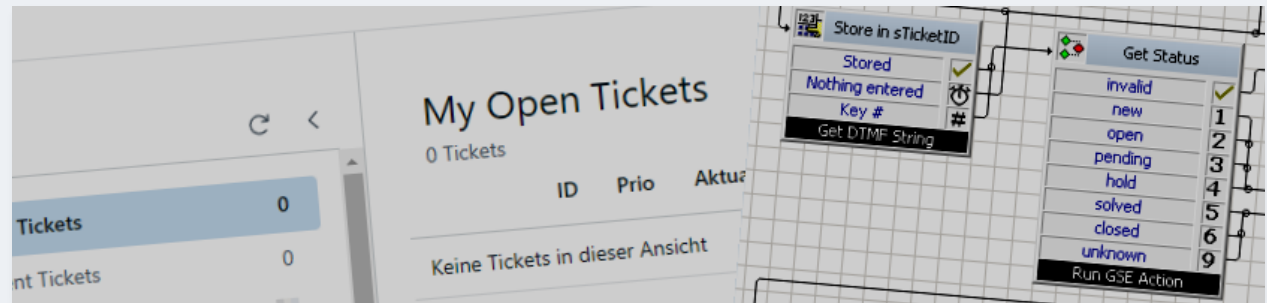
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket
- 3.3 - GSE Action Zendesk Ticket Status
- 3.4 - GSE Action Zendesk Update Ticket
- 3.5 - Zendesk SwyxIt! button
- 3.6 - Trouble Shooting

### APPENDIX A

- A.1 - Example: Check Status



## Zendesk Integration - 1 - Preparations

Followers

0

VBScript

Lua

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

Please note that from version **1.4.0** on, this extension requires **SwyxWare 13.10 (or higher)** to be installed.

Please note that from version **1.3.0** on, this extension requires **SwyxWare 12.40 (or higher)** to be installed.

Older versions of this extension made use of the open source "[curl](#)" tool for the required web requests into the [Zendesk Core API](#) (REST). From version 1.3.0 on this is not needed anymore, as it uses the new [PBXScript.WebRequest](#) api from SwyxWare 12.40 (or higher).

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

A.6 - Example: Check and Announce Status

## **APPENDIX B**

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History



By Tom Wellige

May 14, 2022

 Share

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

2.1 - Setup Zendesk GSE Actions

2.2 - Setup Zendesk SwyxIt! button

### USAGE

3.1 - GSE Action Zendesk Create Ticket

3.2 - GSE Action Zendesk Open Ticket

3.3 - GSE Action Zendesk Ticket Status

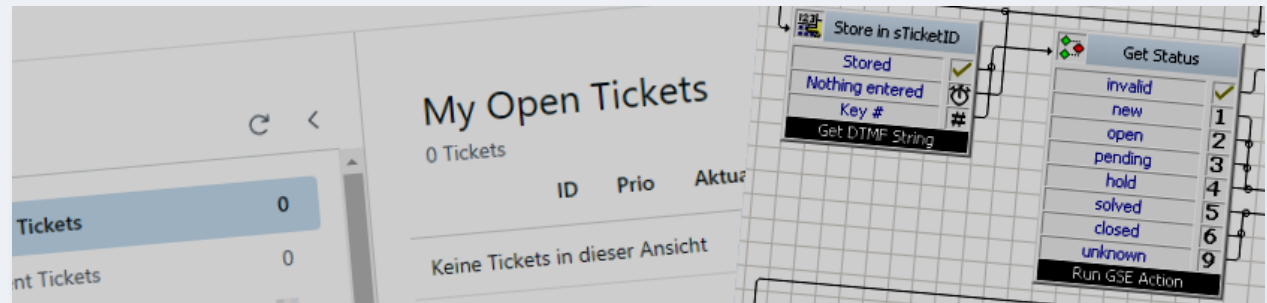
3.4 - GSE Action Zendesk Update Ticket

3.5 - Zendesk SwyxIt! button

3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - 2.1 - Setup Zendesk GSE Actions

Followers 0

VBScript

Lua

The **Zendesk Integration** extension is designed as GSE actions. To install these GSE actions 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....**

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

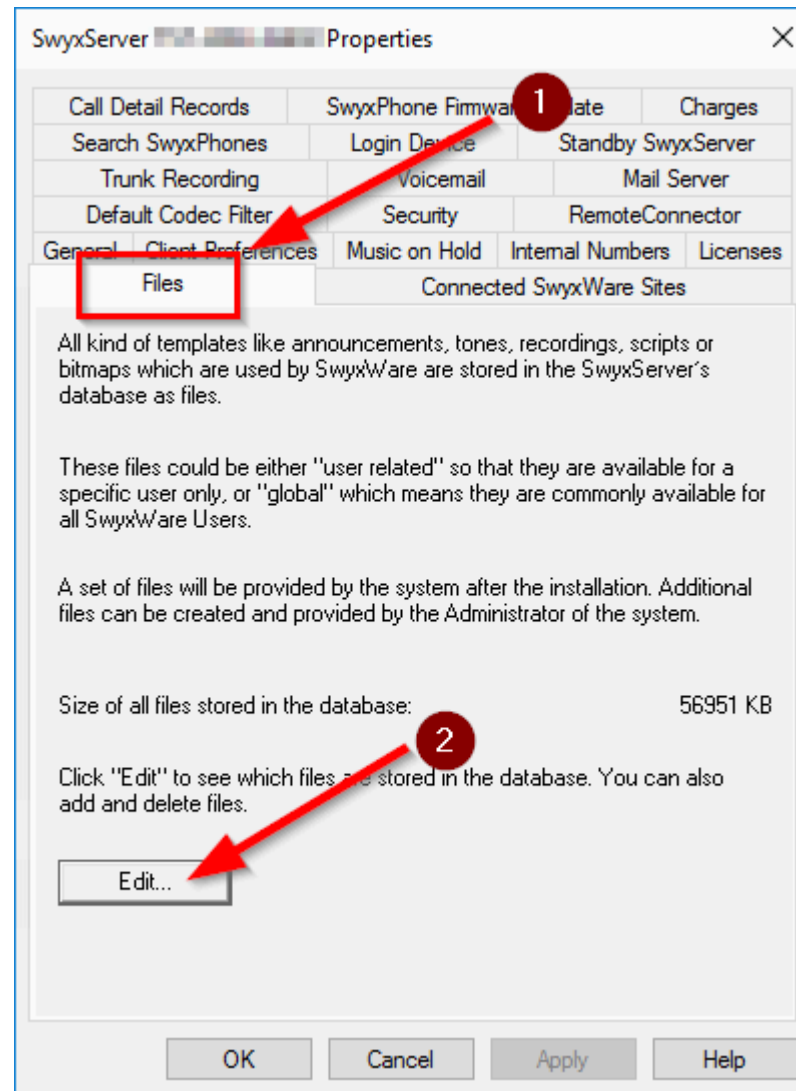
A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

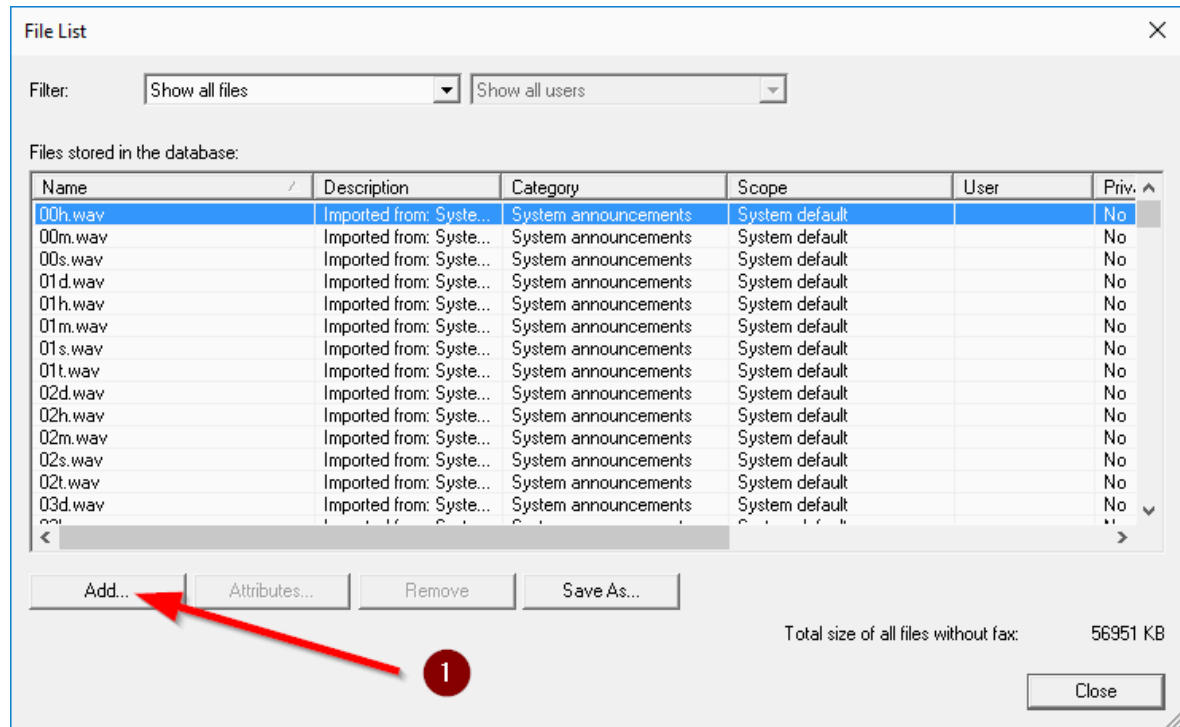
B.2 - Access/modify the code behind

B.3 - Version History



### 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 **Zendesk Integration** 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 the following fields and controls:

- File to add:** A text field containing the path "C:\Setup\Swyx\ZendeskIntegration\_v1.4.0\VBSc" followed by a red circle with the number "1" and a browse button "...".
- Name:** A text field containing "actionZendeskCreateTicket.ase; act".
- Scope:** A dropdown menu with "Global" selected, preceded by a red circle with the number "2".
- Category:** A dropdown menu with "Call Routing VBS scripts" selected, preceded by a red circle with the number "3".
- User:** A dropdown menu with "Conference" selected.
- File Properties:** A section with three unchecked checkboxes: "Private", "Hidden", and "System".
- Description:** A text field containing "Zendesk Integration" preceded by a red circle with the number "4".
- 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 **Zendesk Integration** into the **Description** field, but not necessary.

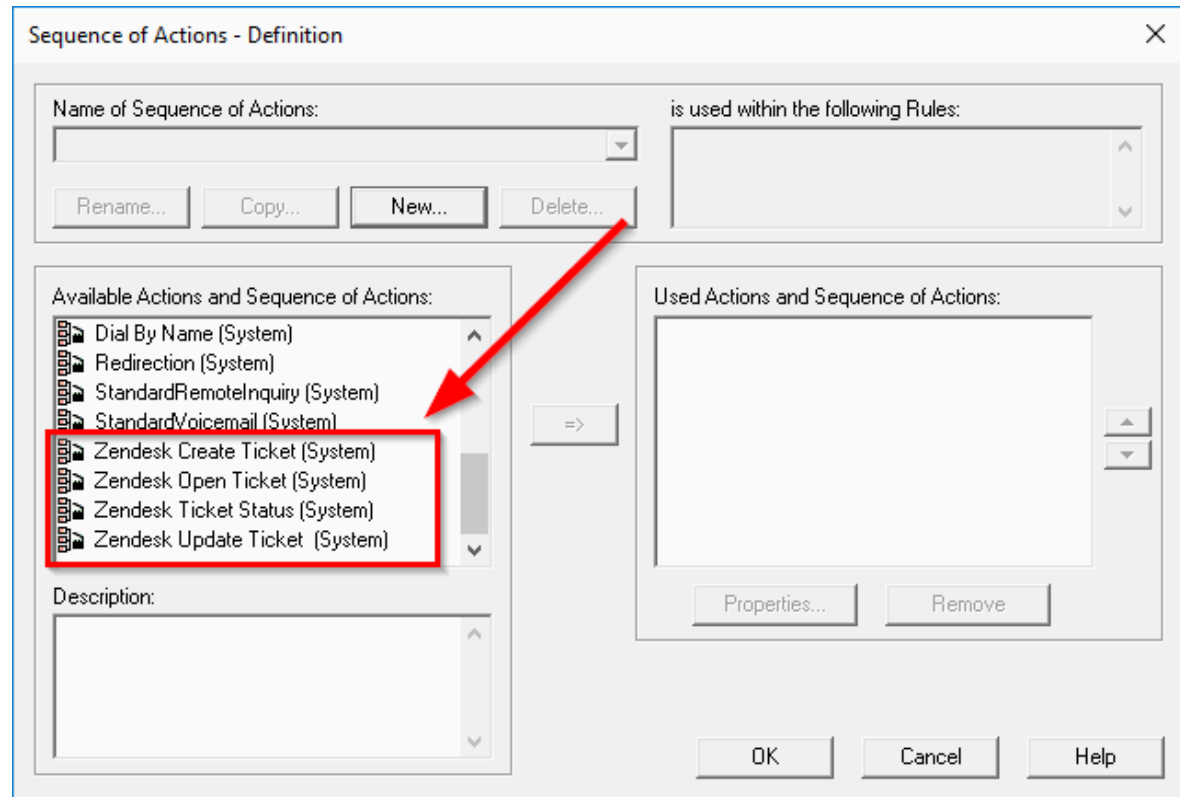


The screenshot shows a Windows-style dialog box titled "Add File to Database" with a close button (X) in the top right corner. The dialog contains the following fields and options:

- File to add:** A text field containing the path "C:\Setup\Swyx\ZendeskIntegration\_v1.4.0\Lua" followed by a browse button "...". A red circle with the number "1" is placed over the browse button.
- Name:** A text field containing "actionZendeskCreateTicket.ase; act".
- Scope:** A dropdown menu with "Global" selected. A red circle with the number "2" is placed over the dropdown arrow.
- Category:** A dropdown menu with "Call Routing Lua scripts" selected. A red circle with the number "3" is placed over the dropdown arrow.
- User:** A dropdown menu with "Conference" selected.
- File Properties:** A section with three unchecked checkboxes: "Private", "Hidden", and "System".
- Description:** A text field containing "Zendesk Integration". A red circle with the number "4" is placed over the text field.
- At the bottom are "OK" and "Cancel" buttons.

### 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 **Zendesk actions**. The **(System)** behind the GSE action name shows that this is a global action being available for every SwyxWare user.



The Zendesk Integration actions are now installed and can be fully used within GSE call routing scripts.



By Tom Wellige  
May 14, 2022

 Share



Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

2.1 - Setup Zendesk GSE Actions

2.2 - Setup Zendesk SwyxIt! button

### USAGE

3.1 - GSE Action Zendesk Create Ticket

3.2 - GSE Action Zendesk Open Ticket

3.3 - GSE Action Zendesk Ticket Status

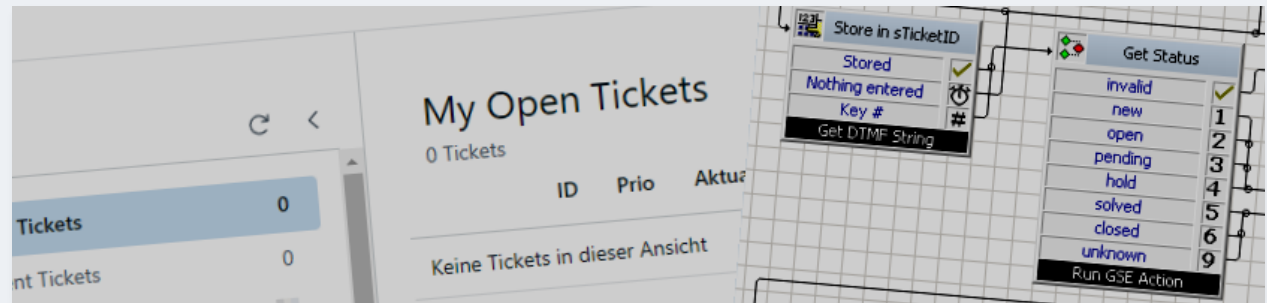
3.4 - GSE Action Zendesk Update Ticket

3.5 - Zendesk SwyxIt! button

3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - 2.2 - Setup Zendesk SwyxIt! button

Followers

0

VBScript

Lua

Setting up the Zendesk SwyxIt! button needs to be done in two steps:

### Step 1

Copy the file

```
\Skin Integration\OpenZendesk.wsf
```

into the SwyxIt! program folder, e.g.

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History

```
C:\Program Files (x86)\SwyxIt!\
```

Open this file in a standard text editor (e.g. notepad.exe) and modify the following line:

```
' You need to modify this!  
Const ZENDESK_DOMAIN = "dummy.zendesk.com"
```

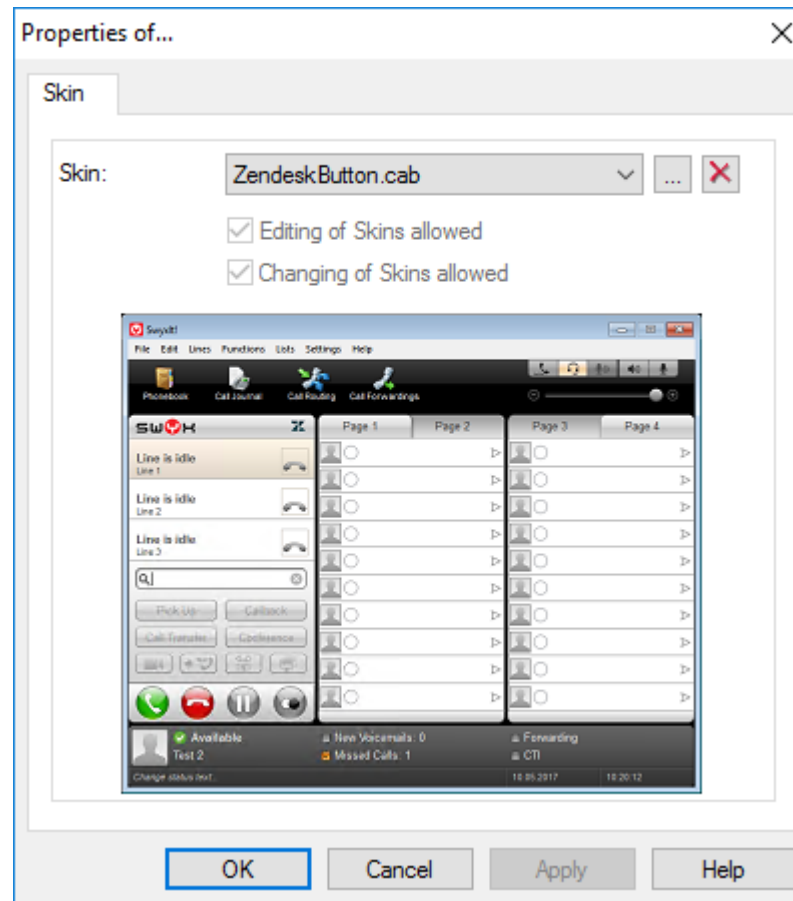
This should point to your Zendesk domain URL. This is something like "yourcompanyname.zendesk.com".

### Step 2

Install the SwyxIt! skin containing the Zendesk button. To do so

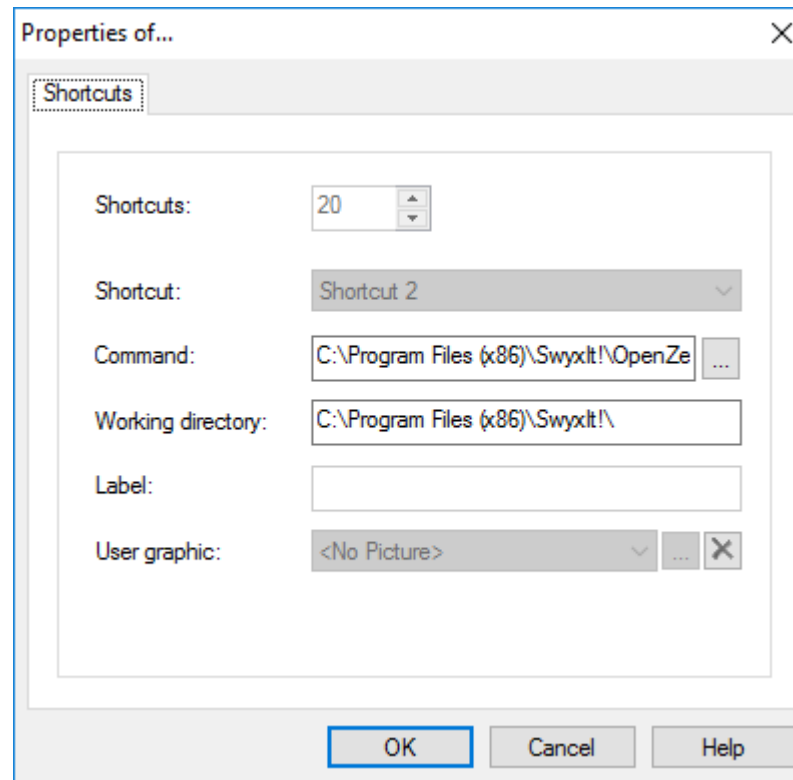
1. Open **SwyxIt!**
2. Navigate to the **File | Skin | Load...** menu item
3. Press the "... " button next to the Skin select box and select the following file from the download package

```
\Skin Integration\ZendeskButton.cab
```



4. Press **Ok**
5. **Right click** the **Zendesk button** right next to the Swyx logo above the line keys.
6. Enter the complete name and path of the file you have copied in step 1 into the **Command** field:

```
C:\Program Files (x86)\SwyxIt!\OpenZendesk.wsf
```



7. Press **Ok**.

This concludes the installation of the Zendesk Swyxit! button.

The usage is explained in chapter [3.5 - Zendesk Swyxit! button](#).

A complete example making use of this button is shown in chapter [A.2 - Example: Check Status and Update Ticket](#).



By Tom Wellige  
May 14, 2022

 Share



Previous Page

2.1 - Setup Zendesk GSE Actions

Theme ▼

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



## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

3.1 - GSE Action Zendesk Create Ticket

3.2 - GSE Action Zendesk Open Ticket

3.3 - GSE Action Zendesk Ticket Status

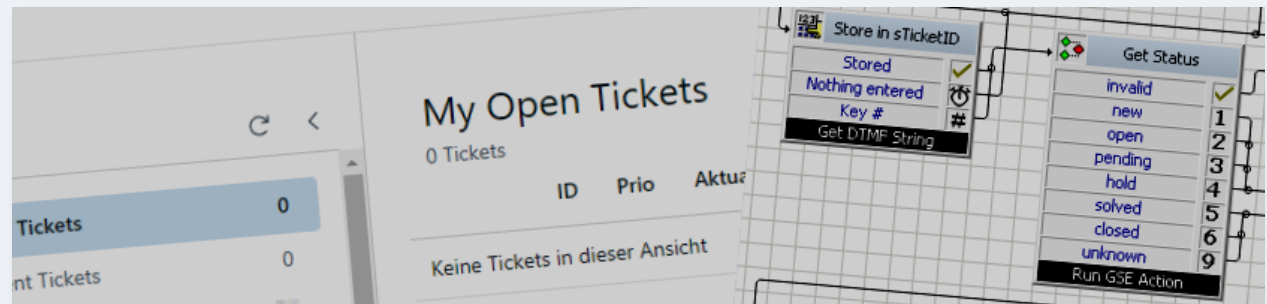
3.4 - GSE Action Zendesk Update Ticket

3.5 - Zendesk SwyxIt! button

3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - 3.1 - GSE Action Zendesk Create Ticket

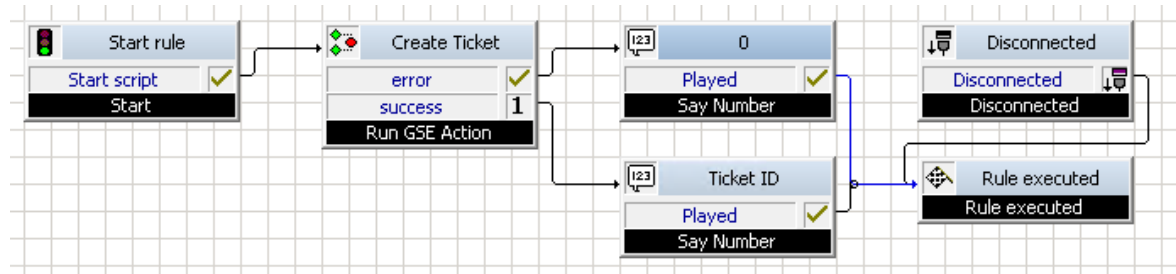
Followers 0

VBScript

Lua

This GSE action can be used to create a new Zendesk ticket from within the call routing.

An example call routing script can be found in [A.3 - Example: Create Ticket](#).

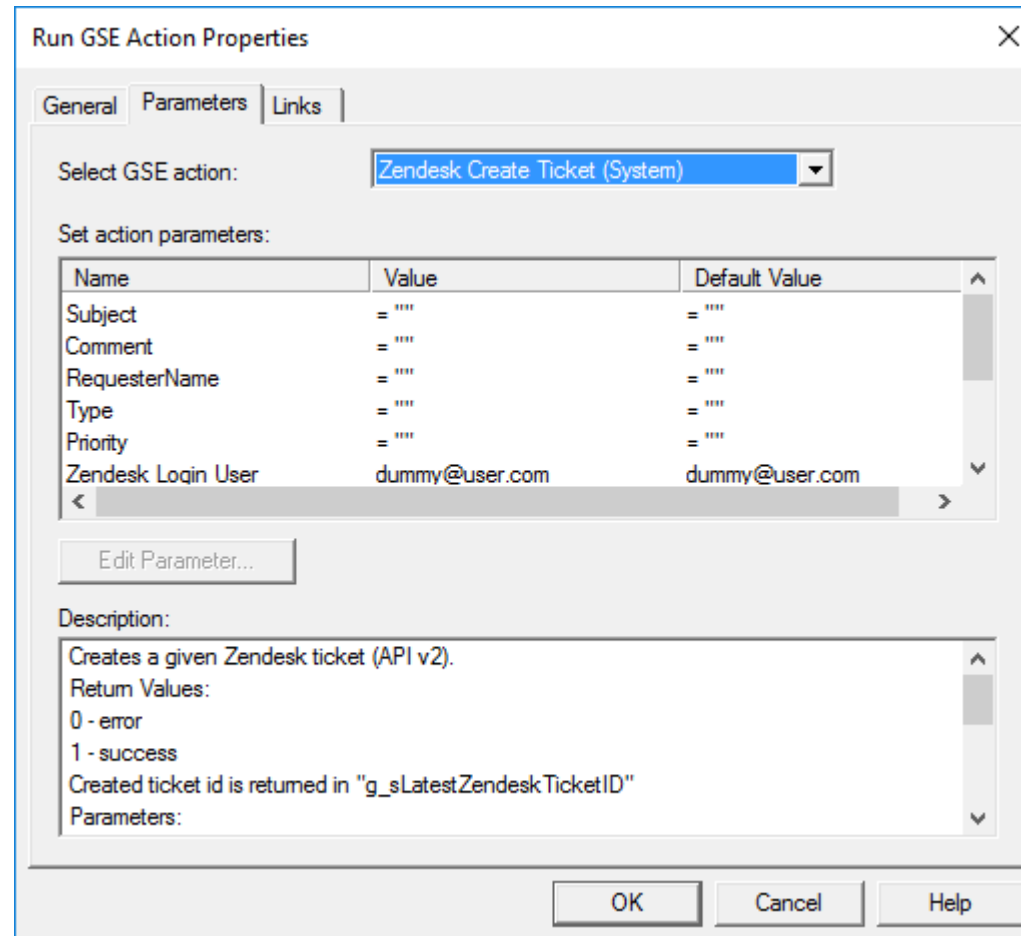


- A.2 - Example: Check Status and Update Ticket
- A.3 - Example: Create Ticket
- A.4 - Example: Update Ticket
- A.5 - Example: Open Ticket
- A.6 - Example: Check and Announce Status

## APPENDIX B

- B.1 - [Obsolete] Ticket Validation against Zendesk
- B.2 - Access/modify the code behind
- B.3 - Version History

### Configure action parameters



The dialog box titled "Run GSE Action Properties" has three tabs: "General", "Parameters", and "Links". The "Parameters" tab is active. It features a dropdown menu for "Select GSE action:" with "Zendesk Create Ticket (System)" selected. Below this is a table for "Set action parameters:" with columns "Name", "Value", and "Default Value". The table lists parameters: Subject, Comment, RequesterName, Type, Priority, and Zendesk Login User. The "Zendesk Login User" row is highlighted. Below the table is an "Edit Parameter..." button. At the bottom of the dialog is a "Description:" text area containing the text: "Creates a given Zendesk ticket (API v2). Return Values: 0 - error, 1 - success. Created ticket id is returned in 'g\_sLatestZendeskTicketID' Parameters:". The dialog has "OK", "Cancel", and "Help" buttons at the bottom right.

Name	Value	Default Value
Subject	= ""	= ""
Comment	= ""	= ""
RequesterName	= ""	= ""
Type	= ""	= ""
Priority	= ""	= ""
Zendesk Login User	dummy@user.com	dummy@user.com

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

#### Subject

The subject of the ticket.

#### Comment

The initial comment string to be placed in the new ticket.

**RequestName**

Name of the requester. If left empty either the **caller name** (if available) or the **caller number** will be used automatically.

**Type**

As of [Zendesk Core API](#) allowed values are **problem**, **incident**, **question**, or **task**. Any other type value will be ignored by the GSE action.

**Priority**

As of [Zendesk Core API](#) allowed values are **urgent**, **high**, **normal**, or **low**. Any other priority value will be ignored by the GSE action.

**Zendesk Login User**

Login user. To connect to Zendesk a Zendesk user login is required. This project uses a username/token authentication instead of username/password. This is a more robust solution, and keeps the call routing script independent from any password changes the Zendesk user might do. Please follow [this link](#) to learn how to obtain the needed login token.

**Zendesk Login Token**

Login token. To connect to Zendesk a Zendesk user login is required. This project uses a username/token authentication instead of username/password. This is a more robust solution, and keeps the call routing script independent from any password changes the Zendesk user might do. Please follow [this link](#) to learn how to obtain the needed login token.

**Zendesk Domain**

Your Zendesk domain URL. This is something like "yourcompanyname.zendesk.com".

**Configure action exits**

Run GSE Action Properties

General Parameters Links

Visible	Fixed name	Link name	Linked to
<input checked="" type="checkbox"/>	Default	error	0
<input checked="" type="checkbox"/>	Return code 1	success	Ticket ID
<input type="checkbox"/>	Return code 2		[no link]
<input type="checkbox"/>	Return code 3		[no link]
<input type="checkbox"/>	Return code 4		[no link]
<input type="checkbox"/>	Return code 5		[no link]
<input type="checkbox"/>	Return code 6		[no link]
<input type="checkbox"/>	Return code 7		[no link]
<input type="checkbox"/>	Return code 8		[no link]
<input type="checkbox"/>	Return code 9		[no link]
<input type="checkbox"/>	Disconnected		Disconnected

OK Cancel Help

#### Exit 0 (Default)

This exit will be reached on any **error** during creating the ticket. Please refer to [3.6 - Trouble shooting](#) for more information on how to do a deeper analysis of the error. You should label this exit to something like **error**.

#### Exit 1

This exit will be reached if the ticket has been **successfully** created. You need to enable this exit (checkbox) and should label it to something like "**success**".

### Additional return value (as global variable)

**g\_sLatestZendeskTicketID** (string)

This global variable holds the ID of the newly created ticket after the **ok** (0) exit has been reached.



By Tom Wellige  
May 14, 2022

 Share

Next Page

3.2 - GSE Action Zendesk Open Ticket



Theme ▼

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

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

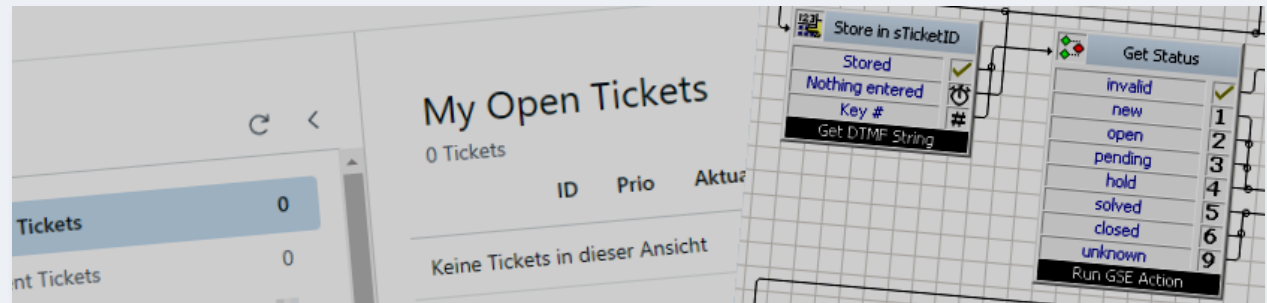
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket**
- 3.3 - GSE Action Zendesk Ticket Status
- 3.4 - GSE Action Zendesk Update Ticket
- 3.5 - Zendesk SwyxIt! button
- 3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - 3.2 - GSE Action Zendesk Open Ticket

Followers

0

VBScript

Lua

1.4.0

This GSE action can be used to open a given Zendesk ticket in an agent's web browser.

An example call routing script can be found in [A.5 - Example: Open Ticket](#).

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

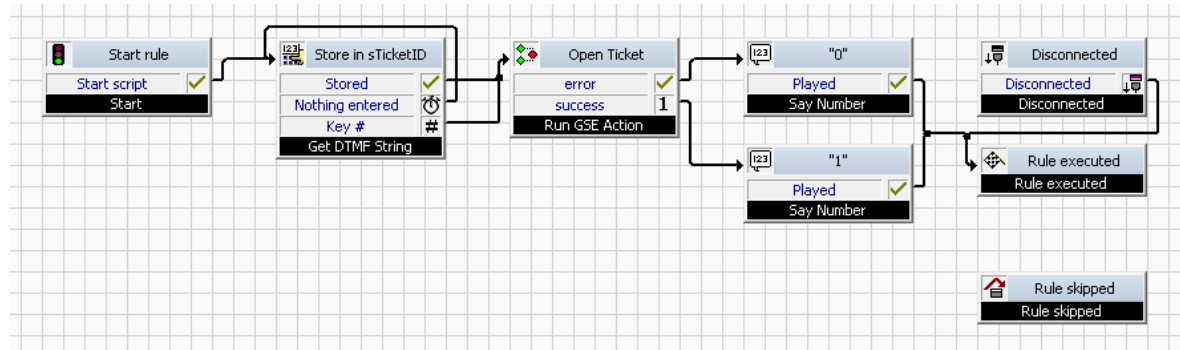
A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History



In order to have this working, the agent needs to have already at least one web browser window opened in which he is logged in into his Zendesk account.

Please note: due to a limitation of the Zendesk Talk Partner Edition API, all web browser windows the agent has currently opened in which he is logged into Zendesk will change to the given ticket.

### Configure action parameters

Run GSE Action Properties

General Parameters Links

Select GSE action: Zendesk Open Ticket (System)

Set action parameters:

Name	Value	Default Value
TicketID	0	0
Zendesk AgentID	0	0
Zendesk Login User	dummy@user.com	dummy@user.com
Zendesk Login Token	dummy token	dummy token
Zendesk Domain	dummy.zendesk.com	dummy.zendesk.com

Edit Parameter...

Description:  
Opens a given Zendesk ticket in agent's browser (v1.0.4)

Return Values:  
0 - error  
1 - success

OK Cancel Help

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

#### **TicketID**

The Zendesk ID of the ticket to update. The ID can be given with or without the Zendesk typical # prefix, i.e. **1234** or **#1234**.

#### **Zendesk AgentD**

The Zendesk user id of the agent in which browser the given ticket should be opened.

#### **Zendesk Login User**



Login user. To connect to Zendesk a Zendesk user login is required. This project uses a username/token authentication instead of username/password. This is a more robust solution, and keeps the call routing script independent from any password changes the Zendesk user might do. Please follow [this link](#) to learn how to obtain the needed login token.

### **Zendesk Login Token**

Login token. To connect to Zendesk a Zendesk user login is required. This project uses a username/token authentication instead of username/password. This is a more robust solution, and keeps the call routing script independent from any password changes the Zendesk user might do. Please follow [this link](#) to learn how to obtain the needed login token.

### **Zendesk Domain**

Your Zendesk domain URL. This is something like yourcompanyname.zendesk.com

### **Configure action exits**

Run GSE Action Properties

General Parameters Links

Visible	Fixed name	Link name	Linked to
<input checked="" type="checkbox"/>	Default	error	[123] "0"
<input checked="" type="checkbox"/>	Return code 1	success	[123] "1"
<input type="checkbox"/>	Return code 2		↓ [no link]
<input type="checkbox"/>	Return code 3		↓ [no link]
<input type="checkbox"/>	Return code 4		↓ [no link]
<input type="checkbox"/>	Return code 5		↓ [no link]
<input type="checkbox"/>	Return code 6		↓ [no link]
<input type="checkbox"/>	Return code 7		↓ [no link]
<input type="checkbox"/>	Return code 8		↓ [no link]
<input type="checkbox"/>	Return code 9		↓ [no link]
<input type="checkbox"/>	Disconnected		↓ Disconnected

OK Cancel Help

#### Exit 0 (Default)

This exit will be reached on any **error** during updating the ticket. Please refer to [3.6 - Trouble shooting](#) for more information on how to do a deeper analysis of the error. You should label this exit to something like "**error**".

#### Exit 1

This exit will be reached if the ticket has been **successfully** updated. You need to enable this exit (checkbox) and should label it to something like "**success**".

### Additional return value (as global variable)

**g\_latestZendeskTicketID** (string) 1.5.0

This global variable holds the ID of the latest opened ticket after the **ok** (0) exit has been reached.



By Tom Wellige

May 28, 2022

 Share

< Previous Page  
3.1 - GSE Action Zendesk Create Ticket

Next Page >  
3.3 - GSE Action Zendesk Ticket Status

Theme ▼

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

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

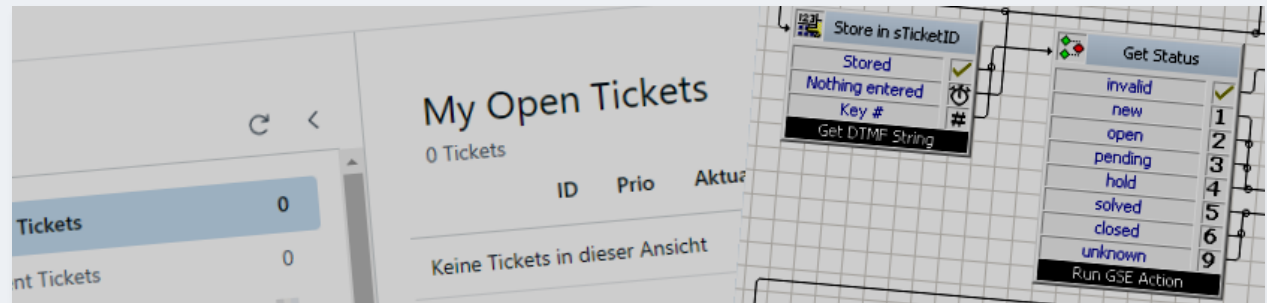
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket
- 3.3 - GSE Action Zendesk Ticket Status**
- 3.4 - GSE Action Zendesk Update Ticket
- 3.5 - Zendesk SwyxIt! button
- 3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - 3.3 - GSE Action Zendesk Ticket Status

Followers

0

VBScript

Lua

This GSE action can be used to check the status of a given Zendesk ticket from within the call routing.

An example call routing script can be found in [A.1 - Example: Check Status](#) or a more advanced example in [A.2 - Example: Check Status and Update Ticket](#).

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

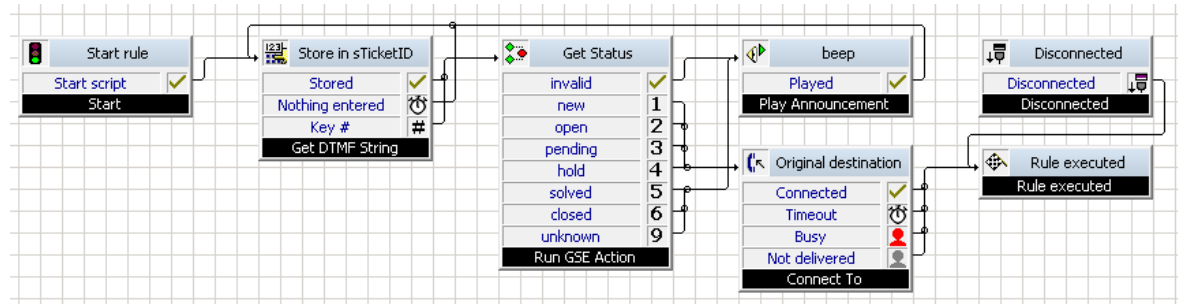
A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History



## Configure action parameters

Run GSE Action Properties

General Parameters Links

Select GSE action: Zendesk Ticket Status (System)

Set action parameters:

Name	Value	Default Value
TicketID	0	0
Zendesk Login User	dummy@user.com	dummy@user.com
Zendesk Login Token	dummy token	dummy token
Zendesk Domain	dummy.zendesk.com	dummy.zendesk.com
CURL Path	C:\Program Files (x86)\curl\c...	C:\Program Files (x86)\curl\c...

Edit Parameter...

Description:

Returns the status of a Zendesk ticket (API v2).

Return Values:

- 0 - invalid ticket id
- 1 - new
- 2 - open
- 3 - pending

OK Cancel Help

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

### **TicketID**

The Zendesk ID of the ticket to check the status from. The ID can be given with or without the Zendesk typical # prefix, i.e. **1234** or **#1234**.

### **TicketIDMinLength** 1.3.0

The minimum length of a valid Zendesk Ticket ID. The default value is 4.  
This prevents the Zendesk API to return huge result lists.

### **Zendesk Login User**

Login user. To connect to Zendesk a Zendesk user login is required. This project uses a username/token authentication instead of username/password. This is a more robust solution, and keeps the call routing script independent from any password changes the Zendesk user might do. Please follow [this link](#) to learn how to obtain the needed login token.

### **Zendesk Login Token**

Login token. To connect to Zendesk a Zendesk user login is required. This project uses a username/token authentication instead of username/password. This is a more robust solution, and keeps the call routing script independent from any password changes the Zendesk user might do. Please follow [this link](#) to learn how to obtain the needed login token.

### **Zendesk Domain**

Your Zendesk domain URL. This is something like "yourcompanyname.zendesk.com".

### **Configure action exits**

Run GSE Action Properties

General Parameters Links

Visible	Fixed name	Link name	Linked to
<input checked="" type="checkbox"/>	Default	invalid	beep
<input checked="" type="checkbox"/>	Return code 1	new	Original destination
<input checked="" type="checkbox"/>	Return code 2	open	Original destination
<input checked="" type="checkbox"/>	Return code 3	pending	Original destination
<input checked="" type="checkbox"/>	Return code 4	hold	Original destination
<input checked="" type="checkbox"/>	Return code 5	solved	beep
<input checked="" type="checkbox"/>	Return code 6	closed	beep
<input type="checkbox"/>	Return code 7		[no link]
<input type="checkbox"/>	Return code 8		[no link]
<input checked="" type="checkbox"/>	Return code 9	unknown	beep
<input type="checkbox"/>	Disconnected		Disconnected

OK Cancel Help

#### Exit 0 (Default)

This exit will be reached if an invalid issue id was given or any kind of problem occurred. It is recommended to name this exit "**invalid**".

If you reach this exit you can refer to [3.6 - Trouble Shooting](#) to figure what went wrong.

#### Exit 9

This exit will be reached if an unknown status was returned by the Zendesk REST API. It is recommended to name this exit "**unkown**".

If you reach this exit you can refer to [3.6 - Trouble Shooting](#) to figure the returned status and modify your status mapping.



## Dynamic Status Mapping 1.5.0

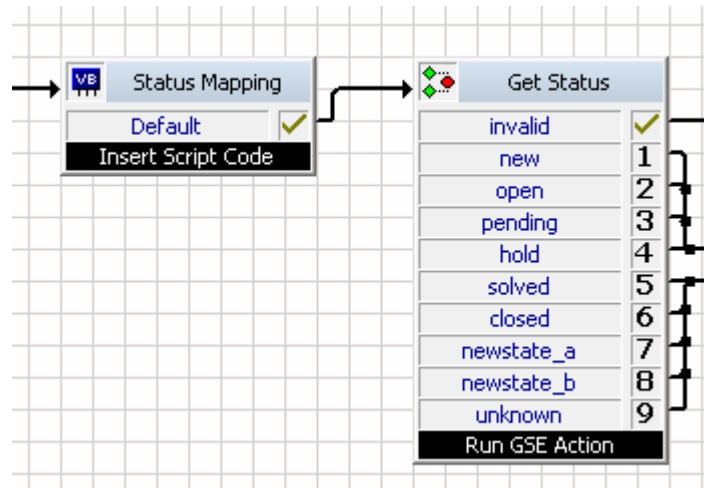
This GSE action provides a flexible mapping of Zendesk ticket status values to exits of the block. All block exits **1 to 8** are of your free choosing.

By default this GSE action maps the status values to exits as follows. If nothing changes on the Zendesk side, you don't need to do any modifications. If however Zendesk makes changes to their status values or you want another mapping of status values to the exit, you can freely do so. This is the default mapping:

Exit	States
1	new
2	open
3	pending
4	hold
5	solved
6	closed

As already mentioned, you can modify this mapping to your precise needs. Just keep in mind that only the exits **1 to 8** are available to you.

All you need to do is to modify a global variable before you use the **Check Status** GSE action. This can best be done in a **Insert Script Code** block.



The content of the Insert Script Code block is then the definition of the mapping list, which is either a **VBScript Array** or a **Lua Table**. In both cases you are free to map any number of states to any exit (1..8).

Lets assume you want to map new states **newstate\_a** to exit **7** and **newstate\_b** to exit **8**. This is the code you need to place into the **Insert Script Code** block:

For **VBScript** based call routing:

```
UseExit = 0

g_aZendeskTicketStatusMapping = Array( _
    "1;new", _
    "2;open", _
    "3;pending", _
    "4;hold", _
    "5;solved", _
    "6;closed", _
```

```
"7;newstate_a", _  
"8;newstate_b")
```

For **Lua** based call routing:

```
UseExit = 0  
  
g_aZendeskTicketStatusMapping = {  
    "1;new",  
    "2;open",  
    "3;pending",  
    "4;hold",  
    "5;solved",  
    "6;closed",  
    "6;newstate_a",  
    "6;newstate_b"  
}
```

As you can see, you are absolutely free in terms of the mapping of the different possible **states** of your service issue to an **exit** of the **Run GSE Action** block.

All you need to do is to do your mapping in the **Insert Script Code** block and afterwards open and name the **exits** of the **Run GSE Action block** accordingly.

An example of some own status mapping can be found in [A.6 - Example: Check and Announce Status](#).

**Additional return values (as global variables)**

#### **g\_sLatestZendeskTicketID** (string)

This global variable holds the ID of the latest checked ticket after the **ok** (0) exit has been reached.

#### **g\_sLatestZendeskTicketStatus** (string) 1.5.0

This global variable holds the status of the latest checked ticket after the **ok** (0) exit has been reached.

The [A.6 - Example: Check and Announce Status](#) makes use of this variable to announce the status via [AzureTTS \(text-to-speech\)](#).

#### **g\_sLatestZendeskTicketModified** (string) 1.5.0

This global variable holds the latest modification date of the latest checked ticket after the **ok** (0) exit has been reached.

The date is as the Zendesk API provides it (in [UTC](#) time), e.g. "2024-09-06T15:52:48Z".

#### **g\_sLatestZendeskTicketModifiedReadable** (string) 1.5.0

This global variable holds the latest modification date of the latest checked ticket after the **ok** (0) exit has been reached.

The date is in a better readable in most importantly in local time, e.g. "06.09.2024 17:52:48".

The [A.6 - Example: Check and Announce Status](#) makes use of this variable to announce the latest modification date via [AzureTTS \(text-to-speech\)](#).



By Tom Wellige

May 14, 2022

Share

[Previous Page](#)  
[3.2 - GSE Action Zendesk Open Ticket](#)

[Next Page](#)  
[3.4 - GSE Action Zendesk Update Ticket](#)

Theme ▼

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

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

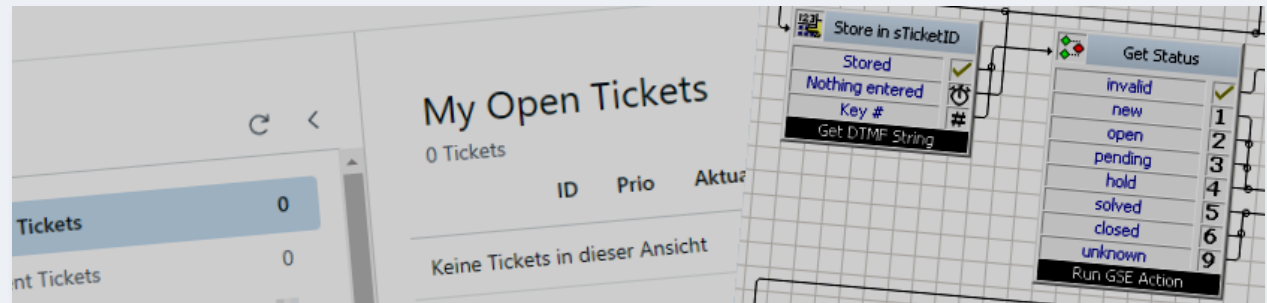
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket
- 3.3 - GSE Action Zendesk Ticket Status
- 3.4 - GSE Action Zendesk Update Ticket**
- 3.5 - Zendesk SwyxIt! button
- 3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - 3.4 - GSE Action Zendesk Update Ticket

Followers

0

VBScript

Lua

This GSE action can be used to update (i.e. add a comment to it) a given Zendesk ticket from within the call routing.

An example call routing script can be found in [A.4 - Example: Update Ticket](#) or a more advanced example in [A.2 - Example: Check Status and Update Ticket](#).

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

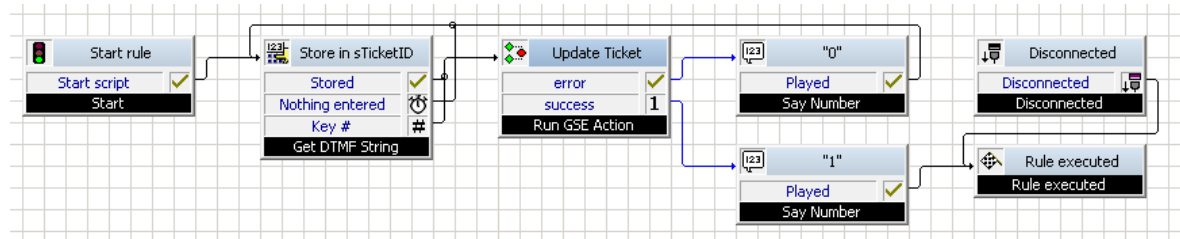
A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History



## Configure action parameters

Run GSE Action Properties

General Parameters Links

Select GSE action: Zendesk Update Ticket (System)

Set action parameters:

Name	Value	Default Value
TicketID	0	0
Comment	= ""	= ""
Zendesk AuthorID	0	0
Zendesk Login User	dummy@user.com	dummy@user.com
Zendesk Login Token	dummy token	dummy token
Zendesk Domain	dummy.zendesk.com	dummy.zendesk.com

Edit Parameter...

Description:

Updates a given Zendesk ticket (API v2).

Return Values:

0 - error

1 - success

Parameters:

TicketID: 0

OK Cancel Help

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

#### **TicketID**

The Zendesk ID of the ticket to update. The ID can be given with or without the Zendesk typical # prefix, i.e. **1234** or **#1234**.

#### **TicketIDMinLength** (from v1.3.0 on)

The minimum length of a valid Zendesk Ticket ID. The default value is 4.  
This prevents the Zendesk API to return huge result lists.

#### **Comment**

The comment string to be placed into the ticket. The comment will be displayed as **none public** comment in with Zendesk.

#### **Zendesk AuthorID**

The Zendesk id of the comment author. This is according to the [Zendesk Core API](#) optional. If give, this user will be displayed as **author** of the comment. Otherwise Zendesk will display the following **authenticated user** as **updater**.

#### **Zendesk Login User**

Login user. To connect to Zendesk a Zendesk user login is required. This project uses a username/token authentication instead of username/password. This is a more robust solution, and keeps the call routing script independent from any password changes the Zendesk user might do. Please follow [this link](#) to learn how to obtain the needed login token.

#### **Zendesk Login Token**

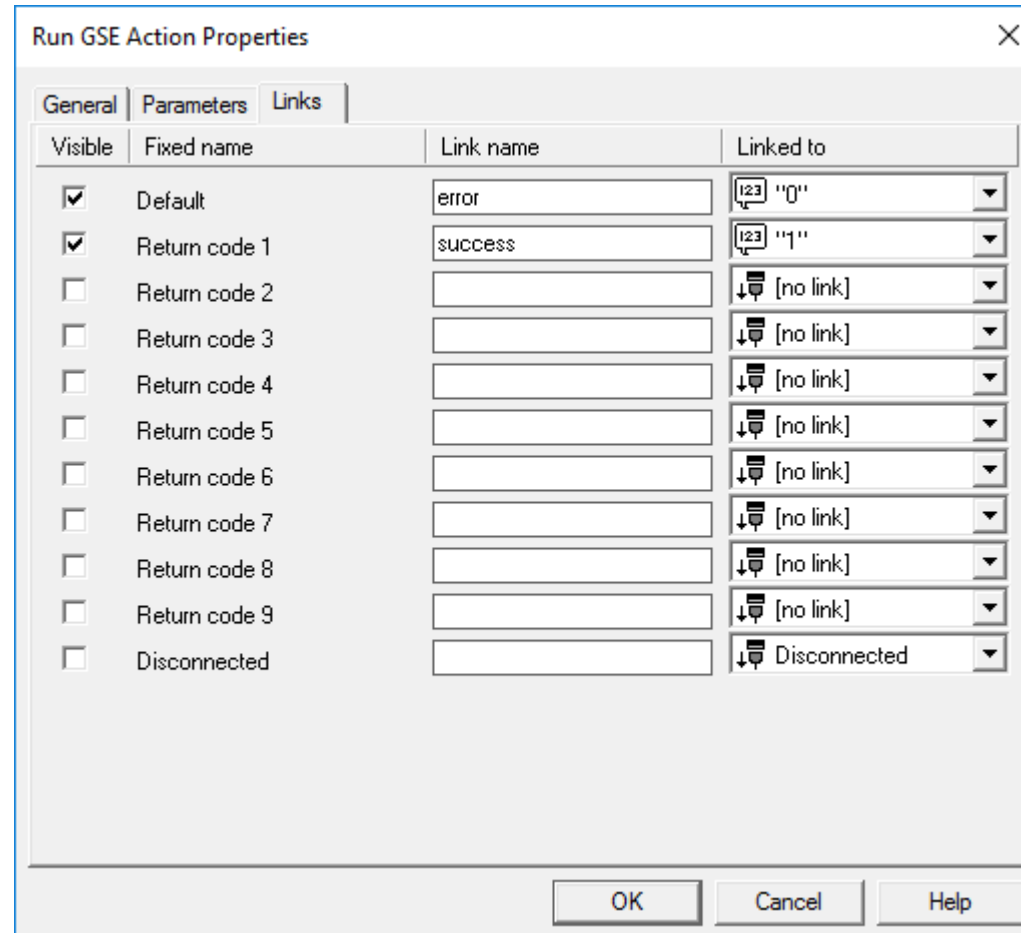
Login token. To connect to Zendesk a Zendesk user login is required. This project uses a username/token authentication instead of username/password. This is a more robust solution, and keeps the call routing script independent from any password changes the Zendesk user might do. Please follow [this link](#) to learn how to obtain the needed login token.

#### **Zendesk Domain**



Your Zendesk domain URL. This is something like "yourcompanyname.zendesk.com".

### Configure action exits



The image shows a dialog box titled "Run GSE Action Properties" with a close button (X) in the top right corner. It has three tabs: "General", "Parameters", and "Links". The "Links" tab is selected. Inside the "Links" tab, there is a table with four columns: "Visible", "Fixed name", "Link name", and "Linked to".

Visible	Fixed name	Link name	Linked to
<input checked="" type="checkbox"/>	Default	error	[123] "0"
<input checked="" type="checkbox"/>	Return code 1	success	[123] "1"
<input type="checkbox"/>	Return code 2		[no link]
<input type="checkbox"/>	Return code 3		[no link]
<input type="checkbox"/>	Return code 4		[no link]
<input type="checkbox"/>	Return code 5		[no link]
<input type="checkbox"/>	Return code 6		[no link]
<input type="checkbox"/>	Return code 7		[no link]
<input type="checkbox"/>	Return code 8		[no link]
<input type="checkbox"/>	Return code 9		[no link]
<input type="checkbox"/>	Disconnected		Disconnected

At the bottom of the dialog box, there are three buttons: "OK", "Cancel", and "Help".

#### Exit 0 (Default)

This exit will be reached on any **error** during updating the ticket. Please refer to [3.6 - Trouble shooting](#) for more information on how to do a deeper analysis of the error. You should label this exit to something like "**error**".

### Exit 1

This exit will be reached if the ticket has been **successfully** updated. You need to enable this exit (checkbox) and should label it to something like "**success**".

### Additional return value (as global variable)

**g\_LatestZendeskTicketID** (string) 1.5.0

This global variable holds the ID of the latest updated ticket after the **ok** (0) exit has been reached.



By Tom Wellige  
May 14, 2022

 Share

< Previous Page  
3.3 - GSE Action Zendesk Ticket Status

Next Page >  
3.5 - Zendesk Swyxt! button

Theme ▼

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

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

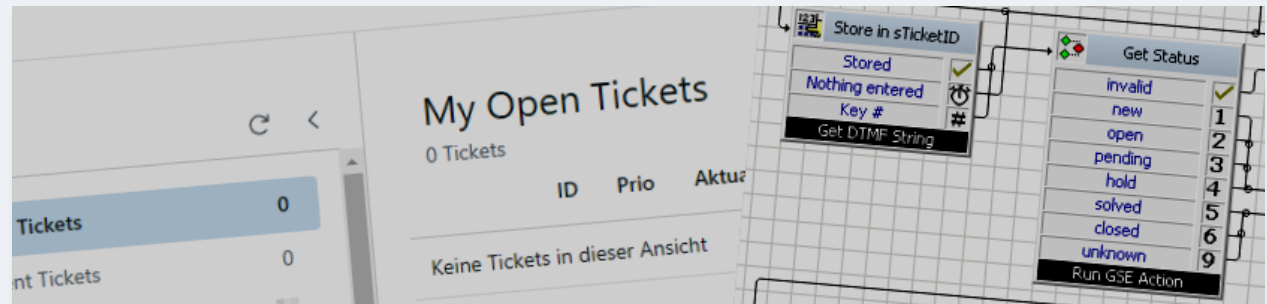
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket
- 3.3 - GSE Action Zendesk Ticket Status
- 3.4 - GSE Action Zendesk Update Ticket
- 3.5 - Zendesk SwyxIt! button**
- 3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - 3.5 - Zendesk SwyxIt! button

Followers

0

VBScript

Lua

The Zendesk Integration provides a SwyxIt! skin with a special Zendesk button.



This button directly opens a Zendesk ticket if the caller name in the SwyxIt! client contains a valid Zendesk ticket id in a special format (##nnnn).

The idea is, that a call routing script asks the caller for a valid ticket id. If the given ticket id is valid the call will be connected to an agent (or group of agents) with the given ticket id stored

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History

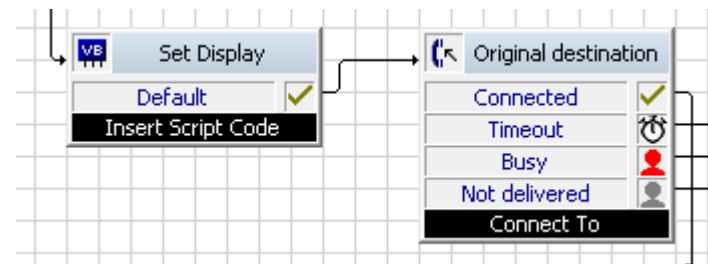
in the display. The agent is now able to click the Zendesk button to directly open that ticket.

The setup and usage is explained in chapter [2.2 - Zendesk SwyxIt! button](#).

A complete example making use of this button is shown in chapter [A.2 - Example: Check Status and Update Ticket](#).

Please note, that it is also possible to open a ticket in an already opened web browser window, in which an agent is also already logged in into his Zendesk account. An example is shown in chapter [A.5 - Example: Open Ticket](#).

Adaption in call routing script

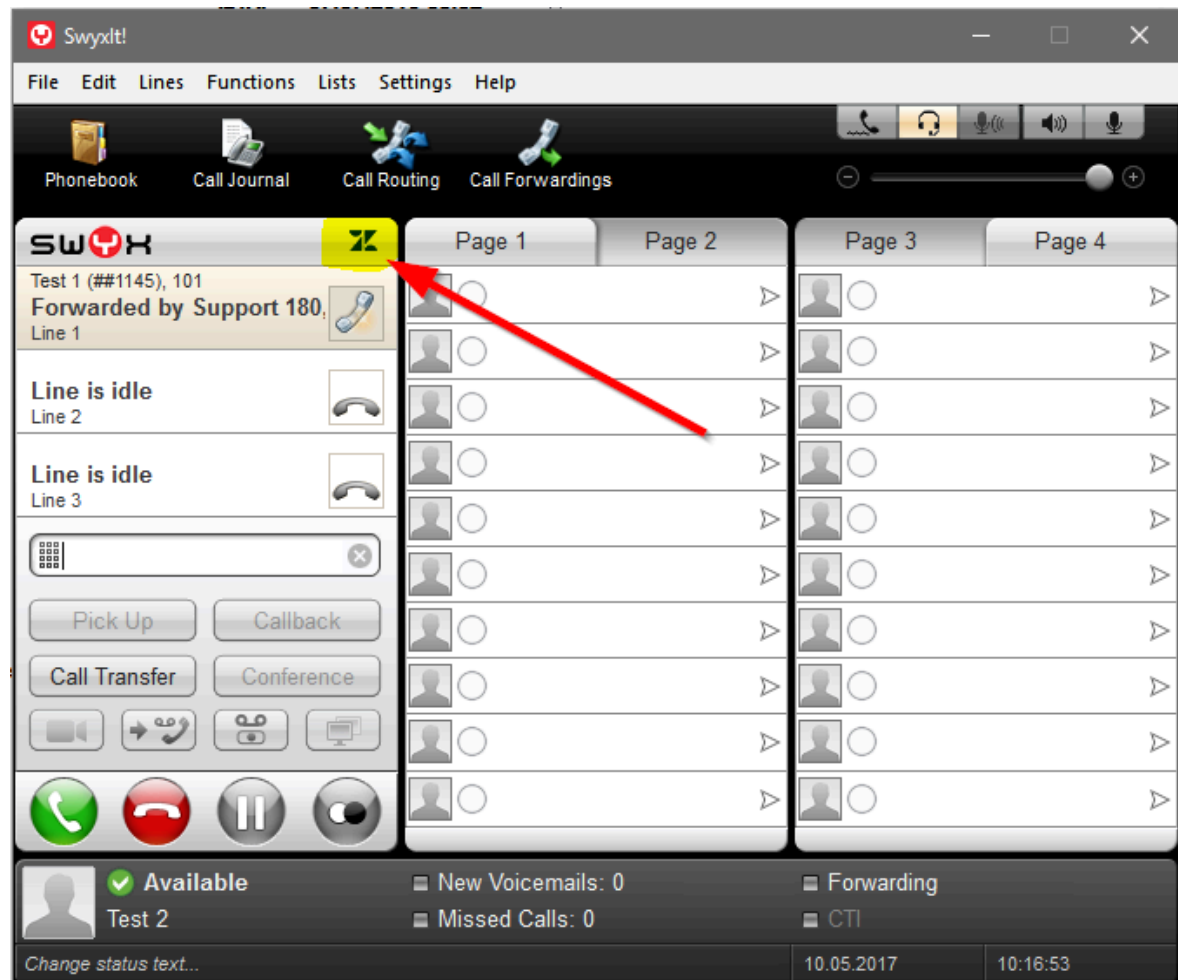


Before a call is getting connected by using the **Connect To** block, an **Insert Script Code** block must be used to modify the caller name, i.e. the name being displayed in the SwyxIt! client.

```
IpPbx.CallingName = IpPbx.CallingName & " (##" & sTicketID & ")"
```

The Zendesk ticket id must be added as shown above in **(##nnnn)** format.

**On incoming calls**



On incoming calls routed through the specially adapted call routing script simply press the Zendesk button.

Zendesk will be opened in a new browser windows (your default web browser) displaying the ticket being entered by the caller.



By Tom Wellige

May 14, 2022

 Share

< Previous Page  
3.4 - GSE Action Zendesk Update Ticket

Next Page >  
3.6 - Trouble Shooting

Theme ▼

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

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

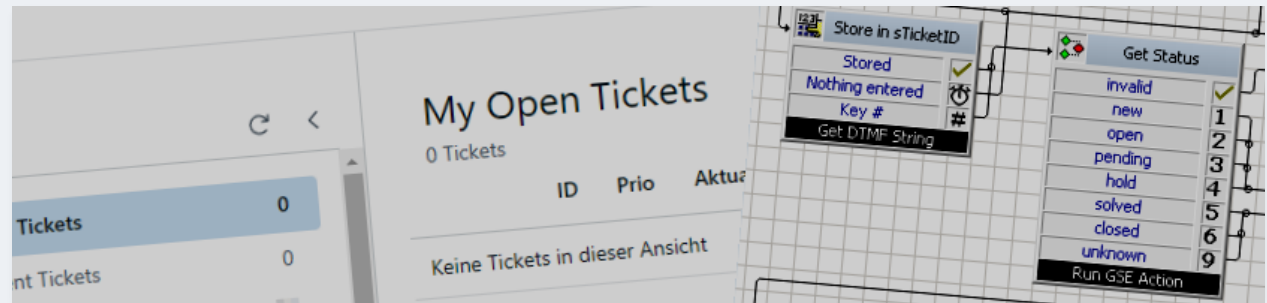
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket
- 3.3 - GSE Action Zendesk Ticket Status
- 3.4 - GSE Action Zendesk Update Ticket
- 3.5 - Zendesk SwyxIt! button
- 3.6 - Trouble Shooting**

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - 3.6 - Trouble Shooting

Followers

0

VBScript

Lua

This project communicates with a few external components.

If the provided GSE actions (or GSE example rules) don't work (you either reach the error exit or there are no changes in Zendesk) you should take a look into the server trace file, identify the call and look for trace output from the GSE action that would look like

For **VBScript** based call routing:

```
-----> Zendesk.GetTicketStatus ()  
or  
-----> Zendesk.UpdateTicket ()  
or
```

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History

```
-----> Zendesk.CreateTicket ()  
or  
-----> Zendesk.OpenTicket ()
```

For **Lua** based call routing:

```
-----> Zendesk:GetTicketStatus ()  
or  
-----> Zendesk:UpdateTicket ()  
or  
-----> Zendesk:CreateTicket ()  
or  
-----> Zendesk:OpenTicket
```

Once you have found this line for your test call you will find all important stuff before and after that line.

The following lines contains the complete command line for the curl tool. You can copy and paste it into a command line to give it a try. Before hitting enter on a command line you have to replace the **%LOGIN%** placeholder by the zendesk login, in the format **%USER%/token:%TOKEN%**. The values for the username and token can be found in the trace right above start line.

Give the complete command line a try in a browser and see if you get something meaningful back from Zendesk.

The next trace line contains the result Zendesk returned for your test call. If it looks like lots of data, then everything seems to be fine, if there is an error message included try to work out that error message until your result looks like lots of data.



I am not going to explain the JSON result format here, this is done in the [Zendesk Core API](#) documentation.

In case you have trouble getting this little project running or have specific questions, please open your own topic in the [project forum](#).



By Tom Wellige

May 14, 2022

 Share



Previous Page

[3.5 - Zendesk Swyxlt! button](#)

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

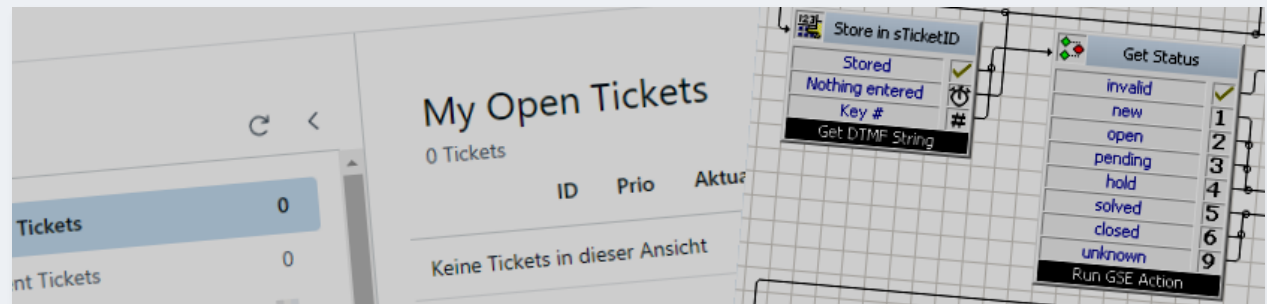
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket
- 3.3 - GSE Action Zendesk Ticket Status
- 3.4 - GSE Action Zendesk Update Ticket
- 3.5 - Zendesk SwyxIt! button
- 3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - A.1 - Example: Check Status

Followers

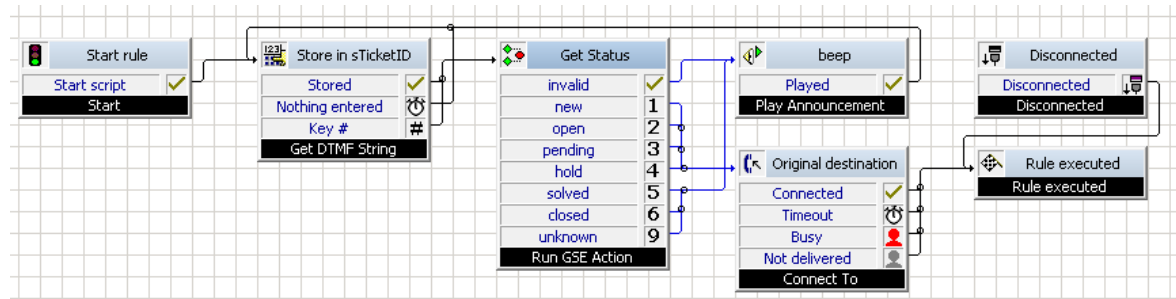
0

VBScript

Lua

This example demonstrates the usage of the **Zendesk Ticket Status** GSE action.

It asks the caller to enter a ticket id, checks if it's status is **new**, **open**, **pending** or **hold** and if so connects the call.



A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History

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. With 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\Check Status.rse
```

For **Lua** usage:

```
Lua based\rse\Check Status.rse
```

6. You need to make some modifications in the "**Get Status**" block. They are explained in detail in chapter [3.3 - GSE Action Zendesk Ticket Status](#).



By Tom Wellige  
May 14, 2022

 Share

Next Page

A.2 - Example: Check Status and Update Ticket



Theme ▼

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

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

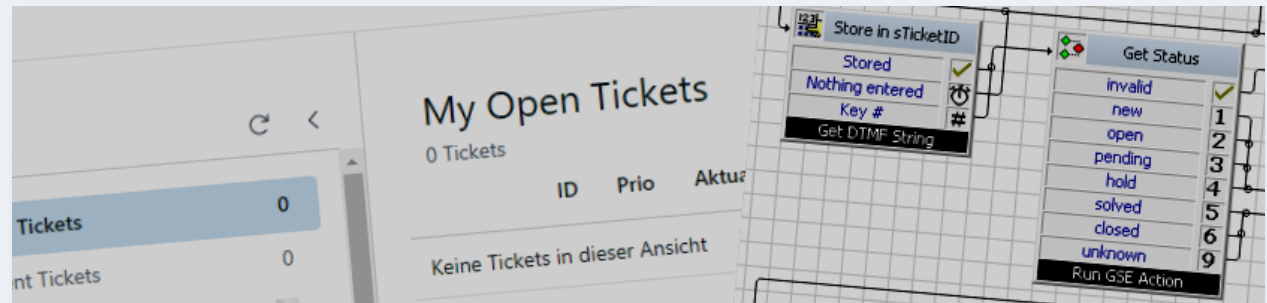
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket
- 3.3 - GSE Action Zendesk Ticket Status
- 3.4 - GSE Action Zendesk Update Ticket
- 3.5 - Zendesk SwyxIt! button
- 3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - A.2 - Example: Check Status and Update Ticket

Followers

0

VBScript

Lua

This example demonstrates the usage of the **Zendesk Ticket Status** and **Zendesk Update Ticket** GSE actions as also the **Zendesk SwyxIt! button**.



It asks the caller to enter a ticket id, checks if it's status is **new**, **open**, **pending** or **hold** and if so connects the call. After the call is finally disconnected it writes some call details into the

## A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

A.6 - Example: Check and Announce Status

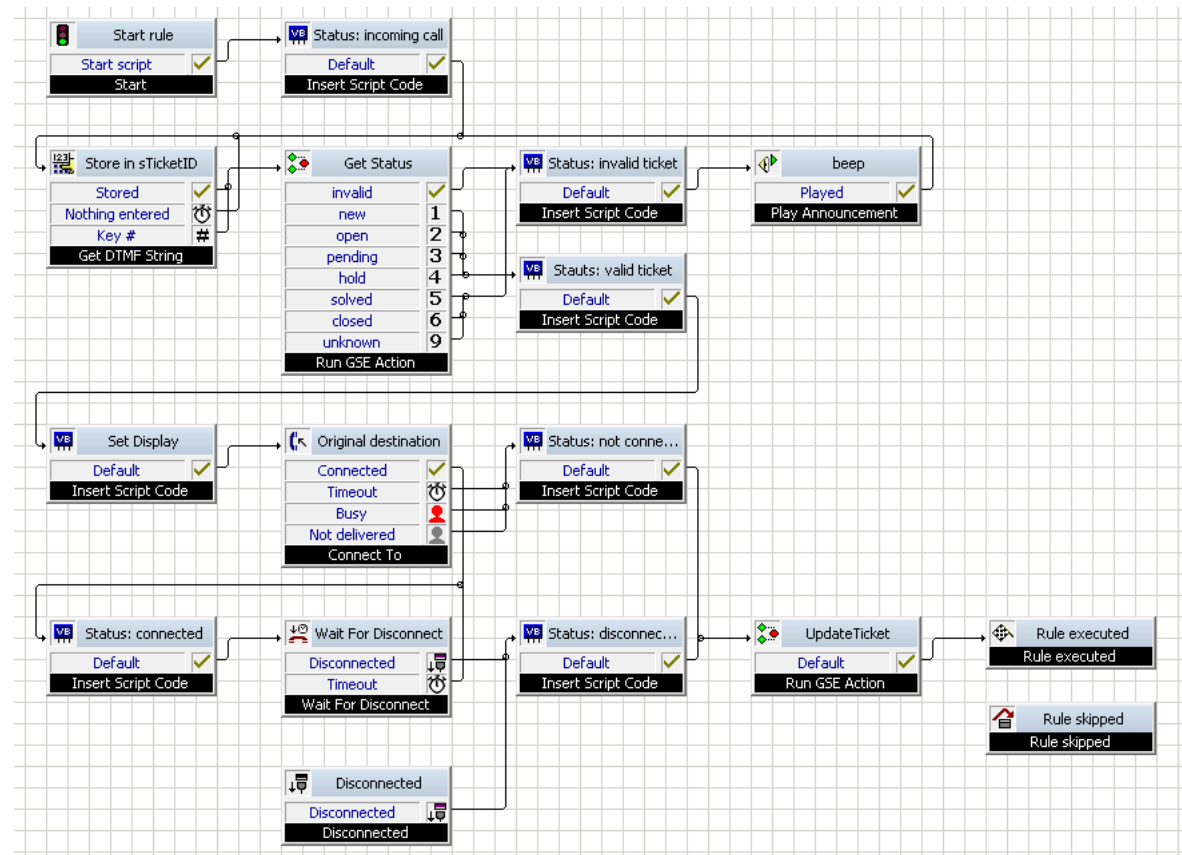
## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History

entered ticket.



To install it:

1. Make sure that the **Zendesk SwyxIt! button** is properly installed.
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. With 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\Check Status and Update Ticket.rse
```

For **Lua** usage:

```
Lua based\rse\Check Status and Update Ticket.rse
```

7. You need to make some modifications in the "**Get Status**" block. They are explained in detail in chapter [3.3 - GSE Action Zendesk Ticket Status](#).
8. You also need to make some modifications in the "**Update Ticket**" block. They are explained in detail in chapter [3.4 - GSE Action Zendesk Update Ticket](#).

After a call has been completely handled by the call routing script and also afterwards by an agent (and finally disconnected), you will find the following call details added as a new comment into the ticket:

```
10.05.2017 10:16:21 : Incoming call from 'Test1 (##1145)', 101
10.05.2017 10:16:50 : Ticket validated.
10.05.2017 10:16:59 : Call connected to Agent 1
10.05.2017 10:18:31 : Call connected to Agent 2
10.05.2017 10:21:03 : Call disconnected.
```



By Tom Wellige

May 14, 2022

 Share



Previous Page

A.1 - Example: Check Status

Next Page



A.3 - Example: Create Ticket

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community



## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

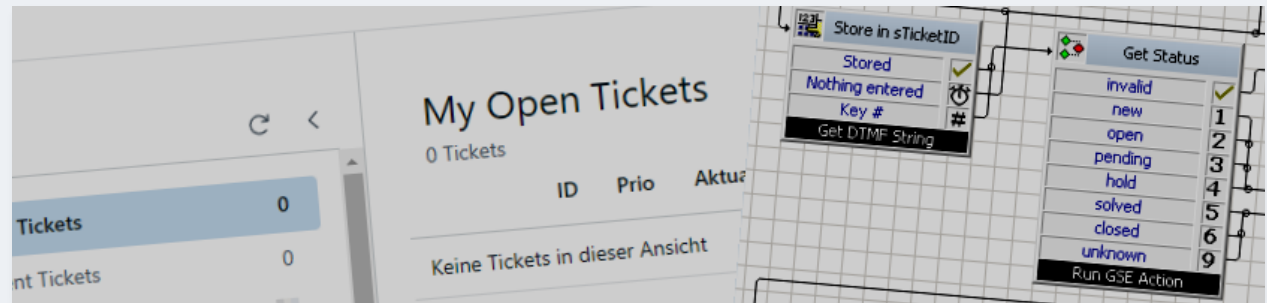
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket
- 3.3 - GSE Action Zendesk Ticket Status
- 3.4 - GSE Action Zendesk Update Ticket
- 3.5 - Zendesk SwyxIt! button
- 3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - A.3 - Example: Create Ticket

Followers 0

VBScript

Lua

This example demonstrates the usage of the **Zendesk Create Ticket** GSE action.

On any incoming call it creates a new ticket and announces afterwards its ticket id.

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

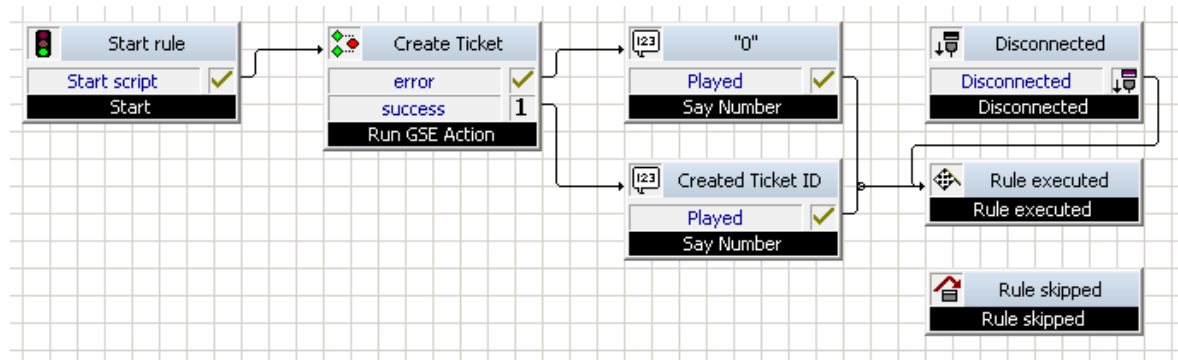
A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History



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. With 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\Create Ticket.rse
```

For **Lua** usage:

```
Lua based\rse\Create Ticket.rse
```

6. You need to make some modifications in the "**Create Ticket**" block. They are explained in detail in chapter [3.1 - GSE Action Zendesk Create Ticket](#).



By Tom Wellige

May 14, 2022

 Share

< Previous Page  
A.2 - Example: Check Status and Update Ticket

Next Page >  
A.4 - Example: Update Ticket

Theme ▼

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

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

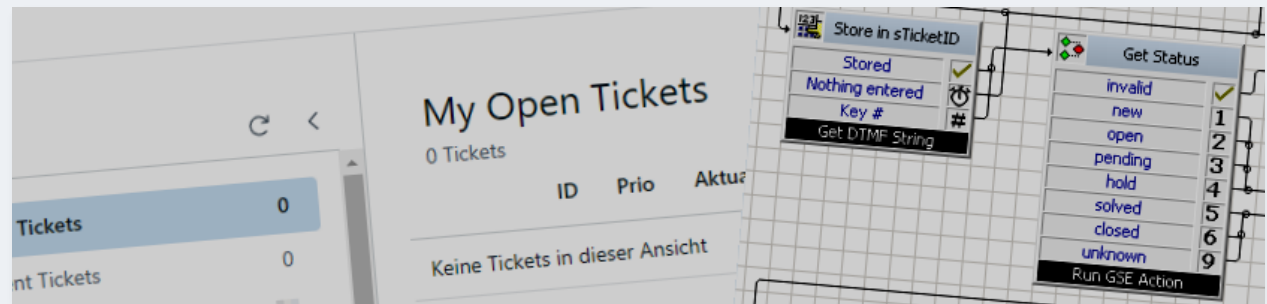
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket
- 3.3 - GSE Action Zendesk Ticket Status
- 3.4 - GSE Action Zendesk Update Ticket
- 3.5 - Zendesk SwyxIt! button
- 3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - A.4 - Example: Update Ticket

Followers

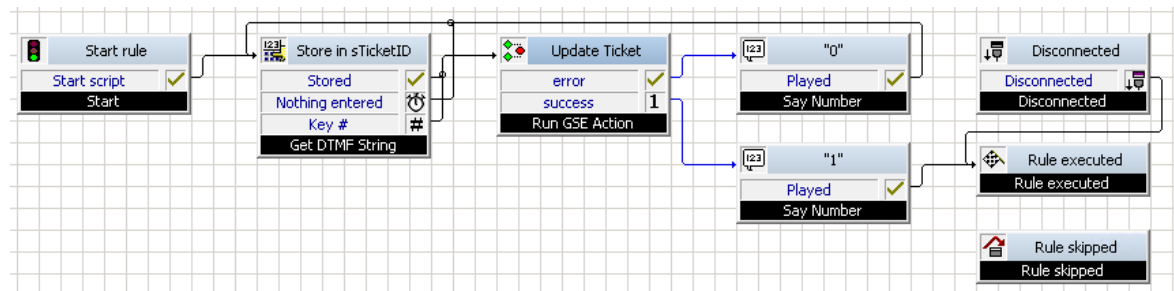
0

VBScript

Lua

This example demonstrates the usage of the **Zendesk Update Ticket** GSE action.

It asks the caller to enter a ticket id, adds a comment to that ticket and announces the result of that (0 - error, 1 - success).



A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History

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. With the GSE open the **File | Import...** menu and click **No**.
5. From the download package select the following file:

For **VBScript** usage:

```
VBScript base\rse\Update Ticket.rse
```

For **Lua** usage:

```
Lua base\rse\Update Ticket.rse
```

6. You need to make some modifications in the "**Update Ticket**" block. They are explained in detail in chapter [3.4 - GSE Action Zendesk Update Ticket](#).

You can use this to add some call details into the given ticket, like a timestamp and caller name and caller number.



By Tom Wellige

May 14, 2022

 Share



Previous Page

[A.3 - Example: Create Ticket](#)

Next Page

[A.5 - Example: Open Ticket](#)



Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

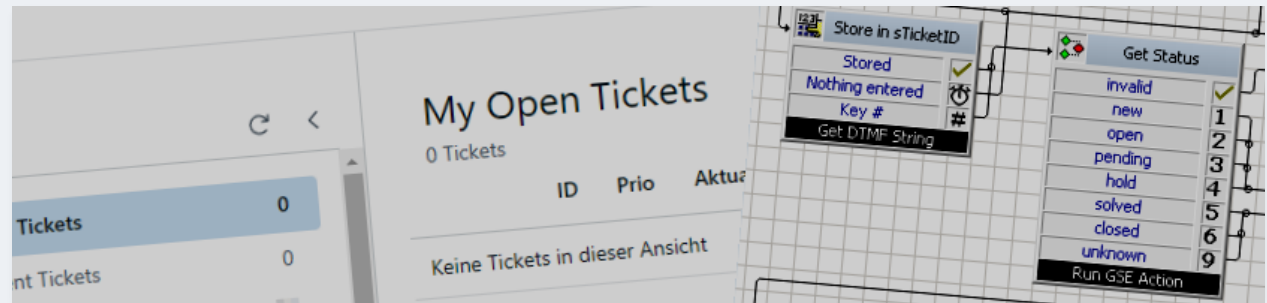
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket
- 3.3 - GSE Action Zendesk Ticket Status
- 3.4 - GSE Action Zendesk Update Ticket
- 3.5 - Zendesk SwyxIt! button
- 3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - A.5 - Example: Open Ticket

Followers

0

VBScript

Lua

1.4.0

This example demonstrates the usage of the **Zendesk Open Ticket** GSE action.

On any incoming call it opens the given ticket in the agent's web browser. In order to have this working, the agent needs to have already at least one web browser window opened in which he is logged in into his Zendesk account.

Please note: due to a limitation of the Zendesk Partner Talk API, all web browser windows the agent has currently opened in which he is logged into Zendesk will change to the given ticket.

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

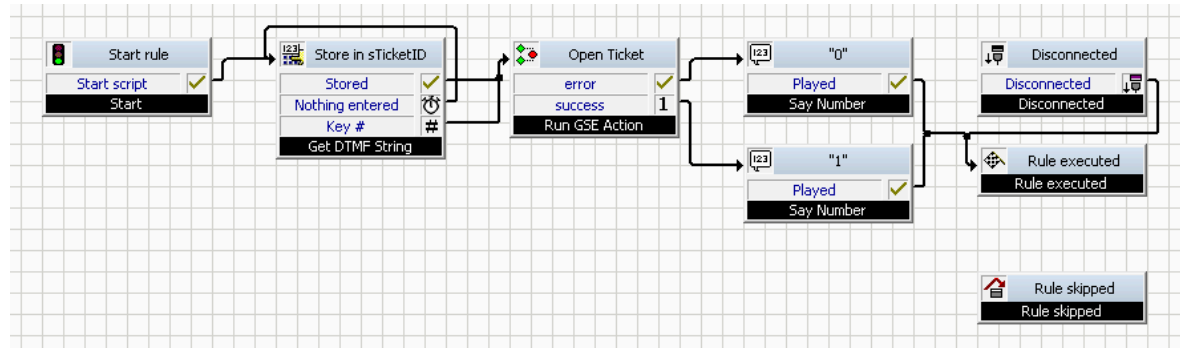
A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History



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. With 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\Open Ticket.rse
```

For **Lua** usage:

```
Lua based\rse\Open Ticket.rse
```



6. You need to make some modifications in the "**Open Ticket**" block. They are explained in detail in chapter [3.2 - GSE Action Zendesk Open Ticket](#).



By Tom Wellige

May 28, 2022

 Share

< Previous Page  
A.4 - Example: Update Ticket

Next Page >  
A.6 - Example: Check and Announce Status

Theme ▼

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

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

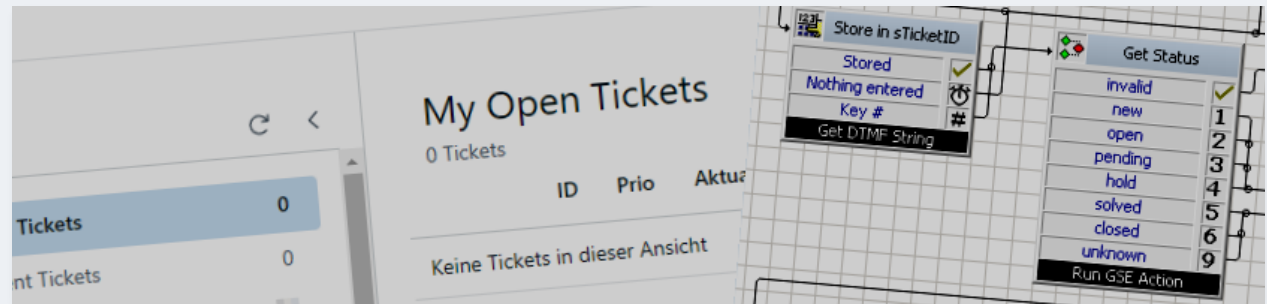
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket
- 3.3 - GSE Action Zendesk Ticket Status
- 3.4 - GSE Action Zendesk Update Ticket
- 3.5 - Zendesk SwyxIt! button
- 3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - A.6 - Example: Check and Announce Status

Followers

0

VBScript

Lua

1.5.0

This example demonstrates the usage of the **Zendesk Ticket Status** and **AzureTTS (text-to-speech)** GSE actions.

You need to have the **AzureTTS (text-to-speech)**. Open ECR Extension installed beside the Zendesk Integration.

It asks for a ticket id (close id with #), checks the status and announces its details.

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

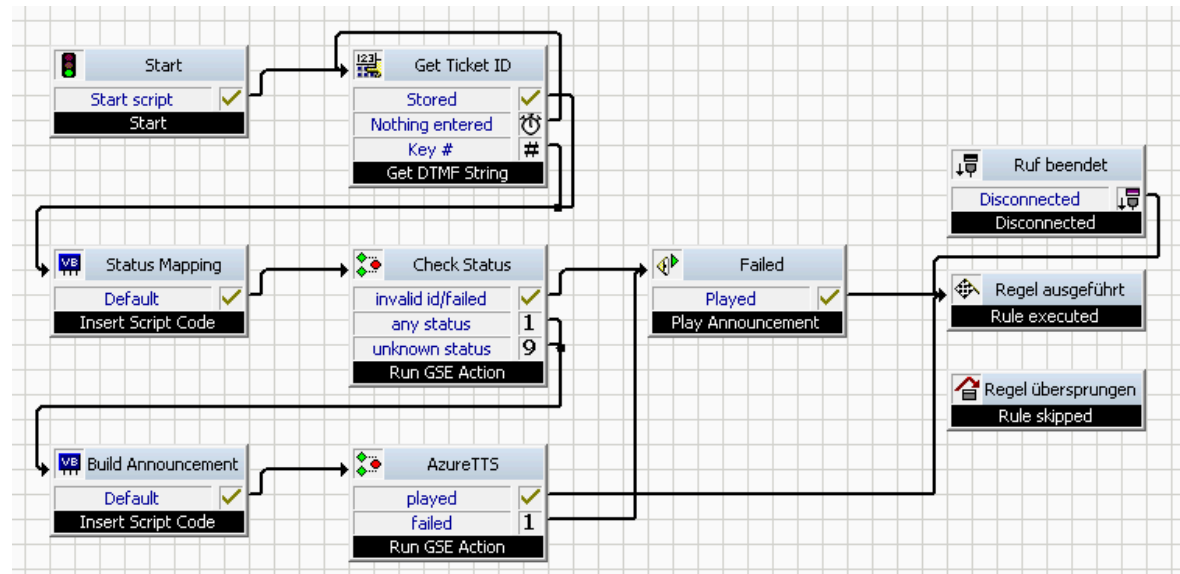
A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History



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:

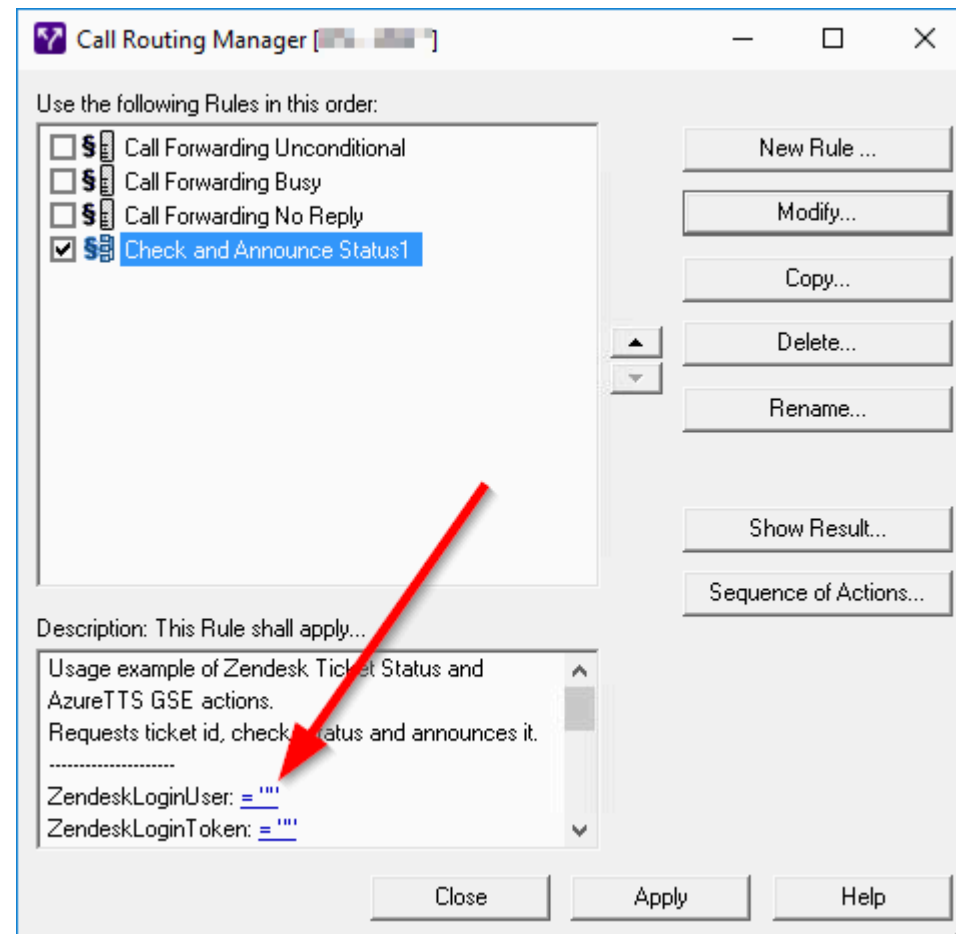
For **VBScript** usage:

VBScript based\rse\Check and Announce Status.rse

For **Lua** usage:

Lua based\rse\Check and Announce Status.rse

6. All needed configuration is done via **GSE Rule Parameters** directly in the **Call Routing Manager**.



For all **Zendesk...** parameters please refer to [Zendesk 3.3 - Ticket Status](#) for instructions.  
For all **Azure...** parameter please refer to [AzureTTS 3.1 - Usage](#) for instructions.

### **EnterTicketID**

Name of the announcement wav file to be played when you have to enter the Zendesk ticket id.

It should announce something like "Please enter your Zendesk ticket id (the trailing number only) and finish with the # (hash) key."

Default: *beep.wav*

### **StatusAnnouncement**

The text of the status announcement. You can make use of placeholders, to insert current values:

*#ID#* - the Zendesk ticket id

*#STATUS#* - the current status of the issue

*#UPDATE#* - the date/time of the latest modification of the ticket

Default: *The current status of ticket #ID# is #STATUS# and it was modified latest at #UPDATE#*

### **FailedAnnouncement**

Name of the announcement wav file to be played if either the Zendesk or the Azure request fails.

Please refer to [Zendesk 3.6 - Trouble Shooting](#) or [AzureTTS 3.2 - Trouble Shooting](#) for information on how to figure what went wrong in detail.

Default: *beep.wav*

### **Hint:**

It makes a lot of sense to use the **AzureTTS (text-to-speech)** extension once to create the announcement files for **EnterTicketID** and **FailedAnnouncement**. This ensures that the entire call routing uses the same voice.

To do so, just use the [A.1 - Example: Announce Text](#) example and 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  
September 7, 2024

 Share



Previous Page  
[A.5 - Example: Open Ticket](#)

Theme ▼

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

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

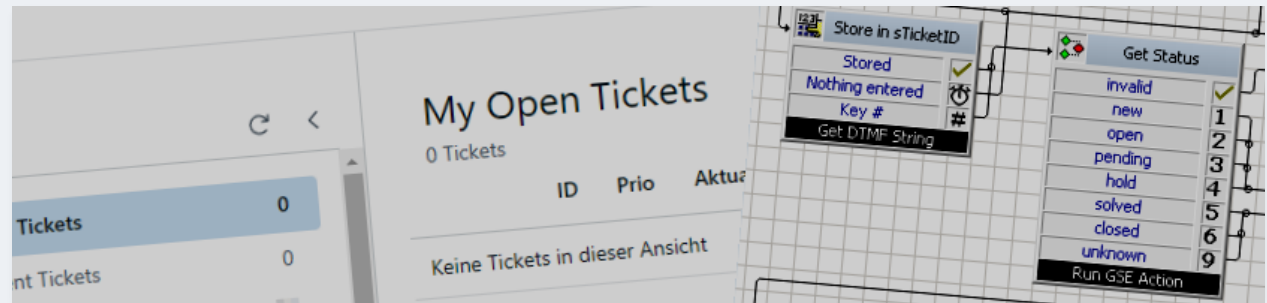
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket
- 3.3 - GSE Action Zendesk Ticket Status
- 3.4 - GSE Action Zendesk Update Ticket
- 3.5 - Zendesk SwyxIt! button
- 3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - B.1 - [Obsolete] Ticket Validation against Zendesk

Followers

0

Is obsolete since v1.1.0

The first version of this project was called **Ticket Validation against Zendesk** (v1.0.0) and only consisted of a GSE action to check the current status of a given ticket.

This has been completely replaced by the **Zendesk Integration** (from v1.1.0 upwards) and is therefore obsolete.

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History



By Tom Wellige

May 14, 2022

Share

Next Page

B.2 - Access/modify the code behind



Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community



## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

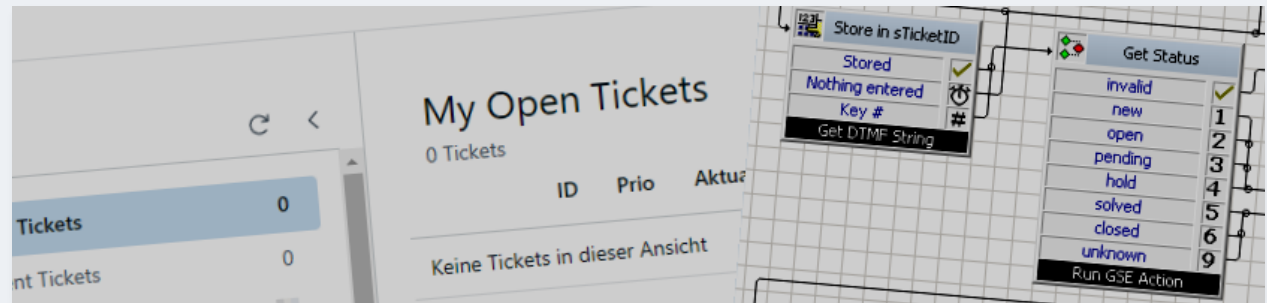
- 2.1 - Setup Zendesk GSE Actions
- 2.2 - Setup Zendesk SwyxIt! button

### USAGE

- 3.1 - GSE Action Zendesk Create Ticket
- 3.2 - GSE Action Zendesk Open Ticket
- 3.3 - GSE Action Zendesk Ticket Status
- 3.4 - GSE Action Zendesk Update Ticket
- 3.5 - Zendesk SwyxIt! button
- 3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status



## Zendesk Integration - B.2 - Access/modify the code behind

Followers

0

How to make your own modifications or simply just take a look onto the implementation to get an idea of how everything works?

The following explanations will refer to the **VBScript** version, the **Lua** version however is more or less identical in its structure.

The entire code of this integration is located in one single file, **Zendesk.vbs**.

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

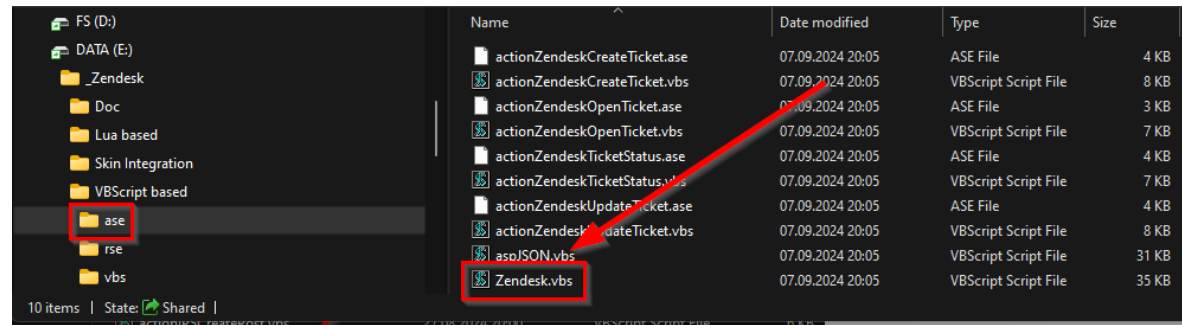
A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

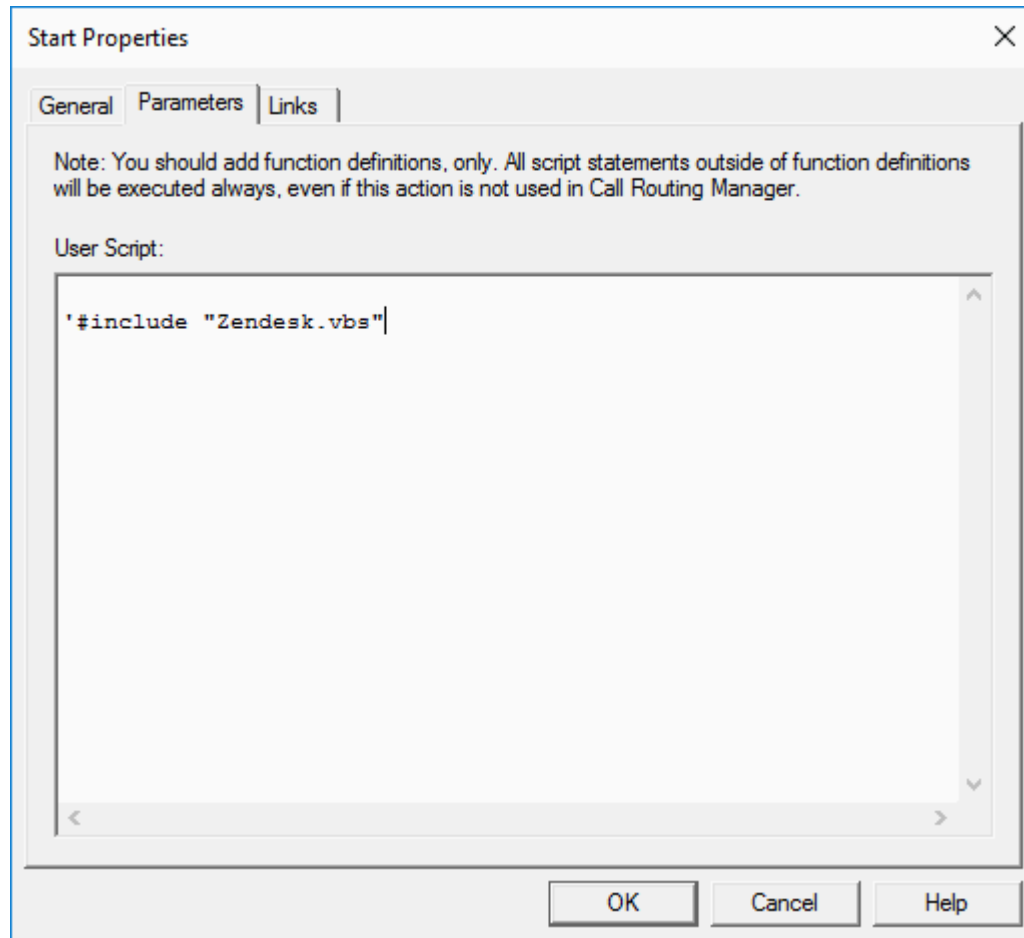
B.3 - Version History



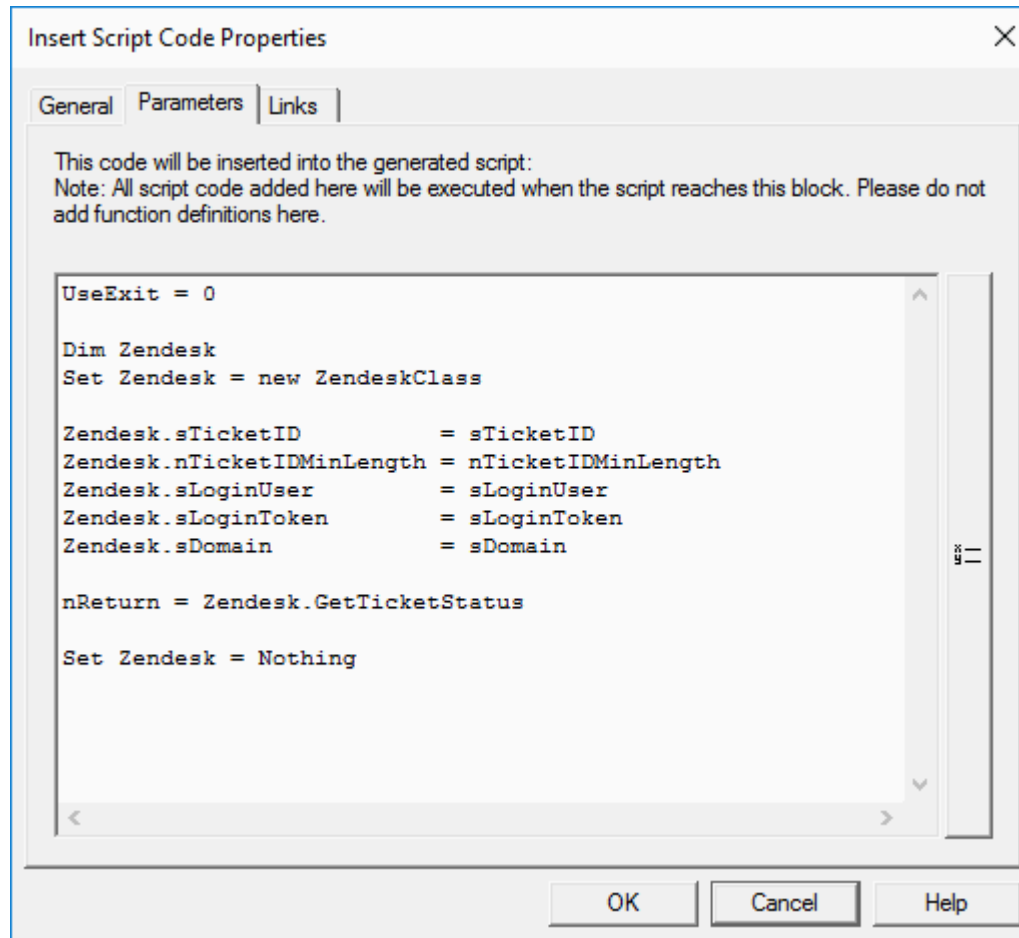
All functionality is encapsulated in a class (ZendeskClass), which defines properties for all parameters (with some validation) as also private and public methods.

To make use of the ZendeskClass the **Zendesk.vbs** file must be included first. This is done in the "**Start**" block of the GSE Action or your own GSE rule.

```
'#include "Zendesk.vbs"
```



Afterwards the class needs to get instantiated, its properties set and the needed public method/function called. The return value of the function is either "0" (ok) or "1" (failed), so it can be used directly for the exits of a "**Insert Script Code**" or "**Run GSE Action**" block.



And this is exactly what the GSE actions of this project are doing.

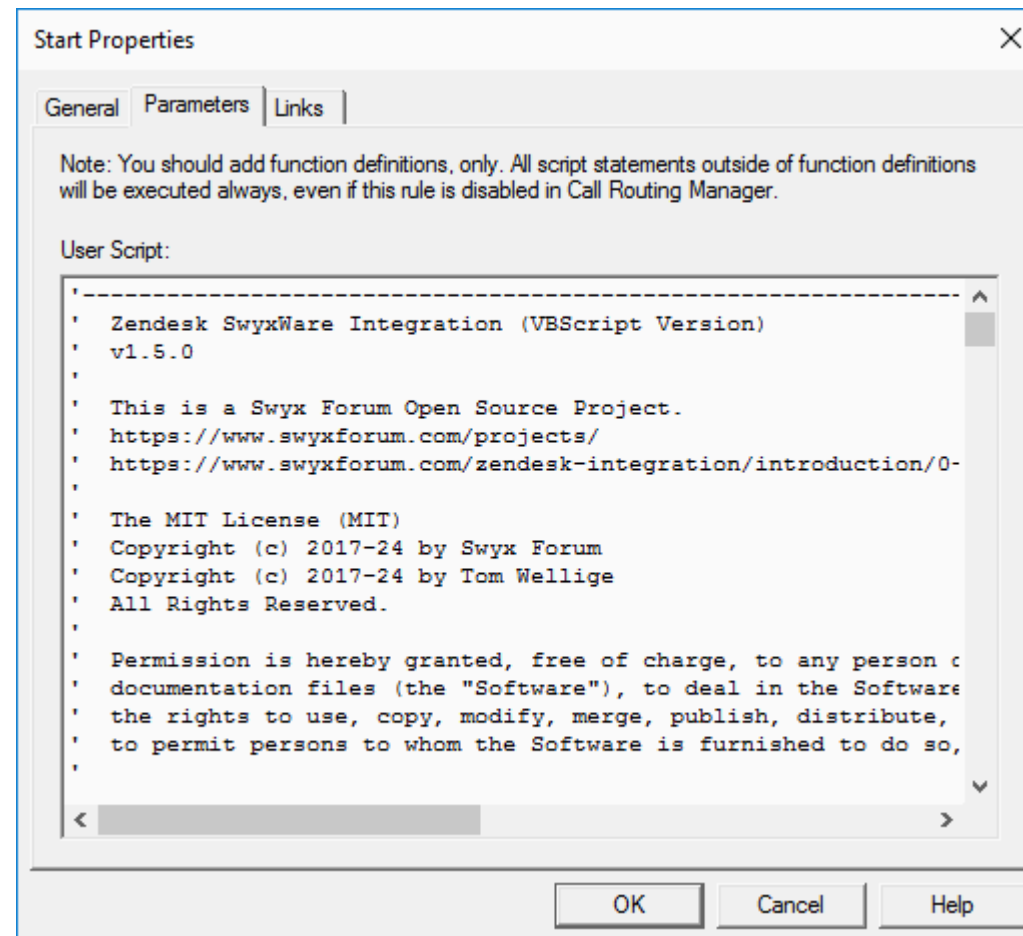
In order to make modifications it is not simply possible to modify the Zensesk.vbs file and re-upload it again into the SwyxWare database. This is because the file is digitally signed, which is necessary to get it included with the **#include** statement.

The easiest way to do your own modifications is to forget about the Zendesk.vbs file all together and use just its content within a start block of a GSE rule (instead of the #include statement), but without the digital signature.

You can strip the signature yourself from the code (first line and all lines from the bottom), or simply use the content of the start.vbs file from the "vbs" folder or the download package.

```
VBScripT based\vbs\start.vbs
```

So just do all your modifications within the **start.vbs** file and afterwards just **copy & paste** the entire content of the file into the "**Start**" block



and then use the Insert "**Insert Script Code**" block to instantiate the ZendeskClass and call its functionality

```
UseExit = 0

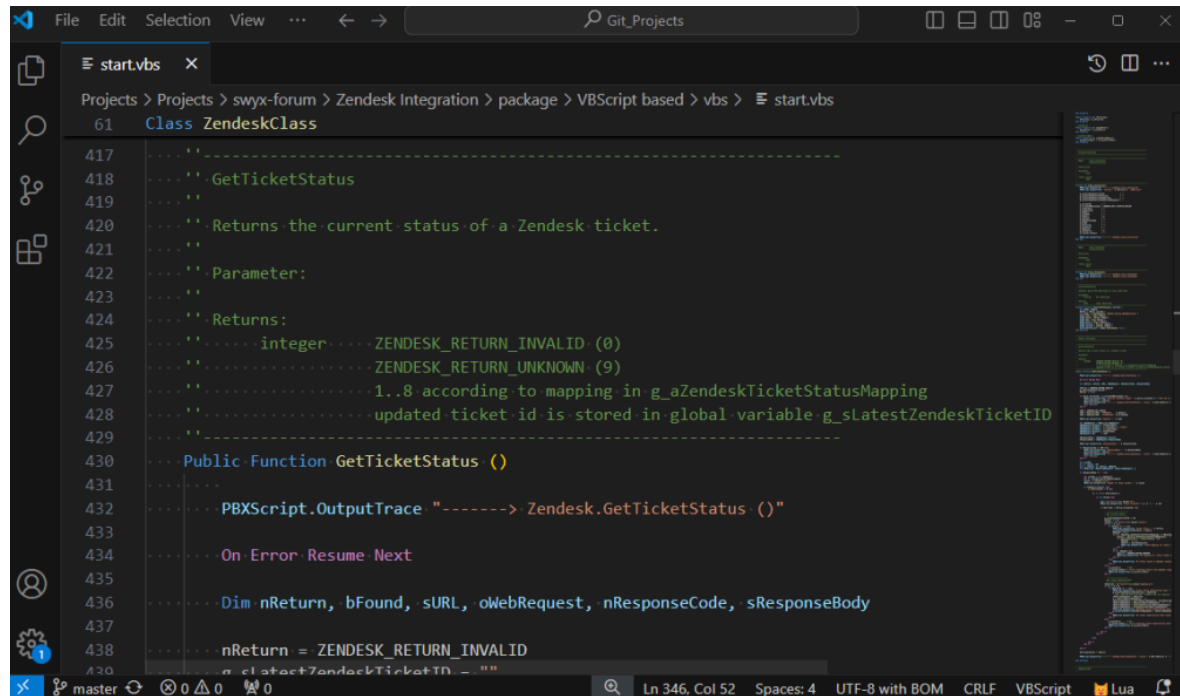
Dim Zendesk
Set Zendesk = new ZendeskClass

Zendesk.sTicketID          = "123456"
Zendesk.nTicketIDMinLength = 6
Zendesk.sLoginUser         = "xxx@yyy.com"
Zendesk.sLoginToken        = "1234567890"
Zendesk.sDomain            = "xxx.yyy.com"

UseExit = Zendesk.GetTicketStatus

Set Zendesk = Nothing
```

It is highly recommended to not do any code modifications directly in the **Start** block, but instead use a proper editor like [VS Code](#) or [Notepad++](#).

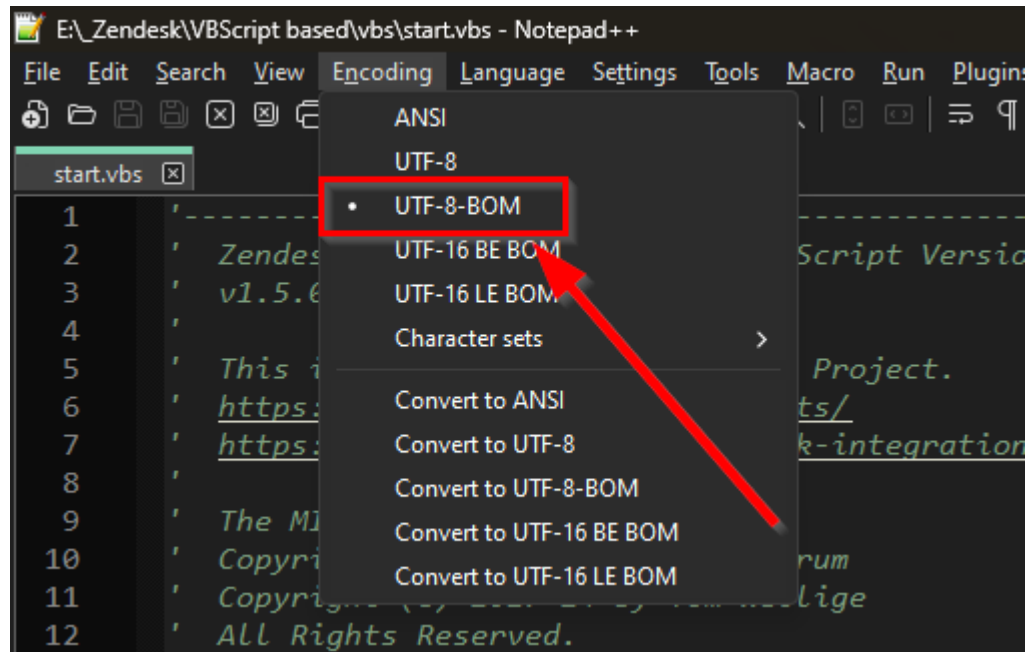


```
61 Class ZendeskClass
417 .....
418 .....GetTicketStatus
419 .....
420 .....Returns the current status of a Zendesk ticket.
421 .....
422 .....Parameter:
423 .....
424 .....Returns:
425 .....integer.....ZENDESK_RETURN_INVALID (0)
426 .....ZENDESK_RETURN_UNKNOWN (9)
427 .....1..8 according to mapping in g_aZendeskTicketStatusMapping
428 .....updated ticket id is stored in global variable g_sLatestZendeskTicketID
429 .....
430 Public Function GetTicketStatus ()
431 .....
432 .....PBXScript.OutputTrace "-----> Zendesk.GetTicketStatus ()"
433 .....
434 .....On Error Resume Next
435 .....
436 .....Dim nReturn, bFound, sURL, oWebRequest, nResponseCode, sResponseBody
437 .....
438 .....nReturn = ZENDESK_RETURN_INVALID
439 .....g_sLatestZendeskTicketID = ""
```

You are now free to to any modifications within the code and use it afterwards in your own GSE rules.

If you want to update the Zendesk.vbs file in the end, you need to update its signature as well. This can be done with the "**SignScript**" tool which can be found in the [Enreach Partner Net](#) (you need a partner login).

In order to get get your own .vbs file signed, make sure that the text encoding is **UTF-8-BOM**. This can easily be checked/configured with Notepad++.



Please feel free to ask any questions regarding the code, implementation or distribution (signing, including, etc.) in the [Open ECR Extensions forum](#).

Most of the above mentioned is also true for the following extensions. So there is lots more to explore 😊

- [Invision Power Services \(IPS\) Integration](#) (REST API usage, API Key/Token authentication)
- [Jira Service Integration](#) (REST API usage, API Key/Token authentication)
- [Longest Waiting](#) (Database usage)
- [Open Queue](#) (Database usage)





By Tom Wellige

September 7, 2024

 Share



Previous Page

B.1 - [Obsolete] Ticket Validation against Zendesk

Next Page



B.3 - Version History

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

## Zendesk Integration

### INTRODUCTION

0 - Introduction

### PREPARATIONS

1 - Preparations

### SETUP

2.1 - Setup Zendesk GSE Actions  
2.2 - Setup Zendesk SwyxIt! button

### USAGE

3.1 - GSE Action Zendesk Create Ticket  
3.2 - GSE Action Zendesk Open Ticket  
3.3 - GSE Action Zendesk Ticket Status  
3.4 - GSE Action Zendesk Update Ticket  
3.5 - Zendesk SwyxIt! button  
3.6 - Trouble Shooting

### APPENDIX A

A.1 - Example: Check Status

## Zendesk Integration - B.3 - Version History

Followers

0

### Version v1.0.0

Initial Release

### Version v1.1.0

This update replaces v1.0.0 of this extension (former name was: Ticket Validation against Zendesk).

The new version provides Zendesk ticket functionality to the SwyxWare call routing:

- Check the current status of a Zendesk ticket
- Create a new Zendesk ticket
- Update an existing Zendesk ticket
- Zendesk SwyxIt! button to open the Zendesk ticket directly with a mouse click within SwyxIt!
- Example call routing scripts.

A.2 - Example: Check Status and Update Ticket

A.3 - Example: Create Ticket

A.4 - Example: Update Ticket

A.5 - Example: Open Ticket

A.6 - Example: Check and Announce Status

## APPENDIX B

B.1 - [Obsolete] Ticket Validation against Zendesk

B.2 - Access/modify the code behind

B.3 - Version History

### Version v1.2.0

All included GSE Actions can now also be called behind the "Disconnected" block.

### Version v1.3.0

This version requires SwyxWare 12.40 (or higher) to be installed.

It comes with the following changes:

- Replacing the external open source "curl" tool by the newly introduced [PBXScript.WebRequest](#) server script api of SwyxWare 12.40.
- Validation of provided TicketID for ZendeskTicketStatus and ZendeskTicketUpdate GSE actions, by a new parameter "TicketIDMinLength" (default: 4). This prevents the Zendesk API to return huge result lists.

### Version v1.4.0

- Open an existing Zendesk ticket in an agent's web browser window
- Comes for VBscript based and Lua based call routing

### Version v1.5.0

- Added new functionality to the "Ticket Status" GSE Action:
  - added global variable "g\_sLatestZendeskTicketStatus" holding the status of the requested ticket
  - added global variable "g\_sLatestZendeskTicketModified" holding the latest modification date of requested ticket (in UTC, as provided by Zendesk)
  - added global variable "g\_sLatestZendeskTicketModifiedReadable" holding the latest modification date of requested ticket in a readable format (in local time format)
  - added dynamic mapping from status to block exit
- Added example "Check and Announce Status" to check status of ticket and announce its details via the [AzureTTS](#) Open ECR extension



By Tom Wellige

October 27, 2024

 Share



Previous Page

B.2 - Access/modify the code behind

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community