

0 - Introduction	3
1 - Preparations	8
2.1 - Setup Database	11
2.2 - Setup Database Table	17
2.3 - Setup GSE Action	21
2.4 - Setup an updated version	27
3.1 - Usage - Create a persistent variable	30
3.2 - Usage - Store a value	34
3.3 - Usage - Retrieve a value	37
3.4 - Usage - Simple Script	41
3.5 - Usage - Simple Script - Retrieve Variable	45
4.01 - Properties & Methods - Name	49
4.02 - Properties & Methods - Value	52
4.03 - Properties & Methods - Default	56
4.04 - Properties & Methods - Scope	59
4.05 - Properties & Methods - Namespace	62
4.06 - Properties & Methods - UserID	65
4.07 - Properties & Methods - GroupID	69
4.08 - Properties & Methods - CallerName	73
4.09 - Properties & Methods - CallerNumber	76
4.10 - Properties & Methods - Created	79
4.11 - Properties & Methods - CreatedByName	82
4.12 - Properties & Methods - CreatedByNumber	85
4.13 - Properties & Methods - Modified	88
4.14 - Properties & Methods - ModifiedByName	91
4.15 - Properties & Methods - ModifiedByNumber	94
4.16 - Properties & Methods - Forget	97
4.17 - Properties & Methods - LatestError	100
4.18 - Properties & Methods - LatestDescription	103
5.1 - Scopes - SCOPE_NAMESPACE	106

5.2 - Scopes - SCOPE_USER	109
5.3 - Scopes - SCOPE_GROUP	112
5.4 - Scopes - SCOPE_GLOBAL	115
6.1 - Miscellaneous - Database field definitions	118
6.2 - Miscellaneous - Master Standby Installations	121
6.3 - Miscellaneous - Use SQL Server on different machine	124
6.4 - Miscellaneous - Use database with different name (other than default name)	128
6.5 - Miscellaneous - Use different database (other than MS SQL Server, e.g. MySQL or Oracle)	131
6.6 - Miscellaneous - Usage of Persistent Variables outside of a call routing script	134
A.1 - Night Switch Manager	140
A.2 - Night Switch enabled call routing script	149
B.1 - Usage outside call routing - Simple ASP webpage	153
B.2 - Usage outside call routing - Simple VBS_WSF script	157
B.3 - Version History	161

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 0 - Introduction

Followers

0

VBScript

If you use variables in your script the content of these variables will be lost when the current call ends. Furthermore it is not possible to share the content of a variable with other simultaneously running scripts (same scripts, different scripts of the same or other user). There are lots of cases where you need to "remember" the content of a variable or simply share information between simultaneously running scripts.

Persistent Variables solve this problem. This extension provides an easy to use VBScript class (PersistentVariable) which stores its content into a database and retrieves it from there.

The visibility, i.e. scope, of a persistent variable can be configured finely grained:

- User
- Group
- Global

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

- Namespace

A variable with the scope **User** is accessible from all scripts of the current user. A user scope variable of another user with the same name does not interfere.

A variable with the scope **Group** is accessible from all scripts of the current group. A user scope variable of another group with the same name does not interfere.

A variable with the scope **Global** is accessible from all scripts of all users.

A variable with the scope **Namespace** is accessible from any script (of any user) defining the same namespace. A variable being stored in another namespace but with the same name does not interfere.

The default scope is User or Group if the variable is used in the call routing of a user or group. That means that each user and group has his own set of persistent variables without any need to worry to interfere with other users or groups call routings.

The Persistent Variables can be used with **ANY** SwyxWare version (in terms of version number). The included examples and the GSE Action itself however might be stored with the most recent SwyxWare version of the time the Persistent Variable version was released. This means that you might need to use the latest SwyxWare version to be able to open the included GSE Rules (.rse files) and the GSE Action (.ase file) within your GSE.

Aside the different SwyxWare version numbers, there are also differences on where you operate the SwyxWare. You can have it either on your own servers on your premise as also somewhere in the cloud. If you have your SwyxWare in the **cloud** (e.g. **SwyxON**) you can't make use of an own database and therefore the Persistent Variables can't be used. There is however still a possibility to use them even in a SwyxON environment. This is discussed in the following blog article:

- 6.3 - Use SQL Server on different machine
- 6.4 - Use database with different name (other than default name)
- 6.5 - Use different database (other than MS SQL Server, e.g. MySQL or Oracle)
- 6.6 - Usage of Persistent Variables outside of a call routing script

APPENDIX A

- A.1 - Night Switch - Night Switch Manager
- A.2 - Night Switch enabled call routing script

APPENDIX B

- B.1 - Usage outside call routing - Simple ASP webpage
- B.2 - Usage outside call routing - Simple VBS/WSF script
- B.3 - Version History

- [#23: Persistent Variables in the cloud \(SwyxON\)](#)

Example: Night Switch

Based upon the Persistent Variables extension this is an example on how to implement night switch functionality into your SwyxWare. There are four scripts included:

- **Night Switch Manager**
Modify the night switch status via **DTMF** menu or **Post Dialing Digits** (e.g. using speed dial keys to call into the manager to switch the night switch on or off)
- **Night Switch enabled call routing script**
A simple example on how to use the night switch persistent variable in a call routing script to differ between day and night call routing
- **WebExtension for SwyxIt! Skin**
A simple ASP web page that displays the current state of the night switch as **red** or **green** area. By clicking into this area/page the status can be toggled.
- **Shortcut for SwyxIt! Skin or Windows Desktop**
A simple WSH script which toggles the current state of the night switch.




Please refer to the [Forums](#) to discuss the Persistent Variables 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.

 You can also follow [this video](#) ([oder dieses Video](#)) which contains a Swyx webinar explaining setup and usage of the persistent variables.

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

License

Persistent Variables for SwyxWare Extended Call Routing
v1.4.0

This is a Swyx Forum Open Source Project.
<https://www.swyxforum.com/projects/>

Copyright (c) 2011-2024 by Swyx Forum
Copyright (c) 2011-2024 by Tom Wellige
All Rights Reserved.

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

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



By Tom Wellige

May 14, 2022

 Share

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 1 - Preparations

Followers

0

VBScript

Instead of reading through the following documentation you can also follow [this video](#) ([oder dieses Video](#)) which contains a Swyx webinar explaining setup and usage of the persistent variables.

[Download](#) the latest version.



By Tom Wellige

May 14, 2022

[Share](#)

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2.1 - Setup Database

2.2 - Setup Database Table

2.3 - Setup GSE Action

2.4 - Setup an updated version

USAGE

3.1 - Create a persistent variable

3.2 - Store a value

3.3 - Retrieve a value

3.4 - Simple Script - Store Variable

3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 2.1 - Setup Database

Followers

0

VBScript

It is recommended to use the [MS SQL Server](#) being already installed on the SwyxWare machine and being used by the SwyxWare.

If you want to use another MS SQL Server on another machine please refer to [6.3 - Use SQL Server on different machine](#). If you want to use another database server like [MySQL](#) or [Oracle](#) please refer to [6.5 - Use different database](#).

The default name of the database to be created is **IpPbxExtensions**. If you want to use another database name please refer to [6.4 - Use database with different name](#).

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

The following setup instruction assumes you install Persistent Variables into the MS SQL Server being installed on the SwyxWare machine and use the default database name. **This is later on the most easiest way to use Persistent Variables.**

Step 1

It is **prohibited** to use SwyxWare's **IpPbx** database for own needs. Therefore all extensions of the Call Routing Extensions project use a separated database called **IpPbxExtensions**.

To create this database you should use the [SQL Server Management Studio](#). Follow this link to download the version matching to your MS SQL Server version.

Step 2

Open **SQL Server Management Studio** and connect to the local database server.

Step 3

Right click **Databases** in the tree view and select **New Database...**

Step 4

Enter the name **IpPbxExtensions** into the top field of the dialog on click on **OK**.

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

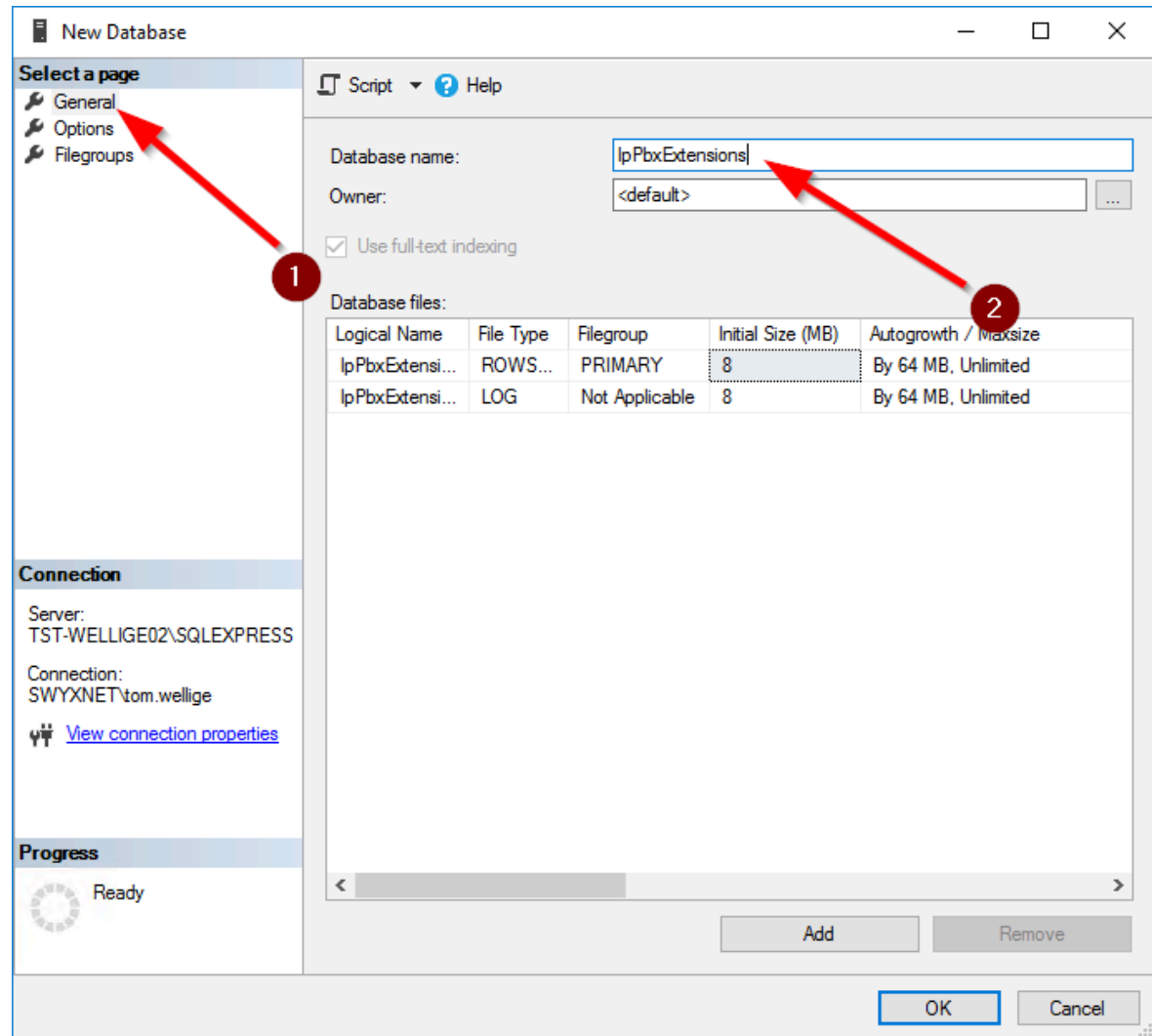
A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History



Step 5

Open the Security branch in the tree view, right click on **Logins** and select **New Login...**

Step 6

Enter or select the name of the **user account** the SwyxWare services (i.e. the SwyxServer) is running under. By default this is the local **SwyxServiceAccount**. You can also set the "Default

Database" to "IpPbxExtensions".

Login - New

Select a page

- General
- Server Role
- User Mapping
- Securables
- Status

Connection

Server: TST-WELLIGE02\SQLEXPRESS

Connection: SWYXNET\Tom.Wellige

[View connection properties](#)

Progress

Ready

Script ? Help

Login name: TST-WELLIGE02\SwyxServiceAccount Search...

☒ Windows authentication

☐ SQL Server authentication

Password:

Confirm password:

☐ Specify old password

Old password:

☒ Enforce password policy

☒ Enforce password expiration

☒ User must change password at next login

☐ Mapped to certificate

☐ Mapped to asymmetric key

☐ Map to Credential

Mapped Credentials

Credential	Provider
------------	----------

Default database: IpPbxExtensions

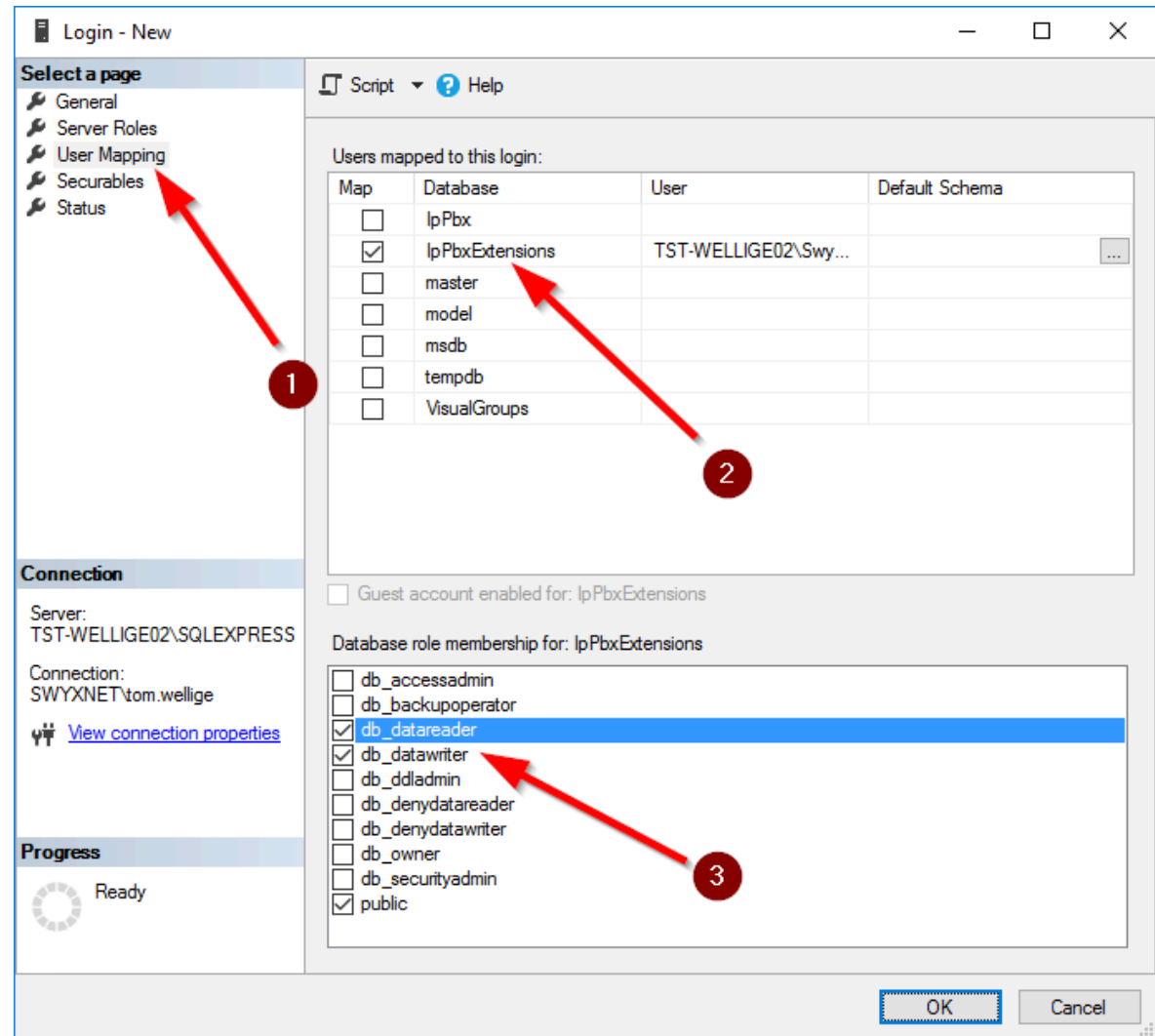
Default language: <default>

OK Cancel

Step 7

Select **User Mapping** from the left list, **check** the **IpPbxExtensions** database from the upper list on the right side and configure (check) **db_datareader** and **db_datawriter** from the lower list on the right side. You should not use **db_owner** as this would give the call routing way too

many access privileges on the database..



Step 8

Click on **OK**.

You have now successfully created the IpPbxExtensions database being used by the PersistentVariable GSE action. By creating a login for the SwyxServiceAccount you have granted access for all call routing scripts to this database.



By Tom Wellige
May 14, 2022

 Share

Next Page >
[2.2 - Setup Database Table](#)

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2.1 - Setup Database

2.2 - Setup Database Table

2.3 - Setup GSE Action

2.4 - Setup an updated version

USAGE

3.1 - Create a persistent variable

3.2 - Store a value

3.3 - Retrieve a value

3.4 - Simple Script - Store Variable

3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 2.2 - Setup Database Table

Followers

0

VBScript

After having created and configured the IpPbxExtensions database the table which is used by the PersistentVariable GSE action needs to be created into this database.

Step 1

Select **Open | Open File...** from the **File** menu of SQL Server Management Studio and select the file **CreateTable.sql** from the **sql** folder of the download package.

Hit the **Execute** button in the toolbar.

Note: if you have chosen another than the default name for the database in the previous installation step you need to modify the first line of the CreateTable.sql file accordingly.

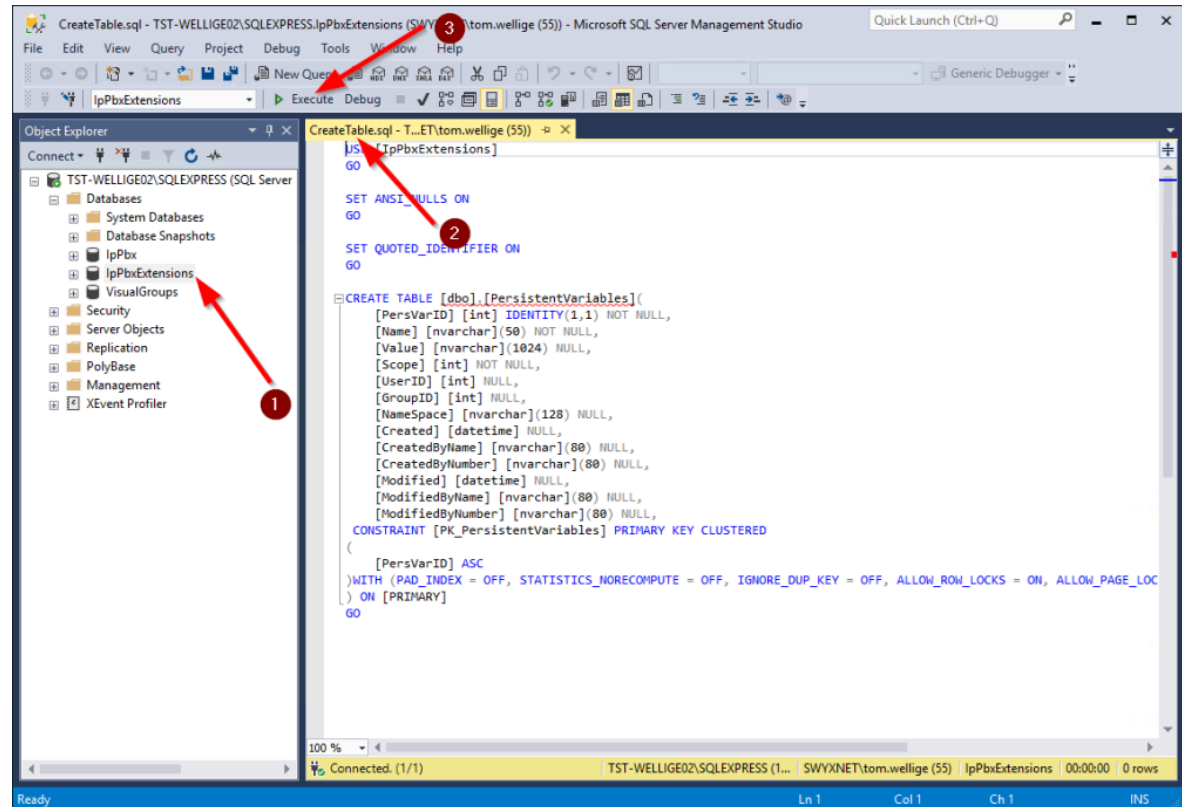
- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations



Step 2

Right click **IpPbxExtensions** in the tree view and select **Refresh**. Open the **IpPbxExtensions** and **Table** branches to check if the new table **dbo.PersistentVariables** was created.

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

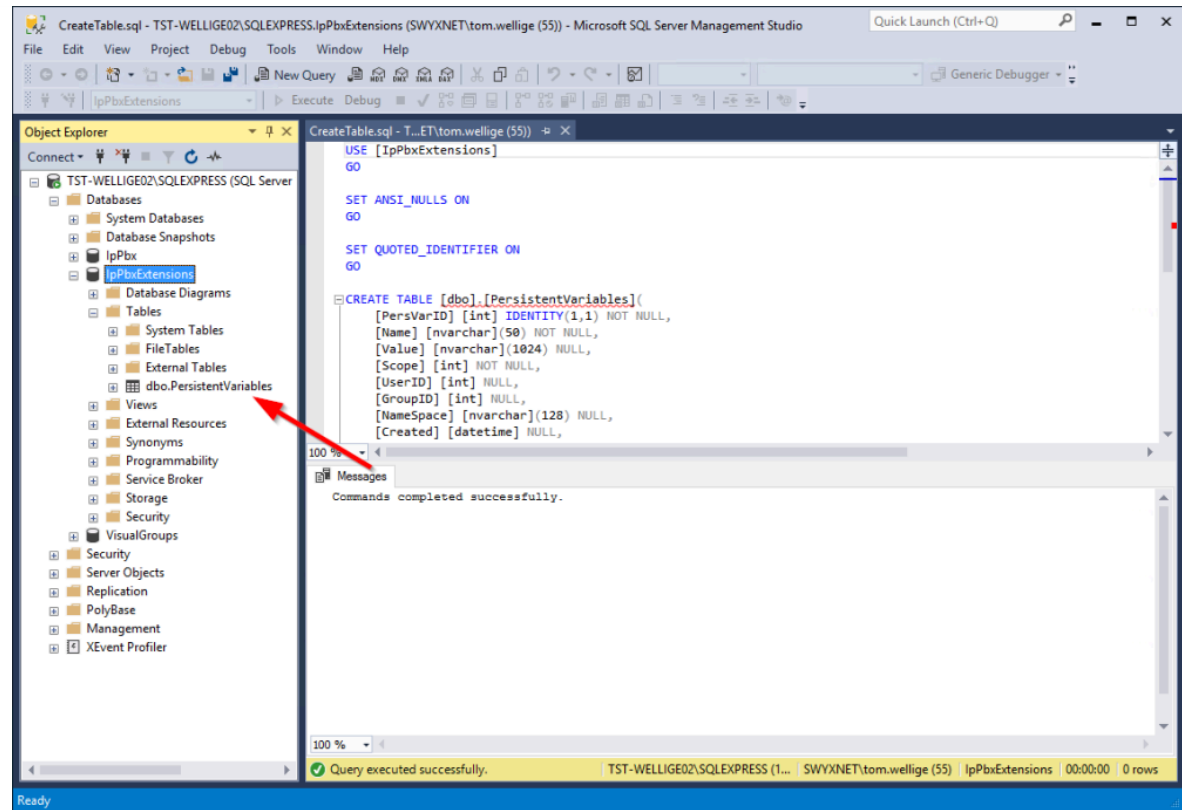
A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History



The database is now fully created and configured !



By Tom Wellige
May 14, 2022

Share

[Previous Page](#)
[2.1 - Setup Database](#)

[Next Page](#)
[2.3 - Setup GSE Action](#)

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2.1 - Setup Database

2.2 - Setup Database Table

2.3 - Setup GSE Action

2.4 - Setup an updated version

USAGE

3.1 - Create a persistent variable

3.2 - Store a value

3.3 - Retrieve a value

3.4 - Simple Script - Store Variable

3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 2.3 - Setup GSE Action

Followers

0

VBScript

The **Persistent Variables** 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....**

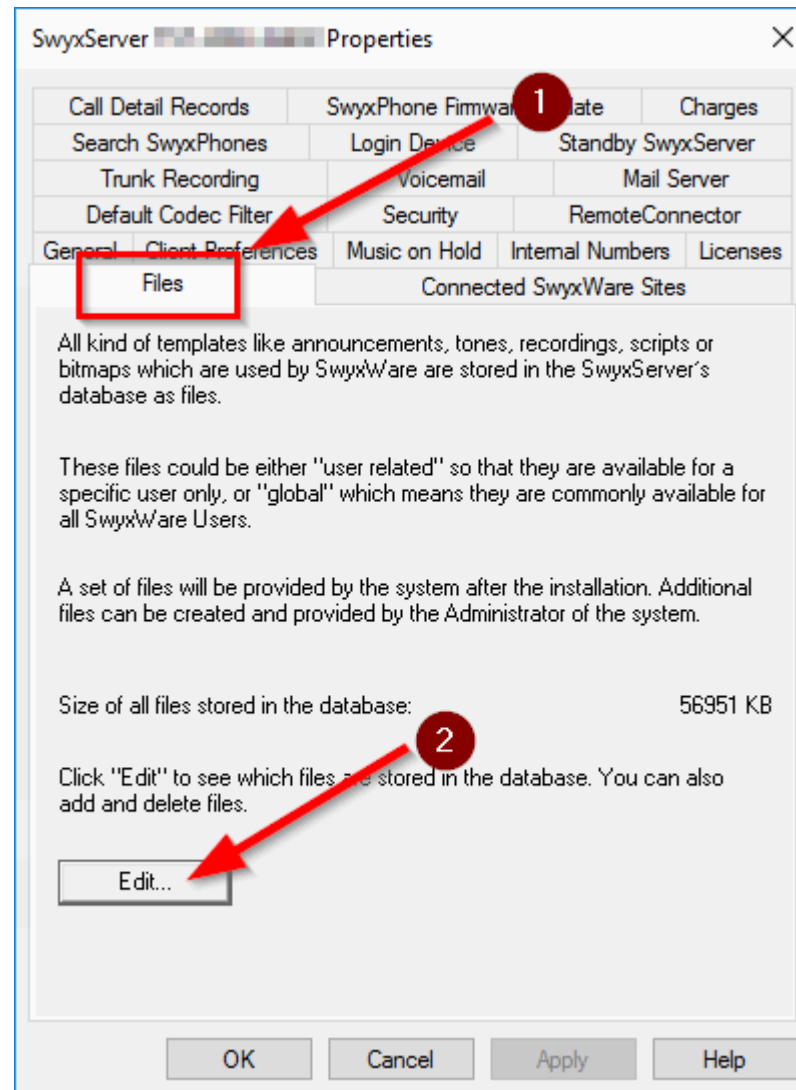
- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations



Step 3

Click on **Add...**

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

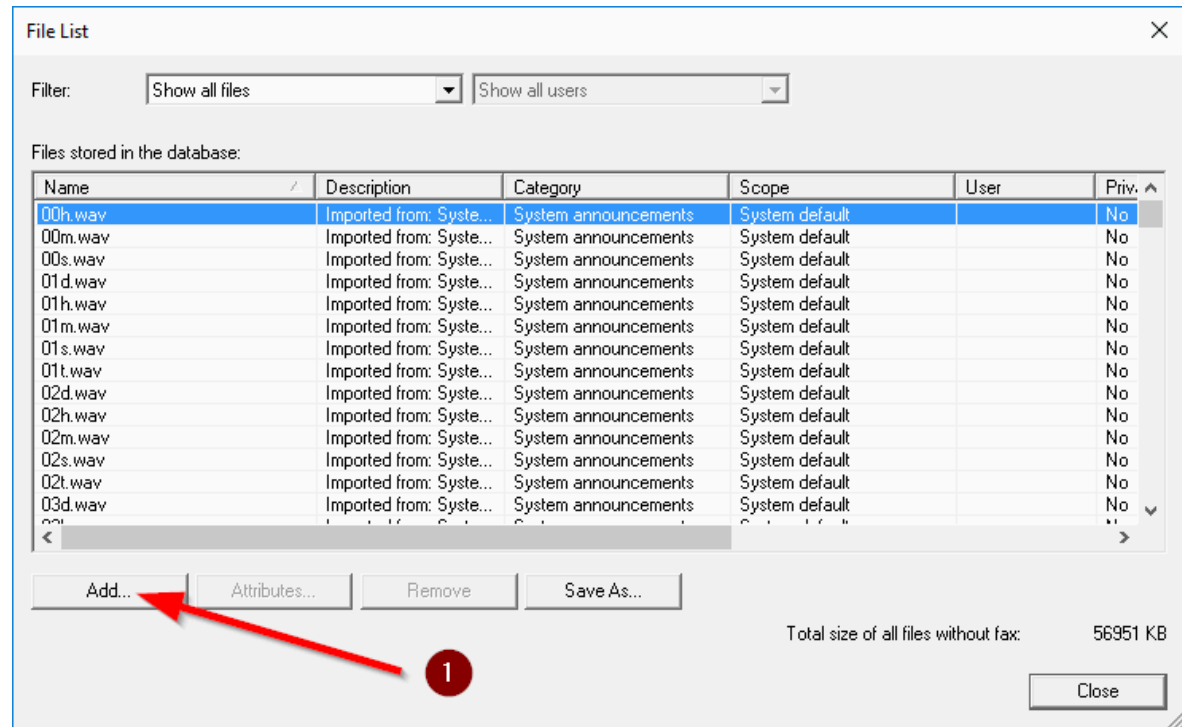
A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History



Step 4

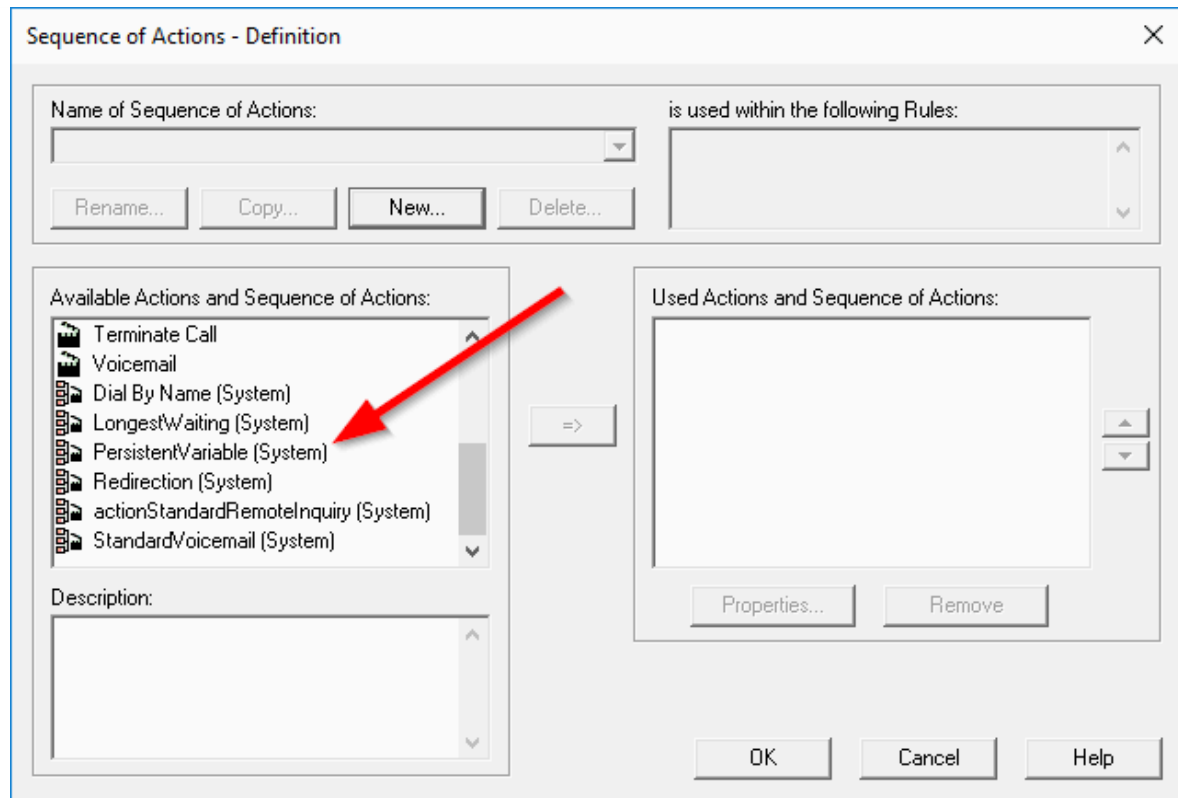
1. Select all files from the **ase** folder from the download package.
2. Select **Global** as **Scope**. (you can also select **User** to load the files into the user scope a your script user or **Group** to load the files into the group scope of your script group)
3. Selec **CallRoutingScripts** (in SwyxWare < 13.10) or **Call Routing VBS scripts** (in SwyxWare >= 13.10) as **Category**.
4. It is recommended to enter **Persistent Variables** 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 box containing the path "C:\Setup\Swyx\actionPersistentVariable.ase; C:" followed by a red circle with the number "1" and a browse button "...".
- Name:** A text box containing "actionPersistentVariable.ase; actionl".
- 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 box containing "Persistent Variables" followed 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 **Persistent Variable action**. The **(System)** behind the GSE action name shows that this is a global action being available for every SwyxWare user.



The PersistenVariable GSE action is now installed completely and persists variables can now be fully used within GSE call routing scripts.



By Tom Wellige
May 14, 2022

 Share

[Previous Page](#)
2.2 - Setup Database Table

[Next Page](#)
2.4 - Setup an updated version

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2.1 - Setup Database

2.2 - Setup Database Table

2.3 - Setup GSE Action

2.4 - Setup an updated version

USAGE

3.1 - Create a persistent variable

3.2 - Store a value

3.3 - Retrieve a value

3.4 - Simple Script - Store Variable

3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 2.4 - Setup an updated version

Followers

0

VBScript

Updating from any version less than 1.3.0 to 1.4.0

1. [Download](#) the version 1.4.0 and follow the steps described in [2.3 - Setup GSE Action](#). You do not need to delete the previously installed two action files, as the new upload will overwrite them.
2. An update of the database is needed as well. To do so follow the steps described in [2.2 - Setup Database Table](#). Instead of using the **CreateTable.sql** file you need to use the **UpdateTable_from_less_1.3.0_to_1.4.0.sql** file from the download package.

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

Updating from version 1.3.0 to 1.4.0

1. [Download](#) the version 1.4.0 and follow the steps described in [2.3 - Setup GSE Action](#). You do not need to delete the previously installed two action files, as the new upload will overwrite them,
2. An update of the database is needed as well. To do so follow the steps described in [2.2 - Setup Database Table](#). Instead of using the **CreateTable.sql** file you need to use the **UpdateTable_from_1.3.0_to_1.4.0.sql** file from the download package.



By Tom Wellige
April 15, 2023

Share

< [Previous Page](#)
[2.3 - Setup GSE Action](#)

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2.1 - Setup Database

2.2 - Setup Database Table

2.3 - Setup GSE Action

2.4 - Setup an updated version

USAGE

3.1 - Create a persistent variable

3.2 - Store a value

3.3 - Retrieve a value

3.4 - Simple Script - Store Variable

3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

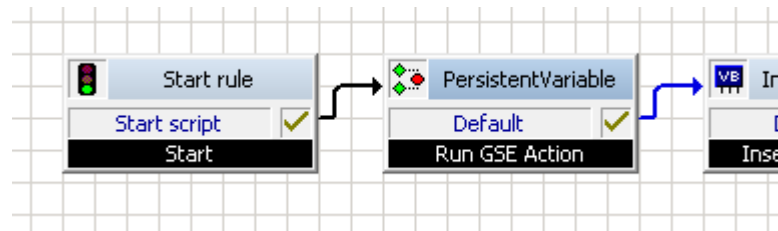
Persistent Variables - 3.1 - Create a persistent variable

Followers

0

VBScript

To be able to use persistent variables you need to add one **Run GSE Action** block into your script, calling the **PersistentVariable** action. It is recommended to place this block right after the **Start** block.



- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

The image shows a 'Run GSE Action Properties' dialog box with three tabs: 'General', 'Parameters', and 'Links'. The 'Parameters' tab is selected. It features a dropdown menu for 'Select GSE action:' set to 'PersistentVariable (System)'. Below this is a table for 'Set action parameters:' with columns 'Name', 'Value', and 'Default Value'. The table contains three rows: 'ServerName', 'DatabaseName', and 'ConnectString', each with a value of '=' and a default value of '""'. An 'Edit Parameter...' button is located below the table. At the bottom, there is a 'Description:' section with a text area containing the following text: 'Persistent Variables for SwyxWare Extended Call Routing v1.4.0', 'This is a Swyx Forum Open Source Project.', 'https://www.swyxforum.com/projects/', 'Copyright (c) 2011-2024 by Swyx Forum', and 'Copyright (c) 2011-2024 by Tom Wellige'. The dialog box has 'OK', 'Cancel', and 'Help' buttons at the bottom right.

Name	Value	Default Value
ServerName	= ''	= ''
DatabaseName	= ''	= ''
ConnectString	= ''	= ''

Description:
 Persistent Variables for SwyxWare Extended Call Routing
 v1.4.0
 This is a Swyx Forum Open Source Project.
<https://www.swyxforum.com/projects/>
 Copyright (c) 2011-2024 by Swyx Forum
 Copyright (c) 2011-2024 by Tom Wellige

If you have installed the Persistent Variables extension as recommended you don't need to set any of the properties of the GSE action. Otherwise you should refer to: [6.3 - Use SQL Server on different machine](#), [6.4 - Use database with different name](#), [6.5 - Use different database](#).

Using persistent variables does not much differ from using common VBScript variables. If you want to use own VBScript variables you do this anywhere in your VBScript code. This can either be in the **Start** block or in a common **Insert Script Code** block.

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Note: if you want to use a persistent variable within some VBScript functions you have defined in the **Start** block as also in the graphical part of the script, lets say in the **Play Accouncement** block you need to define it in the **Start** block.

The **Set Variable** block can't be used for persistent variables (as it generates common VBScript variables).

Afterwards you create persistent variables as following:

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"
```

In line 1 you define a new VBScript variable "Announcement".

In line 2 you set this variable as of type "PersistentVariable".

In line 3 you give the persistent variable a unique Name. This name identifies the variable and it is stored with this name in the database.

With nothing else configured the above persistent variable will be visible/accessible for the current script user only (the User scope is the default scope). To configure another scope (Global or Namespace) you have to change the Scope property.

Once the persistent variable is created you can use it as every other variable. In the moment you put something in it, it will be automatically stored into the database. In the moment where you want to get something out of it, it will be taken from the database.

See the following pages for examples.



By Tom Wellige
May 14, 2022

 Share

Next Page >
3.2 - Store a value

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value**
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 3.2 - Store a value

Followers

0

VBScript

As already mentioned using persistent variables is very straight forward, just like you use common VBScript variables.

The only difference is, that the value you put into a persistent variable will automatically be store into the database.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"

Announcement.Value = "Default Welcome.wav"
```

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

The content you want to store into a persistent variable needs to be put into the Value property.

You can put anything you want into the persistent variables, like strings, booleans, integers or dates. You only need to keep in mind that the maximum size is limited to **1kB** (1024 bytes).

If the above code is used in a call routing of a **user**, the variable will be visible/accessible for the current script user only (the User scope is the default scope in this case).

If the above code is used in a call routing of a **group**, the variable will be visible/accessible for the current script group only (the Group scope is the default scope) in this case.

To configure another scope you have to change the Scope property.

You will find the 3.4 - Simple Script - Store Variable in the download package which includes the above code.



By Tom Wellige
May 14, 2022

 Share



Previous Page

3.1 - Create a persistent variable

Next Page

3.3 - Retrieve a value



6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value**
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 3.3 - Retrieve a value

Followers

0

VBScript

Retrieving a persistent variable like storing it very straight forward, just like common VBScript variables.

The only difference is, that the value you read from the variable will be taken from the database in that moment.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"
Announcement.Default = "Beep.wav"
```

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

`PlaySound Announcement.Value`

Before you read the content from the persistent variable you should define a Default value. This value will be returned in case nothing has been stored previously into the persistent variable.

Reading content from a persistent variable is the same as you would use a common variable, just that you have to read the Value property.

The above code passes the content of the persistent variable as parameter into the PlaySound function (build-in function of the Call Routing Manager, is used by "Play Announcement" action of Rule Assistant).

You can also access the variable directly in nearly any of the GSE blocks:

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

The screenshot shows the 'Play Announcement Properties' dialog box with the 'General' tab selected. The 'Parameters' tab is also visible. The 'Skip from beginning' checkbox is unchecked. The 'Announcement' dropdown is set to '= Announcement.Value'. Below it are buttons for '...', 'Play', 'Stop', 'Pause', and 'Close'. The 'Play control enabled' checkbox is unchecked. The 'DTMF interruption' checkbox is unchecked. The 'Mask' dropdown is set to 'All'. Below it are checkboxes for 'Save input in variable:', 'Replace variable content' (selected), and 'Append to variable content'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

If the above code is used in a call routing of a **user**, the variable will be visible/accessible for the current script user only (the User scope is the default scope in this case).

If the above code is used in a call routing of a **group**, the variable will be visible/accessible for the current script group only (the Group scope is the default scope) in this case.

To configure another scope you have to change the Scope property.

You will find the 3.5 - Simple Script - Retrieve Variable in the download package which includes the above code (except line 6).



By Tom Wellige

May 14, 2022

 Share

< Previous Page
3.2 - Store a value

Next Page >
3.4 - Simple Script - Store Variable

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable**
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 3.4 - Simple Script - Store Variable

Followers

0

VBScript

Within the download package of this project you will find an **rse** folder. This folder includes the file **Store.rse**. This is a small example script to demonstrate the usage of persistent variables. It shows how to store a value persistently into the scope of the current user or group.

4.1 - Name

4.2 - Value

4.3 - Default

4.4 - Scope

4.5 - Namespace

4.6 - UserID

4.7 - GroupID

4.8 - CallerName

4.9 - CallerNumber

4.10 - Created

4.11 - CreatedByName

4.12 - CreatedByNumber

4.13 - Modified

4.14 - ModifiedByName

4.15 - ModifiedByNumber

4.16 - Forget

4.17 - LatestError

4.18 - LatestDescription

SCOPES

5.1 - SCOPE_NAMESPACE

5.2 - SCOPE_USER

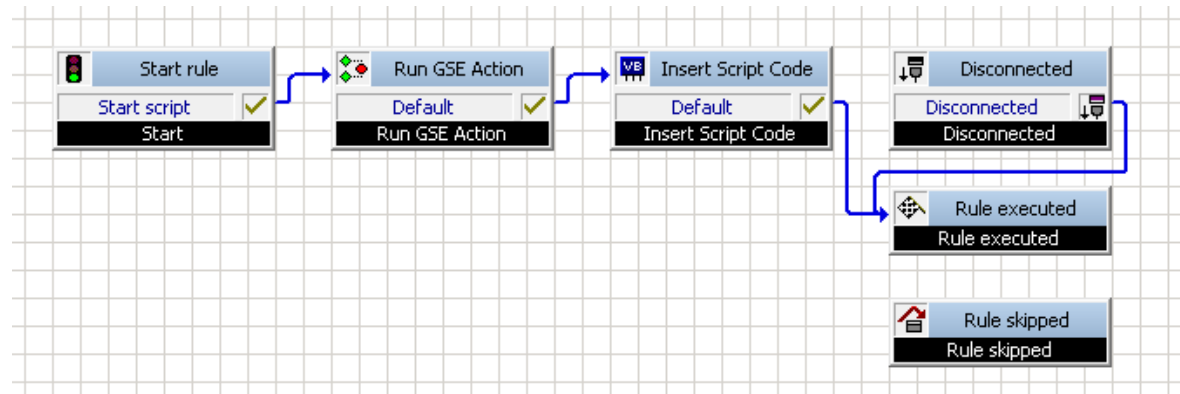
5.3 - SCOPE_GROUP

5.4 - SCOPE_GLOBAL

MISCELLANEOUS

6.1 - Database field definitions

6.2 - Master Standby Installations



Insert Script Code Properties

General Parameters Links

This code will be inserted into the generated script:
Note: All script code added here will be executed when the script reaches this block. Please do not add function definitions here.

```
' TODO: Insert you script code here

UseExit = 0 ' Please use the variable UseExit for the block

Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"

Announcement.Value = "Default Welcome.wav"
```

OK Cancel Help

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

The script creates a variable with the Name "Welcome".

If the script runs for a user, the variable will only be visible/accessible for the **current user**. The so called Default scope is User. If the script runs for a group, the variable will only be visible/accessible for the **current group**. The so called default scope is Group.

Afterwards the name of a SwyxWare default announcement (Default Welcome.wav) is stored into the new persistent variable.

To install this script generate a new GSE rule, via the menu **File | Import...** select the file **Store.rse**, save the rule, close the GSE and make sure the rule is activated to give it a try.

The next example will read this persistent value and announce it.

Hint: the Persistent Variables can be used with **ANY** SwyxWare version. The included examples and the GSE Action itself however might be stored with the most recent SwyxWare version of the time the Persistent Variable version was released. This means that you might need to use the latest SwyxWare version to be able to open the included GSE Rules (.rse files) and the GSE Action (.ase file) within your GSE.



By Tom Wellige
May 14, 2022

 Share

[Previous Page](#)
3.3 - Retrieve a value

[Next Page](#)
3.5 - Simple Script - Retrieve Variable

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 3.5 - Simple Script - Retrieve Variable

Followers

0

VBScript

Within the download package of this project you will find an **rse** folder. This folder includes the file **Retrieve.rse**. This is a small example script to demonstrate the usage of persistent variables. It shows how to retrieve a persistently stored value from the scope of the current user or group.

4.1 - Name

4.2 - Value

4.3 - Default

4.4 - Scope

4.5 - Namespace

4.6 - UserID

4.7 - GroupID

4.8 - CallerName

4.9 - CallerNumber

4.10 - Created

4.11 - CreatedByName

4.12 - CreatedByNumber

4.13 - Modified

4.14 - ModifiedByName

4.15 - ModifiedByNumber

4.16 - Forget

4.17 - LatestError

4.18 - LatestDescription

SCOPES

5.1 - SCOPE_NAMESPACE

5.2 - SCOPE_USER

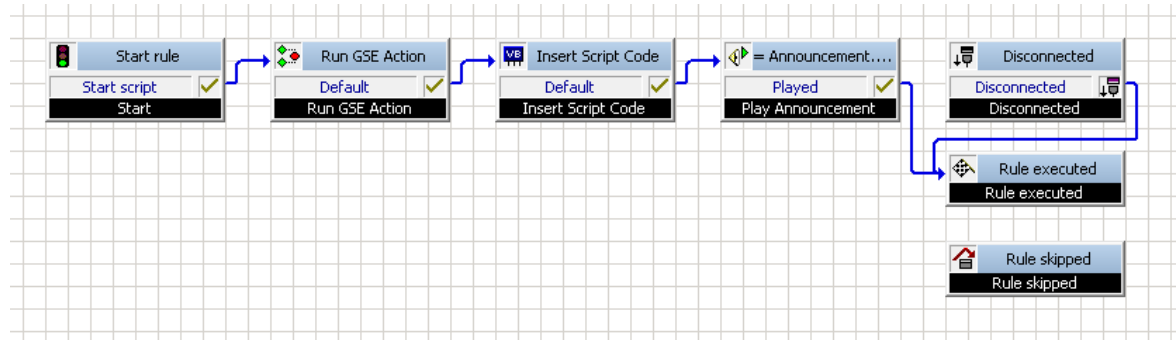
5.3 - SCOPE_GROUP

5.4 - SCOPE_GLOBAL

MISCELLANEOUS

6.1 - Database field definitions

6.2 - Master Standby Installations



Insert Script Code Properties

General Parameters Links

This code will be inserted into the generated script:
Note: All script code added here will be executed when the script reaches this block. Please do not add function definitions here.

```
' TODO: Insert you script code here

UseExit = 0 ' Please use the variable UseExit for the block

Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"
Announcement.Default = "Beep.wav"

PlaySound Announcement.Value
```

OK Cancel Help

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

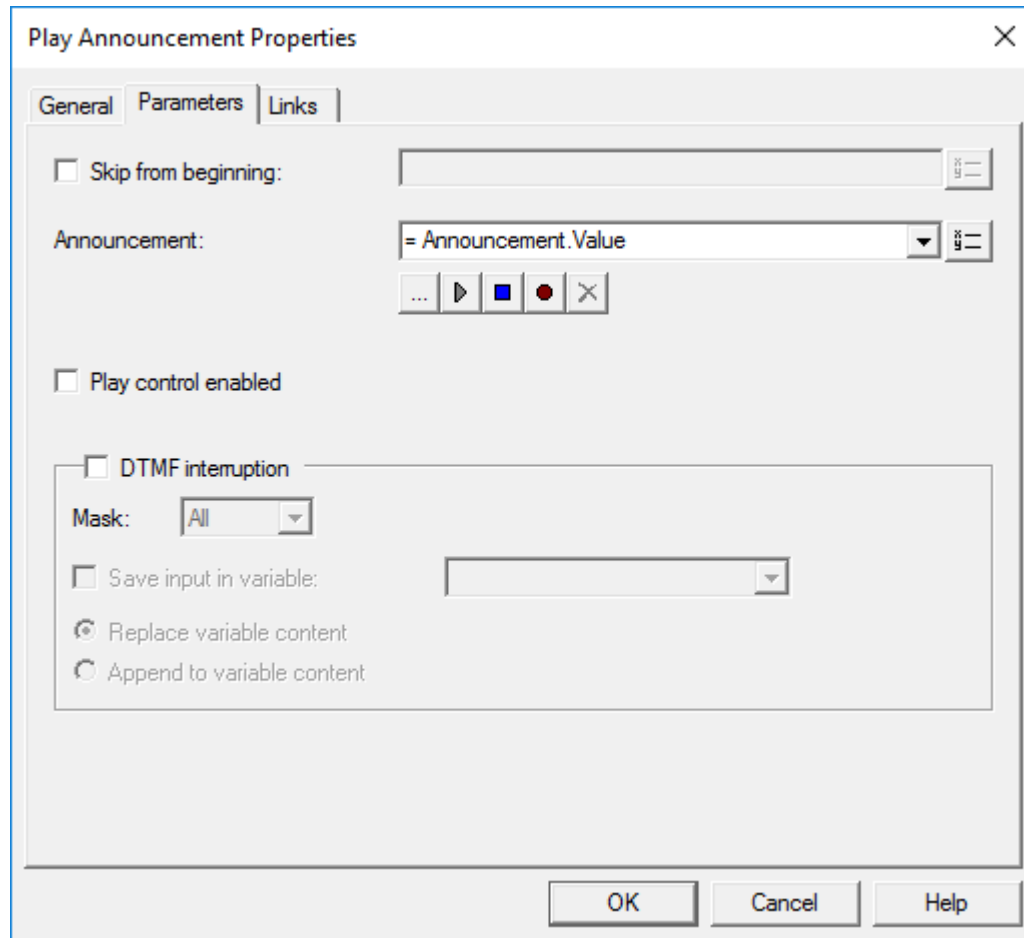
A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History



The script reads a persistent variable called **Welcome** and uses a **Play Announcement** block to announce the current value (which should be the name of a wav file). As nothing else was configured the value is taken from the user scope if the script runs for a user. If the script runs for a group the value will be taken from the group scope.

If you have called the previous example before the content will be **Default Welcome.wav**. Otherwise the configured default value **Beep.wav** will be returned.

To install this script generate a new GSE rule, via the menu **File | Import...** select the file **Retrieve.rse**, save the rule, close the GSE and make sure the rule is activated to give it a try.

Hint: the Persistent Variables can be used with **ANY** SwyxWare version. The included examples and the GSE Action itself however might be stored with the most recent SwyxWare version of the time the Persistent Variable version was released. This means that you might need to use the latest SwyxWare version to be able to open the included GSE Rules (.rse files) and the GSE Action (.ase file) within your GSE.



By Tom Wellige

May 14, 2022

 Share



Previous Page

3.4 - Simple Script - Store Variable

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 4.1 - Name

Followers

0

VBScript

This property defines the name of the persistent variable. The name is used to access/identify the persistent variable.

Depending on the used Scope (SCOPE_USER, SCOPE_GROUP or SCOPE_GLOBAL) the name is already sufficient to identify the variable.

If the used scope is SCOPE_NAMESPACE the Namespace property must be given as well.

The given name may contain any character. It has a maximum length of 50 characters. Longer names will be truncated.

```
Dim Announcement
Set Announcement = new PersistentVariable
```

4.1 - Name

4.2 - Value

4.3 - Default

4.4 - Scope

4.5 - Namespace

4.6 - UserID

4.7 - GroupID

4.8 - CallerName

4.9 - CallerNumber

4.10 - Created

4.11 - CreatedByName

4.12 - CreatedByNumber

4.13 - Modified

4.14 - ModifiedByName

4.15 - ModifiedByNumber

4.16 - Forget

4.17 - LatestError

4.18 - LatestDescription

SCOPES

5.1 - SCOPE_NAMESPACE

5.2 - SCOPE_USER

5.3 - SCOPE_GROUP

5.4 - SCOPE_GLOBAL

MISCELLANEOUS

6.1 - Database field definitions

6.2 - Master Standby Installations

```
Announcement.Name = "Welcome"
```


The above example defines a persistent variable with the name **Welcome** in the **user** scope of the current script user or in the **group** scope of the current script group.



By Tom Wellige

May 14, 2022

 Share

Next Page 
[4.2 - Value](#)

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 4.2 - Value

Followers

0

VBScript

This property represents the value (i.e. the content) of the persistent variable.

While accessing this property the content will be either automatically written into the database or retrieved from the database.

The value takes up to **1kB** (1024 Byte) of any data, like strings, booleans, numbers or dates. Data above the 1kB limit will be truncated.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"

Announcement.Value = "Default Welcome.wav"
```


4.1 - Name

4.2 - Value

4.3 - Default

4.4 - Scope

4.5 - Namespace

4.6 - UserID

4.7 - GroupID

4.8 - CallerName

4.9 - CallerNumber

4.10 - Created

4.11 - CreatedByName

4.12 - CreatedByNumber

4.13 - Modified

4.14 - ModifiedByName

4.15 - ModifiedByNumber

4.16 - Forget

4.17 - LatestError

4.18 - LatestDescription

SCOPES

5.1 - SCOPE_NAMESPACE

5.2 - SCOPE_USER

5.3 - SCOPE_GROUP

5.4 - SCOPE_GLOBAL

MISCELLANEOUS

6.1 - Database field definitions

6.2 - Master Standby Installations

```
PlaySound Announcement.Value
```

The above example defines a persistent variable with the name **Welcome** in the **user** scope of the current script user or in the **group** scope of the current script group.

The next line shows how to write into the persistent variable. The new content will be directly written into the database and is therefore persistently/permanently available.

The next line shows how to read from the persistent variable. The content will be directly retrieved from the database. Of course it is possible to use the content of the persistent variable directly in nearly any GSE block, e.g.:

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

The screenshot shows the 'Play Announcement Properties' dialog box with the 'General' tab selected. The 'Parameters' tab is also visible. The 'Skip from beginning' checkbox is unchecked. The 'Announcement' dropdown is set to '= Announcement.Value'. Below it are buttons for '...', 'Play', 'Stop', 'Pause', and 'Close'. The 'Play control enabled' checkbox is unchecked. The 'DTMF interruption' checkbox is unchecked. The 'Mask' dropdown is set to 'All'. Below it are checkboxes for 'Save input in variable:', 'Replace variable content' (selected), and 'Append to variable content'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Play Announcement Properties

General Parameters Links

☐ Skip from beginning:

Announcement: = Announcement.Value

☐ Play control enabled

☐ DTMF interruption

Mask: All

☐ Save input in variable:

☒ Replace variable content

☐ Append to variable content

OK Cancel Help



By Tom Wellige

May 14, 2022

[Share](#)

< Previous Page
4.1 - Name

Next Page >
4.3 - Default

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 4.3 - Default

Followers

0

VBScript

This property defines the default value of a persistent variable.

When reading the content of a persistent variable it is possible that the variable has not been set before. In this case the configured default value will be returned.

The default value takes up to **1kB** (1024 Byte) of any data, like strings, booleans, numbers or dates. Data above the 1kB limit will be truncated.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"
Announcement.Default = "Beep.wav"
```

4.1 - Name

4.2 - Value

4.3 - Default

4.4 - Scope

4.5 - Namespace

4.6 - UserID

4.7 - GroupID

4.8 - CallerName

4.9 - CallerNumber

4.10 - Created

4.11 - CreatedByName

4.12 - CreatedByNumber

4.13 - Modified

4.14 - ModifiedByName

4.15 - ModifiedByNumber

4.16 - Forget

4.17 - LatestError

4.18 - LatestDescription

SCOPES

5.1 - SCOPE_NAMESPACE

5.2 - SCOPE_USER

5.3 - SCOPE_GROUP

5.4 - SCOPE_GLOBAL

MISCELLANEOUS

6.1 - Database field definitions

6.2 - Master Standby Installations

```
PlaySound Announcement.Value
```

The above example defines a persistent variable with the name **Welcome** in the **user scope** of the current script user or in the **group** scope of the current script group.

Before accessing the content the default value **Beep.wav** is set.

If at script runtime no other script has set the content of the variable before the default value will be used, i.e. a beep will be played in this example. Otherwise the previously stored content will be played.

The default value will also be returned in case of any error accessing the database the persistent variables are stored in. If the Persistent Variables extension is properly installed no errors should happen, but you never know...

Optional error handling will be discussed in LatestError and LatestDescription.



By Tom Wellige

May 14, 2022

Share

< Previous Page
4.2 - Value

Next Page >
4.4 - Scope

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 4.4 - Scope

Followers

0

VBScript

This property sets the scope, i.e. the visibility, of a persistent variable. It takes a numerical value from 1 - 4. Any other values will be ignored.

If you omit this property the **default scope** is used. This is the **user scope** if you use the persistent variables within a call routing script of a user, or the group scope if you use the persistent variable within a call routing script of a group.

If you use the persistent variable outside of a call routing script (new in v1.1.0) the default scope is the **global scope**.

For convenience purpose there are four VBScript constants defined for the four available scopes: SCOPE_NAMESPACE (1), SCOPE_USER (2 - default in user call routings), SCOPE_GROUP (4 - default in group call routings) or SCOPE_GLOBAL (3).

4.1 - Name

4.2 - Value

4.3 - Default

4.4 - Scope

4.5 - Namespace

4.6 - UserID

4.7 - GroupID

4.8 - CallerName

4.9 - CallerNumber

4.10 - Created

4.11 - CreatedByName

4.12 - CreatedByNumber

4.13 - Modified

4.14 - ModifiedByName

4.15 - ModifiedByNumber

4.16 - Forget

4.17 - LatestError

4.18 - LatestDescription

SCOPES

5.1 - SCOPE_NAMESPACE

5.2 - SCOPE_USER

5.3 - SCOPE_GROUP

5.4 - SCOPE_GLOBAL

MISCELLANEOUS

6.1 - Database field definitions

6.2 - Master Standby Installations

```
Dim Announcement
```

```
Set Announcement = new PersistentVariable
```

```
Announcement.Name = "Welcome"
```

```
Announcement.Scope = SCOPE_GLOBAL
```

The above example set the scope of the persistent variable Welcome to global, meaning that this variable is visible and accessible from all scripts of all users and groups.



By Tom Wellige

May 14, 2022

Share

< Previous Page
4.3 - Default

Next Page >
4.5 - Namespace

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 4.5 - Namespace

Followers

0

VBScript

This property sets the name of a namespace if the Scope if the persistent variable is set to SCOPE_NAMESPACE.

The given name may contain any characters. It has a maximum length of 128 characters. Longer strings will be truncated.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"
Announcement.Scope = SCOPE_NAMESPACE
Announcement.Namespace = "Support"
```

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace**
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

The above example defines a persistent variable **Welcome** in the namespace **Support**. All scripts defining the same namespace (regardless on which user or group they are running) will be able to access this variable.

Another persistent variable **Welcome** in a namespace **Sales** would not interfere.



By Tom Wellige
May 14, 2022

 Share

< Previous Page
4.4 - Scope

Next Page >
4.6 - UserID

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 4.6 - UserID

Followers

0

VBScript

1.1.0

The usual way of working with persistent variables is most likely from within a call routing script. From v1.1.0 the persistent variables can also be used outside of a call routing script, i.e. in a standard windows scripting file (.wsf) or an asp web page.

In these cases the persistent variables switch their default Scope from User/Group to Global to reflect the fact, that they don't have access to a SwyxWare user id, and therefore can't use the User Scope completely by themselves.

If for what reason ever you need to access a persistent variable in the user scope of a certain user from outside a call routing script, you need to specify the SwyxWare user id belonging to that user and switch the scope to User:

4.1 - Name

4.2 - Value

4.3 - Default

4.4 - Scope

4.5 - Namespace

4.6 - UserID

4.7 - GroupID

4.8 - CallerName

4.9 - CallerNumber

4.10 - Created

4.11 - CreatedByName

4.12 - CreatedByNumber

4.13 - Modified

4.14 - ModifiedByName

4.15 - ModifiedByNumber

4.16 - Forget

4.17 - LatestError

4.18 - LatestDescription

SCOPES

5.1 - SCOPE_NAMESPACE

5.2 - SCOPE_USER

5.3 - SCOPE_GROUP

5.4 - SCOPE_GLOBAL

MISCELLANEOUS

6.1 - Database field definitions

6.2 - Master Standby Installations

```
<package>
<job id="set_variable_in_user_scope">
<script language="VBScript" src="PersistentVariables.vbs"/>
<script language="VBScript">
```

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"
Announcement.Scope = SCOPE_USER
Announcement.UserID = 15
```

```
Announcement.Value = "Beep.wav"
```

```
Set Announcement = Nothing
```

```
WScript.Quit
```

```
</script>
</job>
</package>
```

If you save the above code into a text file and name it **SetAnnouncement.wsf** you can call it directly from the **command prompt**: (the **PersistentVariables.vbs** file needs to be in the same folder!)

```
C:\PersistentVariables> SetAccouncement
```

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

The above example demonstrates the usage of persistent variables in a standard windows scripting host file (.wsf).

There are many way to figure the user id of a SwyxWare user:

- Assuming you are accessing the variable also from inside a call routing script you can simply take a look into the [PersistentVariables](#) table and take the user id from there.
- You can also [trace it](#) into the server trace file from within a call routing and read it from there.
- Use any of the SwyxWare APIs ([Server Script API](#), [Client SDK](#), [ConfigDataStore SDK](#), [Powershell](#)) to obtain it from SwyxWare.
- ...

Please find a complete explanation of the usage of persistent variables outside of call routing scripts here:

- [6.6 - Usage of Persistent Variables outside of a call routing script](#)



By Tom Wellige

May 14, 2022

Share



Previous Page
[4.5 - Namespace](#)

Next Page
[4.7 - GroupID](#)

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

Persistent Variables - 4.7 - GroupID

Followers

0

VBScript

1.4.0

The usual way of working with persistent variables is most likely from within a call routing script. From v1.1.0 the persistent variables can also be used outside of a call routing script, i.e. in a standard windows scripting file (.wsf) or an asp web page.

In these cases the persistent variables switch their default Scope from User/Group to Global to reflect the fact, that they don't have access to a SwyxWare user id or group id, and therefore can't use the User or Group Scope automatically.

If for what reason ever you need to access a persistent variable in the group scope of a certain group from outside a call routing script, you need to specify the SwyxWare group id belonging to that group and switch the scope to Group:

4.1 - Name

4.2 - Value

4.3 - Default

4.4 - Scope

4.5 - Namespace

4.6 - UserID

4.7 - GroupID

4.8 - CallerName

4.9 - CallerNumber

4.10 - Created

4.11 - CreatedByName

4.12 - CreatedByNumber

4.13 - Modified

4.14 - ModifiedByName

4.15 - ModifiedByNumber

4.16 - Forget

4.17 - LatestError

4.18 - LatestDescription

SCOPES

5.1 - SCOPE_NAMESPACE

5.2 - SCOPE_USER

5.3 - SCOPE_GROUP

5.4 - SCOPE_GLOBAL

MISCELLANEOUS

6.1 - Database field definitions

6.2 - Master Standby Installations

```
<package>
<job id="set_variable_in_user_scope">
<script language="VBScript" src="PersistentVariables.vbs"/>
<script language="VBScript">
```

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"
Announcement.Scope = SCOPE_GROUP
Announcement.UserID = 3
```

```
Announcement.Value = "Beep.wav"
```

```
Set Announcement = Nothing
```

```
WScript.Quit
```

```
</script>
</job>
</package>
```

If you save the above code into a text file and name it **SetAnnouncement.wsf** you can call it directly from the **command prompt**: (the **PersistentVariables.vbs** file needs to be in the same folder!)

```
C:\PersistentVariables> SetAccouncement
```

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

The above example demonstrates the usage of persistent variables in a standard windows scripting host file (.wsf).

There are many way to figure the group id of a SwyxWare group:

- Assuming you are accessing the variable also from inside a call routing script you can simply take a look into the [PersistentVariables](#) table and take the group id from there.
- You can also [trace it](#) into the server trace file from within a call routing and read it from there.
- Use any of the SwyxWare APIs ([Server Script API](#), [Client SDK](#), [ConfigDataStore SDK](#), [Powershell](#)) to obtain it from SwyxWare.
- ...

Please find a complete explanation of the usage of persistent variables outside of call routing scripts here:

- [6.6 - Usage of Persistent Variables outside of a call routing script](#)



By Tom Wellige

February 29, 2024

Share



Previous Page
4.6 - UserID

Next Page

4.8 - CallerName



Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

Persistent Variables - 4.8 - CallerName

Followers

0

VBScript

1.4.0

With this property you can set the **name** which is written into the CreatedByName and ModifiedByName properties when a variable is created the first time or modified.

Usualy you don't need to set this property manually as the persistent variable will use the name of the current caller automatically. If however you need to overwrite that name or use the persistent variable outside of a call routing script you can use this property to do so,

You must set this property **before** setting the value property in order to apply the new name.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"
Announcement.CallerName = "Erika Mustermann"
```

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName**
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

```
Announcement.CallerNumber = "+4930123457890"
```

```
Announcement.Value = "Beep.wav"
```

The maximum length of the name you can set is **80** characters. Any additional characters will be truncated.

With the CallerNumber property you can also modify the number which is written into the CreatedByNumber and ModifiedByNumber properties when a variable is created the first time or modified.



By Tom Wellige
February 29, 2024

Share

< Previous Page
4.7 - GroupID

Next Page >
4.9 - CallerNumber

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

Persistent Variables - 4.9 - CallerNumber

Followers

0

VBScript

1.4.0

With this property you can set the **number** which is written into the CreatedByNumber and ModifiedByNumber properties when a variable is created the first time or modified.

Usualy you don't need to set this property manually as the persistent variable will use the number of the current caller automatically. If however you need to overwrite that number or use the persistent variable outside of a call routing script you can use this property to do so,

You must set this property **before** setting the value property in order to apply the new number.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"
```


- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber**
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

```
Announcement.CallerName  = "Erika Mustermann"  
Announcement.CallerNumber = "+4930123457890"  
  
Announcement.Value       = "Beep.wav"
```

The maximum length of the number you can set is **80** characters. Any additional characters will be truncated.

With the CallerName property you can also modify the number which is written into the CreatedByName and ModifiedByName properties when a variable is created the first time or modified.



By Tom Wellige
February 29, 2024

Share

< Previous Page
4.8 - CallerName

Next Page >
4.10 - Created

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

Persistent Variables - 4.10 - Created

Followers

0

VBScript

1.3.0

This property contains the date/time when the persistent variable was stored first time.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"

If Not IsNull(Announcement.Created) Then
    PBXScript.OutputTrace "Created: " & FormatDateTime(Announcement.Created)
    PBXScript.OutputTrace "CreatedByName: " & Announcement.CreatedByName
    PBXScript.OutputTrace "CreatedByNumber: " & Announcement.CreatedByNumber
End If
```

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created**
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

This property returns a **datetime** value.

This property allows read access only.

If a variable wasn't set before you access this property, a **NULL** value will be returned. This is also the case for variables which were originally set with a persistent variable version less than 1.3.0.

You should check for NULL (as in above code snippet) to prevent problems in your code, as not every function is capable of dealing with NULL properly.



By Tom Wellige
February 25, 2024

Share

< Previous Page
4.9 - CallerNumber

Next Page >
4.11 - CreatedByName

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

Persistent Variables - 4.11 - CreatedByName

Followers

0

VBScript

1.4.0

This property contains the name of the caller when the persistent variable was stored first time.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"

If Not IsNull(Announcement.Created) Then
    PBXScript.OutputTrace "Created:           " & FormatDateTime(Announcement.Created)
    PBXScript.OutputTrace "CreatedByName:      " & Announcement.CreatedByName
```

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName**
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

```
PBXScript.OutputTrace "CreatedByNumber: " & Announcement.CreatedByNumber  
End If
```

This property returns a **string** value.

This property allows read access only.

If a variable wasn't set before you access this property, an **empty string** will be returned.

You can use the CallerName property before writing the first time into this variable to define an own name to be written into CreatedByName, instead of the name of the current caller. ▶



By Tom Wellige
February 29, 2024

Share

< Previous Page
4.10 - Created

Next Page >
4.12 - CreatedByNumber

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

Persistent Variables - 4.12 - CreatedByNumber

Followers

0

VBScript

1.4.0

This property contains the number of the caller when the persistent variable was stored first time.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"

If Not IsNull(Announcement.Created) Then
    PBXScript.OutputTrace "Created:           " & FormatDateTime(Announcement.Created)
    PBXScript.OutputTrace "CreatedByName:    " & Announcement.CreatedByName
```

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber**
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

```
PBXScript.OutputTrace "CreatedByNumber: " & Announcement.CreatedByNumber  
End If
```

This property returns a **string** value.

This property allows read access only.

If a variable wasn't set before you access this property, an **empty string** will be returned.

You can use the CallerNumber property before writing the first time into this variable to define an own number to be written into CreatedByNumber, instead of the number of the current caller.



By Tom Wellige
February 29, 2024

Share



Previous Page
4.11 - CreatedByName

Next Page
4.13 - Modified

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2.1 - Setup Database

2.2 - Setup Database Table

2.3 - Setup GSE Action

2.4 - Setup an updated version

USAGE

3.1 - Create a persistent variable

3.2 - Store a value

3.3 - Retrieve a value

3.4 - Simple Script - Store Variable

3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

Persistent Variables - 4.13 - Modified

Followers

0

VBScript

1.3.0

This property contains the date/time when the persistent variable was modified the last time.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"

If Not IsNull(Announcement.Modified) Then
    PBXScript.OutputTrace "Modified: " & FormatDateTime(Announcement.Modified)
    PBXScript.OutputTrace "ModifiedByName: " & Announcement.ModifiedByName
    PBXScript.OutputTrace "ModifiedByNumber: " & Announcement.ModifiedByNumber
End If
```

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified**
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

This property returns a **datetime** value.

This property allows read access only.

If a variable wasn't modified before you access this property, a **NULL** value will be returned. This is also the case for variables which were originally set with a persistent variable version less than 1.3.0.

You should check for NULL (as in above code snippet) to prevent problems in your code, as not every function is capable of dealing with NULL properly.



By Tom Wellige
February 25, 2024

Share



Previous Page
4.12 - CreatedByNumber

Next Page

4.14 - ModifiedByName



6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

Persistent Variables - 4.14 - ModifiedByName

Followers

0

VBScript

1.4.0

This property contains the name of the caller when the persistent variable was stored/modified the last time.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"

If Not IsNull(Announcement.Modified) Then
    PBXScript.OutputTrace "Modified:           " & FormatDateTime(Announcement.Modified)
    PBXScript.OutputTrace "ModifiedByName:      " & Announcement.ModifiedByName
```

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName**
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

```
PBXScript.OutputTrace "ModifiedByNumber: " & Announcement.ModifiedByNum  
End If
```

This property returns a **string** value.

This property allows read access only.

If a variable wasn't set before you access this property, an **empty string** will be returned.

You can use the CallerName property before updating this variable to define an own name to be written into ModifiedByName, instead of the name of the current caller.



By Tom Wellige
February 29, 2024

Share

< Previous Page
4.13 - Modified

Next Page >
4.15 - ModifiedByNumber

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

Persistent Variables - 4.15 - ModifiedByNumber

Followers

0

VBScript

1.4.0

This property contains the number of the caller when the persistent variable was stored/modified the last time.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"

If Not IsNull(Announcement.Modified) Then
    PBXScript.OutputTrace "Modified:           " & FormatDateTime(Announcement.Modified)
    PBXScript.OutputTrace "ModifiedByName:      " & Announcement.ModifiedByNumber
```

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber**
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

```
PBXScript.OutputTrace "ModifiedByNumber: " & Announcement.ModifiedByNum  
End If
```

This property returns a **string** value.

This property allows read access only.

If a variable wasn't set before you access this property, an **empty string** will be returned.

You can use the CallerNumber property before updating this variable to define an own number to be written into ModifiedByNumber, instead of the number of the current caller. ▶



By Tom Wellige
February 29, 2024

Share

< Previous Page
4.14 - ModifiedByName

Next Page >
4.16 - Forget

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

Persistent Variables - 4.16 - Forget

Followers

0

VBScript

1.3.0

This method removes a persistent variable from the database, i.e. its content is getting forgotten.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"
Announcement.Forget
```

The above example removes the persistent variable from the user or group scope of the current call routing user from the database.

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget**
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

Usually it shouldn't be needed to use this method at all. If for what reason ever you really need to remove a persistent variable from the database, this method will do it for you.

Of course it is also possible to access the persistent variables database (i.e. with the [SQL Server Management Studio](#)) and delete variables directly from there,



By Tom Wellige
February 25, 2024

Share

< Previous Page
4.15 - ModifiedByNumber

Next Page >
4.17 - LatestError

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 4.17 - LatestError

Followers

0

VBScript

This property returns a **Windows Error Code** (numerical) of the last read/write access of the persistent variable. The **Windows Error Text** can be found in [LatestDescription](#).

A list of all possible error codes and their meaning can be found in the MSDN (Microsoft Developer Network):

- [System Error Codes](#)

This property is read only.

As persistent variables are stored into a database there is a possibility of runtime errors. To check for any such runtime errors during read or write access of a persistent variable you can check this property for not equal zero.

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError**
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

If you have installed the Persistent Variable extension as recommended the chances for runtime errors are very low. Therefore it shouldn't be necessary to add error handling.

If you have installed the extension lets say into a database server on another machine it is highly recommended to add proper error handling.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"

Announcement.Value = "Welcome Default.wav"

if Announcement.LatestError <> 0 then
    ' add your error handling here
    ' error code in LatestError
    ' error descr in LatestDescription
end if
```

The above example gives a template for own error handling.



By Tom Wellige
May 14, 2022

Share

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

< Previous Page
4.16 - Forget

Next Page >
4.18 - LatestDescription

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 4.18 - LatestDescription

Followers

0

VBScript

This property returns a **Windows Error Text** (string) of the last read/write access of the persistent variable. The **Windows Error Code** (numerical) can be found in [LatestError](#).

This property is read only.

As persistent variables are stored into a database there is a possibility of runtime errors. To check for any such runtime errors during read or write access of a persistent variable you can check [LatestError](#) for not equal zero.

If you have installed the Persistent Variable extension as recommended the chances for runtime errors are very low. Therefore it shouldn't be necessary to add error handling.

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

If you have installed the extension lets say into a database server on another machine it is highly recommended to add proper error handling.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"

Announcement.Value = "Welcome Default.wav"

if Announcement.LatestError <> 0 then
    ' add your error handling here
    ' error code in LatestError
    ' error descr in LatestDescription
end if
```

The above example gives a template for own error handling.



By Tom Wellige
May 14, 2022

[Share](#)



Previous Page
[4.17 - LatestError](#)

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

2.1 - Setup Database

2.2 - Setup Database Table

2.3 - Setup GSE Action

2.4 - Setup an updated version

USAGE

3.1 - Create a persistent variable

3.2 - Store a value

3.3 - Retrieve a value

3.4 - Simple Script - Store Variable

3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 5.1 - SCOPE_NAMESPACE

Followers

0

VBScript

This is a VBScript constant defined to the value of 1. This constant will be used to set the Scope of a persistent variable to the namespace scope.

Variables in the namespace scope are visible/accessible for all user's scripts defining the same Namespace name.

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"
```

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

5.1 - SCOPE_NAMESPACE

- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

```
Announcement.Scope      = SCOPE_NAMESPACE
Announcement.Namespace = "Support"
```

The above example defines a persistent variable **Welcome** in the namespace **Support**. All scripts defining the same namespace will be able to access this variable.

Another persistent variable **Welcome** in a namespace **Sales** will not interfere.



By Tom Wellige
May 14, 2022

Share

Next Page >
5.2 - SCOPE_USER

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 5.2 - SCOPE_USER

Followers

0

VBScript

This is a VBScript constant defined to the value of 2. This constant will be used to set the Scope of a persistent variable to the user scope.

Variables with a scope set to SCOPE_USER are visible/accessible for all scripts of the current script user.

This is the **default scope** of every new persistent variable if being used within a call routing for a **user**. If the variable is used in a call routing for a group the default is SCOPE_GROUP. If being used outside of a call routing script (new in v1.1.0) the default is SCOPE_GLOBAL.

```
Dim Announcement
Set Announcement = new PersistentVariable
```

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

```
Announcement.Name = "Welcome"  
Announcement.Scope = SCOPE_USER
```

If you use the persistent variable in a user call routing, the default scope is SCOPE_USER, so you can shorten your code to:

```
Dim Announcement  
Set Announcement = new PersistentVariable  
Announcement.Name = "Welcome"
```

The above examples define a persistent variable **Welcome** in the User scope. This variable is visible/accessible for all scripts of the same user.

Another script of another user which defines also a persistent variable **Welcome** does not interfere.



By Tom Wellige
May 14, 2022

[Share](#)

[< Previous Page](#)
[5.1 - SCOPE_NAMESPACE](#)

[Next Page >](#)
[5.3 - SCOPE_GROUP](#)

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

Persistent Variables - 5.3 - SCOPE_GROUP

Followers

0

VBScript

1.4.0

This is a VBScript constant defined to the value of 4. This constant will be used to set the Scope of a persistent variable to the group scope.

Variables with a scope set to SCOPE_GROUP are visible/accessible for all scripts of the current script group.

This is the **default scope** of every new persistent variable if being used within a call routing for a **group**. If the variable is used in a call routing for a user the default is SCOPE_USER. If being used outside of a call routing script (new in v1.1.0) the default is SCOPE_GLOBAL.

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"
Announcement.Scope = SCOPE_GROUP
```

If you use the persistent variable in a group call routing, the default scope is SCOPE_GROUP, so you can shorten your code to:

```
Dim Announcement
Set Announcement = new PersistentVariable
Announcement.Name = "Welcome"
```

The above examples define a persistent variable **Welcome** in the Group scope. This variable is visible/accessible for all scripts of the same group.

Another script of another group which defines also a persistent variable **Welcome** does not interfere.



By Tom Wellige

February 29, 2024

[Share](#)

[< Previous Page](#)
[5.2 - SCOPE_USER](#)

[Next Page >](#)
[5.4 - SCOPE_GLOBAL](#)

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 5.4 - SCOPE_GLOBAL

Followers

0

VBScript

This is a VBScript constant defined to the value of 3. This constant will be used to set the Scope of a persistent variable to the global scope.

Variables in the global scope are visible/accessable for all scripts of all users and groups.

This is the **default scope** of every new persistent variable if being used outside a call routing script (new in v1.1.0). If being used inside of a user call routing SCOPE_USER is the default scope. If being used inside of a group call routing SCOPE_GROUP is the default.

```
Dim Announcement
Set Announcement = new PersistentVariable
```

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

```
Announcement.Name = "Welcome"
Announcement.Scope = SCOPE_GLOBAL
```

The above example defines a persistent variable in the global scope. This variable is visible and accessible for all scripts of all users and groups.



By Tom Wellige
May 14, 2022

 Share



Previous Page

[5.3 - SCOPE_GROUP](#)

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```

1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
15

```

Persistent Variables - 6.1 - Database field definitions

Followers

0

VBScript

This is how the PersistentVariables table with it's fields is defined:

PersistentVariables

PersVarID	int	unique id of variable
Name	nvarchar(50)	variable name
Value	nvarchar(1024) *	value of variable
Scope	int	scope of variable (SCOPE_SCRIPT)
UserID	int *	SwyxWare UserID of user who is
GroupID	int *	SwyxWare GroupID of group which
Namespace	nvarchar(128) *	Namespace in which this variable

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

Created	datetime *	Timestamp of when the variable
CreatedByName	nvarchar(80)*	Name of caller in call routing
CreatedByNumber	nvarchar(80)*	Number of caller in call routing
Modified	datetime *	Timestamp of when the variable
ModifiedByName	nvarchar(80)*	Name of caller in call routing
ModifiedByNumber	nvarchar(80)*	Number of caller in call routing

* = Allow Nulls



By Tom Wellige
May 14, 2022

[Share](#)

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 6.2 - Master Standby Installations

Followers

0

VBScript

By default persistent variables are not compatible to Master Standby scenarios.

This is simply to the fact that the **IpPbxExtensions** database is not replicated automatically into the MS SQL server of the Standby machine.

To use persistent variables in a Master Standby scenario you have two possibilities:

- Setup the **IpPbxExtensions** database along with the **PersistentVariables** table also on the Standby system and configure a **replication** between the both database. This has to be done manually as SwyxWare's config wizard does not know about the

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

IpPbxExtensions database.

- Do not setup a replication, ignore possible switches from Master to Standby and just rely on proper Default values for the persistent variables. This is most definately not recommended.



By Tom Wellige
May 14, 2022

Share

< Previous Page
6.1 - Database field definitions

Next Page >
6.3 - Use SQL Server on different machine

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 6.3 - Use SQL Server on different machine

Followers

0

VBScript

When setting up the Persistent Variables extension you are free to setup the **IpPbxExtensions** database into an SQL server on another machine than the one SwyxWare is installed on.

The Persistent Variables extension is able to connect to the local SQL Server automatically.

If you create the database into an SQL Server on another machine you need to tell the Persistent Variables extension where to look for the database.

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

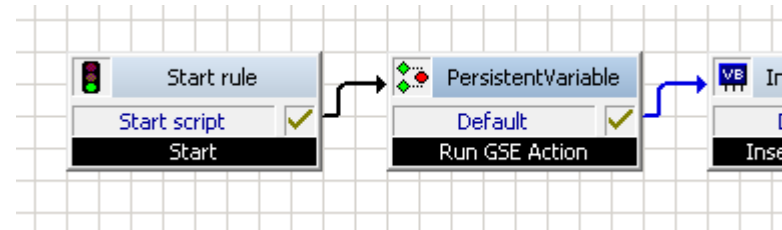
- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

This is done in the properties of the **Run GSE Action** block that loads the PersistentVariables GSE action.

Please note: it is highly recommended to use the suggested defaults for the database (local MS SQL Server, IpPbxExtensions database)!



6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

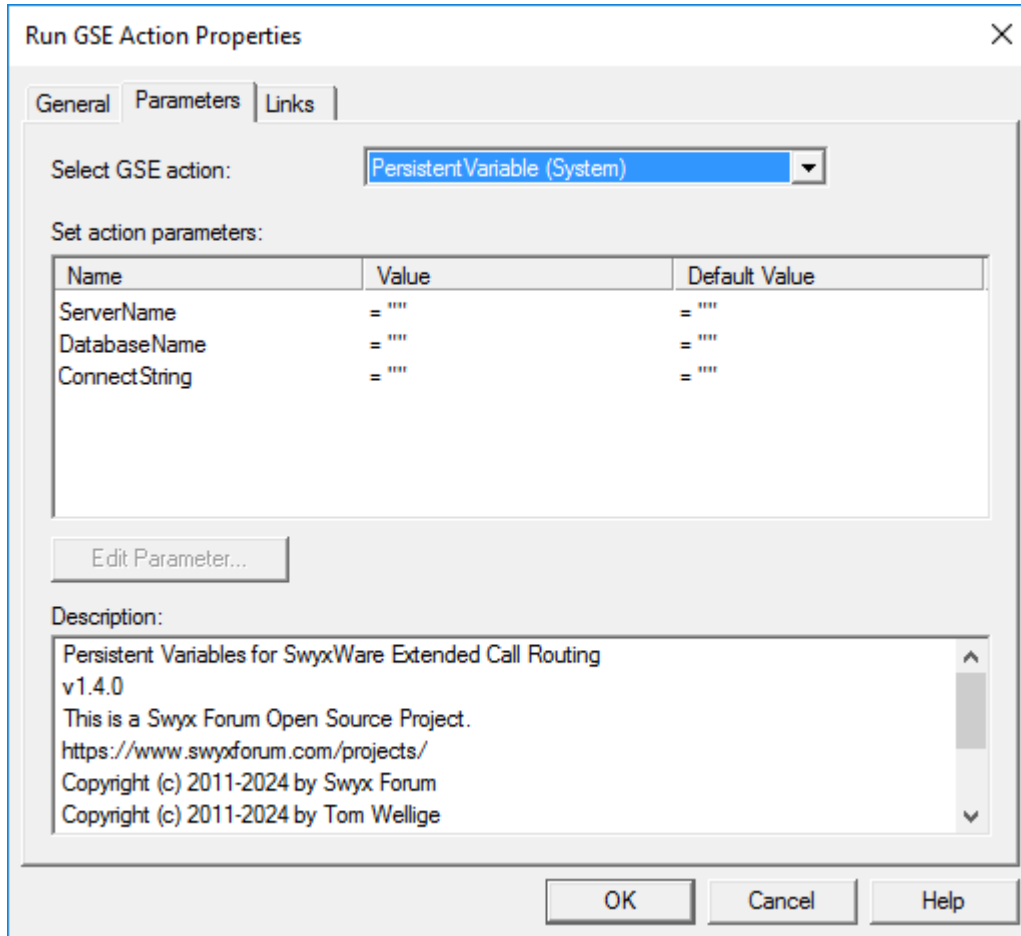
A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History



Run GSE Action Properties

General Parameters Links

Select GSE action: PersistentVariable (System)

Set action parameters:

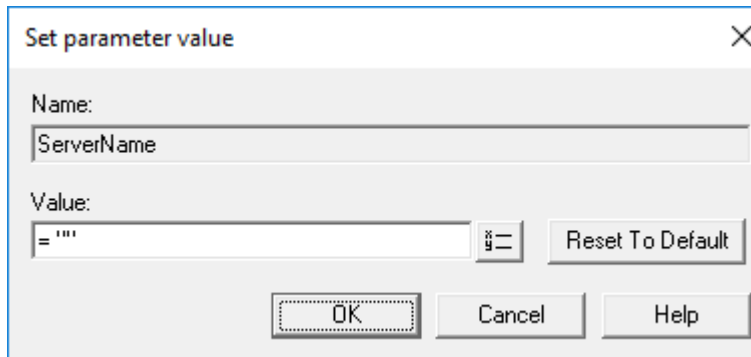
Name	Value	Default Value
ServerName	= ''	= ''
DatabaseName	= ''	= ''
ConnectionString	= ''	= ''

Edit Parameter...

Description:

Persistent Variables for SwyxWare Extended Call Routing
v1.4.0
This is a Swyx Forum Open Source Project.
<https://www.swyxforum.com/projects/>
Copyright (c) 2011-2024 by Swyx Forum
Copyright (c) 2011-2024 by Tom Wellige

OK Cancel Help



Set parameter value

Name:
ServerName

Value:
= ''

Reset To Default

OK Cancel Help

In order to create the database for the Persistent Variable extension into another SQL Server you have to configure the SwyxWare services to run under a domain user instead of the default local user. By doing this you will be able to grant access for this domain user also to the **IpPbxExtensions** on the other machine.

Alternatively you can also use the [6.5 - Use different database](#) configuration.



By Tom Wellige

May 14, 2022

Share

< Previous Page
6.2 - Master Standby Installations

Next Page >
6.4 - Use database with different name (other than ...)

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 Dim NightSwitch
2
3 Set NightSwitch = new PersistentVariable
4 NightSwitch.Name = "NightSwitch"
5
6 ' the NightSwitch can be accessed by any user
7 NightSwitch.Scope = SCOPE_GLOBAL
8
9 ' by default the NightSwitch is off ( 0 - off / 1 - on )
10 NightSwitch.Default = 0
11
12 ' pass status of NightSwitch directly on block exists
13 UseExit = CInt(NightSwitch.Value)
14
```

Persistent Variables - 6.4 - Use database with different name (other than default name)

Followers

0

VBScript

When setting up the Persistent Variables extension you are free to create/use into another database than the recommended **IpPbxExtensions** database.

The Persistent Variables extension uses the **IpPbxExtensions** database by default.

If you use another database you need to tell the Persistent Variables extension the name of it.

This is done in the properties of the **Run GSE Action** block that loads the PersistentVariables GSE action.

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

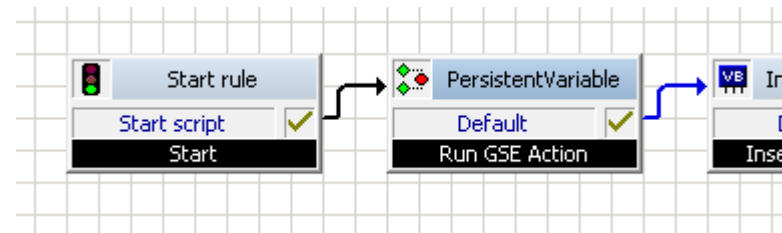
SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

Please note: it is highly recommended to use the suggested defaults for the database (local MS SQL Server, IpPbxExtensions database)!



Run GSE Action Properties

General Parameters Links

Select GSE action: PersistentVariable (System)

Set action parameters:

Name	Value	Default Value
ServerName	= ""	= ""
DatabaseName	= ""	= ""
ConnectString	= ""	= ""

Edit Parameter...

Description:

Persistent Variables for SwyxWare Extended Call Routing
v1.4.0
This is a Swyx Forum Open Source Project.
<https://www.swyxforum.com/projects/>
Copyright (c) 2011-2024 by Swyx Forum
Copyright (c) 2011-2024 by Tom Wellige

OK Cancel Help

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

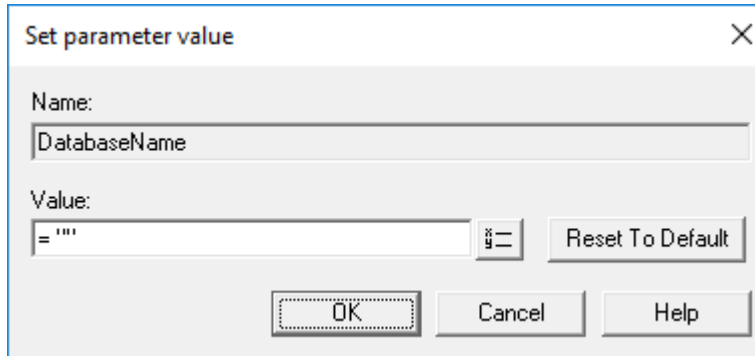
A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple ASP webpage

B.2 - Usage outside call routing - Simple VBS/WSF script

B.3 - Version History



The screenshot shows a 'Set parameter value' dialog box. It has a title bar with a close button. Inside, there are two input fields: 'Name' and 'Value'. The 'Name' field contains the text 'DatabaseName'. The 'Value' field contains the text ' = '''. To the right of the 'Value' field is a button labeled 'Reset To Default'. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Alternatively you can also use the [6.5 - Use different database](#) configuration.



By Tom Wellige
May 14, 2022

[Share](#)

[Previous Page](#)
6.3 - Use SQL Server on different machine

[Next Page](#)
6.5 - Use different database (other than MS SQL Se... [>](#)

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

4.1 - Name

```
1
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch of off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 6.5 - Use different database (other than MS SQL Server, e.g. MySQL or Oracle)

Followers

0

VBScript

When setting up the Persistent Variables extension you are free to use any type of database on any machine in your network.

If you do so you need to give the Persistent Variable extension a clue of it by giving the complete Connect String.

This is done in the properties of the **Run GSE Action** block that loads the PersistentVariables GSE action.

Please note: it is highly recommended to use the suggested defaults for the database (local MS SQL Server, IpPbxExtensions database)!

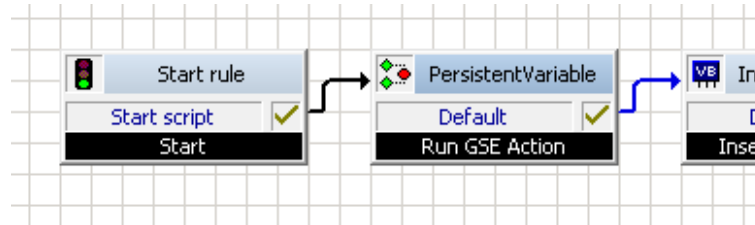
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations
- 6.3 - Use SQL Server on different machine



Run GSE Action Properties

General Parameters Links

Select GSE action: PersistentVariable (System)

Set action parameters:

Name	Value	Default Value
ServerName	= ""	= ""
DatabaseName	= ""	= ""
ConnectionString	= ""	= ""

Edit Parameter...

Description:

Persistent Variables for SwyxWare Extended Call Routing
v1.4.0
This is a Swyx Forum Open Source Project.
<https://www.swyxforum.com/projects/>
Copyright (c) 2011-2024 by Swyx Forum
Copyright (c) 2011-2024 by Tom Wellige

OK Cancel Help

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

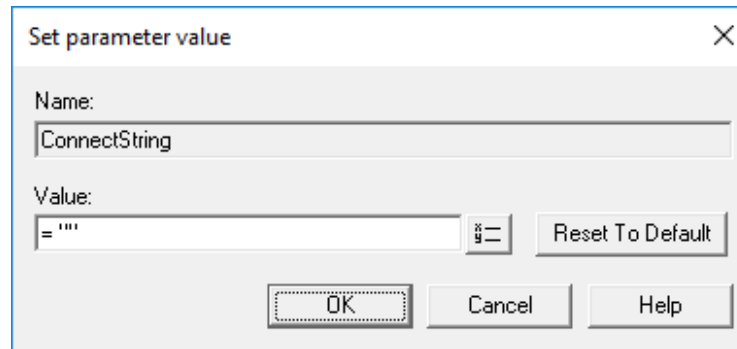
A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History



The screenshot shows a 'Set parameter value' dialog box. It has a title bar with a close button. Inside, there are two text input fields: 'Name' and 'Value'. The 'Name' field contains the text 'ConnectionString'. The 'Value' field contains the text '='. To the right of the 'Value' field is a 'Reset To Default' button. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Samples for a connection strings for the different available type of database can be found here:

- [ConnectionStrings.com](https://www.connectionstrings.com)



By Tom Wellige
May 14, 2022

[Share](#)

< Previous Page
6.4 - Use database with different name (other than ...

Next Page >
6.6 - Usage of Persistent Variables outside of a call ...

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - 6.6 - Usage of Persistent Variables outside of a call routing script

Followers

0

VBScript

1.1.0

The persistent variables have been developed to be used inside of call routing scripts. But as the underlying technology is plain VBScript they can also be used everywhere else where you can use VBScript. Please note that for this to work some adaptations needed to be made in the initial releases.

The idea behind is of course to have the same comfort in using persistent variables somewhere else, like e.g. in web pages or standard VBScripts.

When using the persistent variables outside of a call routing script there is however a difference to the normal usage within call routing scripts: the default Scope is not User but

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

Global. This is because from outside the call routing there is no SwyxWare user easily available and therefore no User scope possible by default. So without defining the scope explicitly, it is **global by default**.

```
Dim NightSwitch
Set NightSwitch = new PersistentVariable
NightSwitch.Name = "NightSwitch"
```

If you need to access a variable in the user scope of a specific SwyxWare user all you need to know is it's SwyxWare user id. There is a **UserID** property that needs to be set in that case.

```
Dim RedirectTo
Set RedirectTo = new PersistentVariable
RedirectTo.Name = "RedirectTo"
RedirectTo.Scope = SCOPE_USER
RedirectTo.UserID = 15
```

If you need to access a variable in the group scope of a specific SwyxWare user group all you need to know is it's SwyxWare group id. There is a **GroupID** property that needs to be set in that case.

```
Dim RedirectTo
Set RedirectTo = new PersistentVariable
RedirectTo.Name = "RedirectTo"
RedirectTo.Scope = SCOPE_GROUP
RedirectTo.UserID = 3
```

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

There are many ways to figure the id of a SwyxWare user or group:

- Assuming you are accessing the variable also from inside a call routing script you can simply take a look into the PersistentVariables table and take the id from there.
- You can also trace it into the server trace file from within a call routing and read it from there.
- Use any of the SwyxWare APIs (Server Script API, Client SDK, ConfigDataStore SDK, Powershell) to obtain it from SwyxWare.
- and more...

In order to use the persistent variables in other VBScript file, they need to get **included**. In a call routing script this is done by adding the PersistentVariable GSE action into the GSE rule. Standard VBScript doesn't know an include statement, but fortunately there are two environments like a call routing script, that add this missing functionality: Active Server Pages (ASP) and Windows Scripting Files (WSF).

Active Server Pages (.asp)

These are simply web pages hosted on an MS IIS webserver. The internet is full of documentation on ASP pages, so here it is just shown how to include persistent variables into an own asp page.

```
<!-- #include file = "PersistentVariables.inc"-->
<%
    ' vbscript code follows here
    ' ...
%>
```

Please note, that it is not the **PersistentVariables.vbs** file that is being included, but the **PersistentVariables.inc** file. This is because for asp pages <% and %> needs to be placed around VBScript code.

You can find this file in the **\examples\External\asp** folder of the download package. There you will also find a [complete example](#).

Windows Scripting Files (.wsf)

These are VBScripts that can be directly called by their filename, not even the .wsf file extension is needed when calling them.

```
<package>
<job id="Example">
<script language="VBScript" src="PersistentVariables.vbs"/>
<script language="VBScript">

    ' vbscript code follows here
    ' ...

    WScript.Quit
</script>
</job>
</package>
```

You can find the **PersistentVariables.vbs** file either in the **\vbs** as also in the **\examples\External\vbs** folders of the download package. There you will also find a [complete example](#).

The next step before being able to use any persistent variable is to configure them. While there is no configuration needed if you have installed everything according the [suggestion](#), here it is now necessary to give one important parameter: the **database connect string**. This is stored into the global variable **g_sPersistentVariableConnectionString** being defined by the persistent variables.

```
' configure the complete db connect string
g_sPersistentVariableConnectionString = _
    "Provider=sqloledb;" & _
    "Data Source=SQL_SERVER_NAME;" & _
    "Initial Catalog=" & PERSISTENT_VARIABLE_DATABASE & ";" & _
    "User Id=PersistentVariables;" & _
    "Password=PersistentVariables"
```

The **Data Source** is the sql server machine name, i.e. the SwyxServer machine name.

The **Initial Catalog** is the name of the database. If you installed everything according the suggestion this is **IpPbxExtensions**. This name is defined in the constant **PERSISTENT_VARIABLE_DATABASE**.

You most likely can't use integrated security to login to the database, so you need to configure a login with username and password (see [2.1 - Setup Database](#), **Step 5** onward, instead of of **Windows authentication** you have to use **SQL Server authentication**). The chosen username and password need to be given as **User Id** and **Password**.

From now on everything is just like the usage of persistent variables in a call routing script, i.e. like the two first code snippets on this page.



By Tom Wellige

May 14, 2022

 Share



Previous Page

[6.5 - Use different database \(other than MS SQL Se...](#)

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - A.1 - Night Switch - Night Switch Manager

Followers

0

VBScript

The **Night Switch Manager** script is used to enable or disable the night switch.

How to install it

- Make sure you have the Persistent Variables extension properly installed
- Create or use a **dummy/test** user
- Open it's **Call Routing Manager**
- Create a **new rule** using the GSE
- **Import** the file **NightSwitchManager.rse** from the download package

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

- You might want to record your **own announcement** for the **Get DTMF Char** block explaining the usage of the menu (press 0 to switch the night switch off, press 1 to switch it on).
- **Save** the rule and exit the GSE

How to use it

There are two ways to use the Night Switch Manager:

1. Simply call the dummy user. You will hear the current status of the night switch (0 for off and 1 for on). Afterwards you are in the DTMF menu to enable or disable the night switch. If you have recorded your own announcement you will hear it now. Press 0 to disable the night switch, press 1 to enable it. Afterwards the new status of the night switch will be announced.
2. You can skip the DTMF menu by passing the desired new status of the night switch (0 for off and 1 for on) as **post dialing digit** when calling the dummy user. By doing so the night switch will be set accordingly and you will only listen to the new status.

This enables you to configure two **speed dial keys** in your SwyxIt! or on your SwyxPhone (assuming your dummy user has the internal number 300):

- Label: Night Switch ON
Number: 3001
- Label: Night Switch OFF
Number: 3000

How it works

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

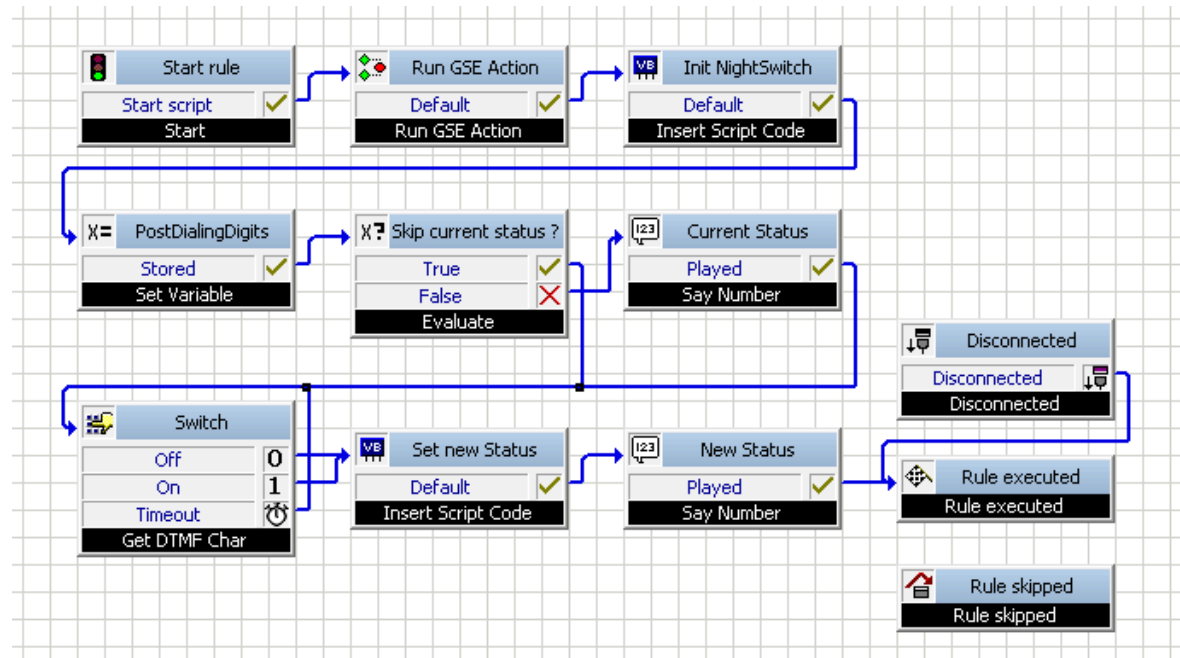
A.2 - Night Switch enabled call routing script

APPENDIX B

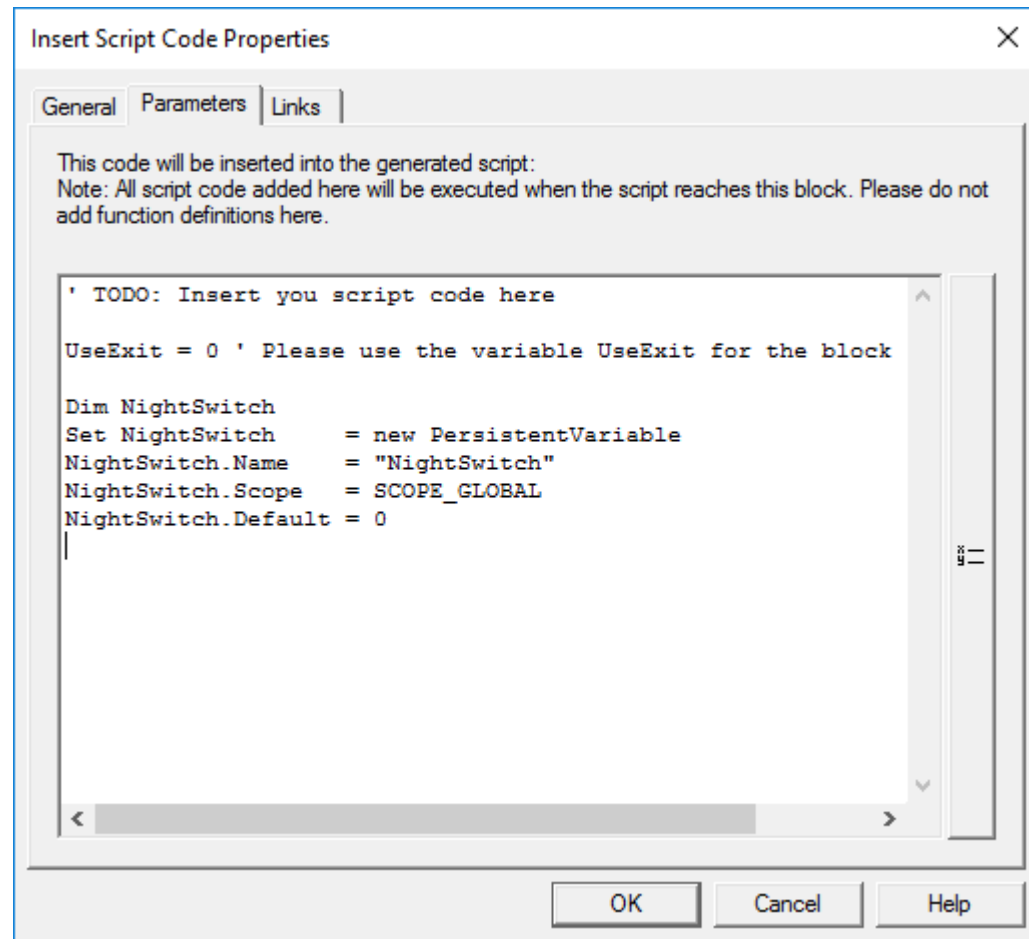
B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

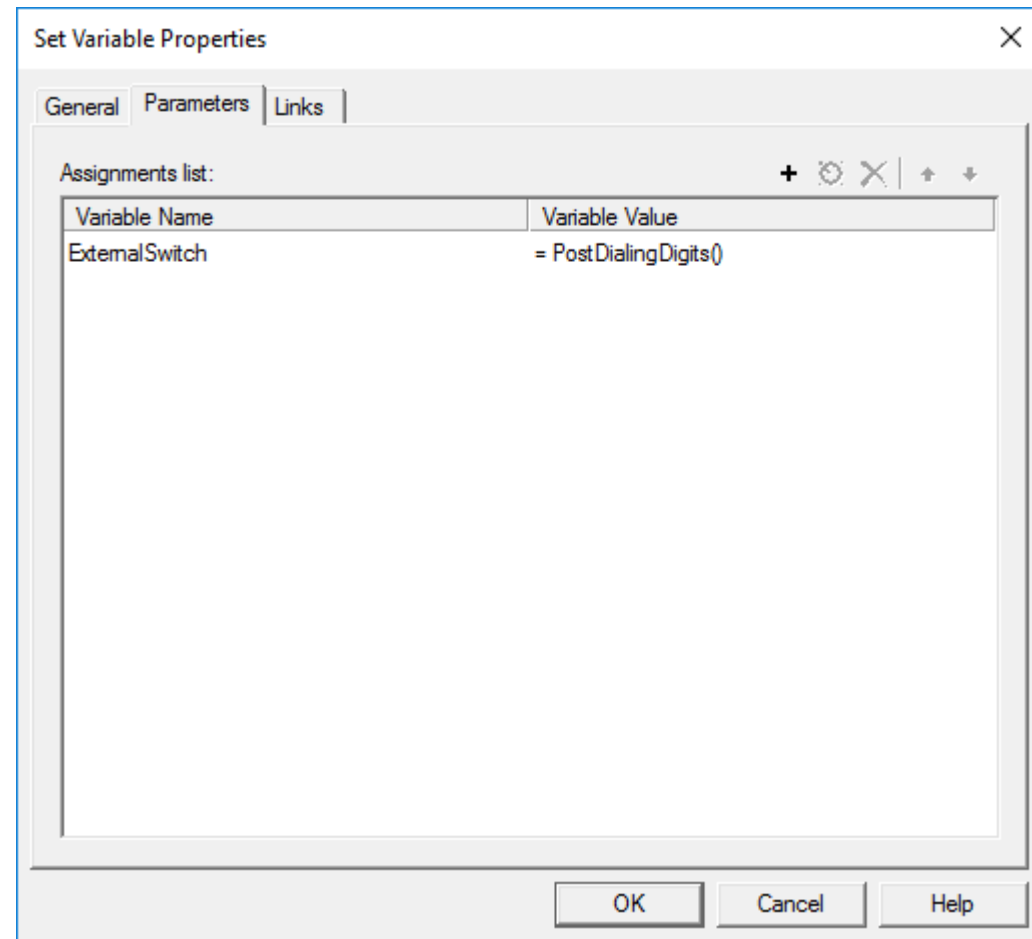
B.3 - Version History



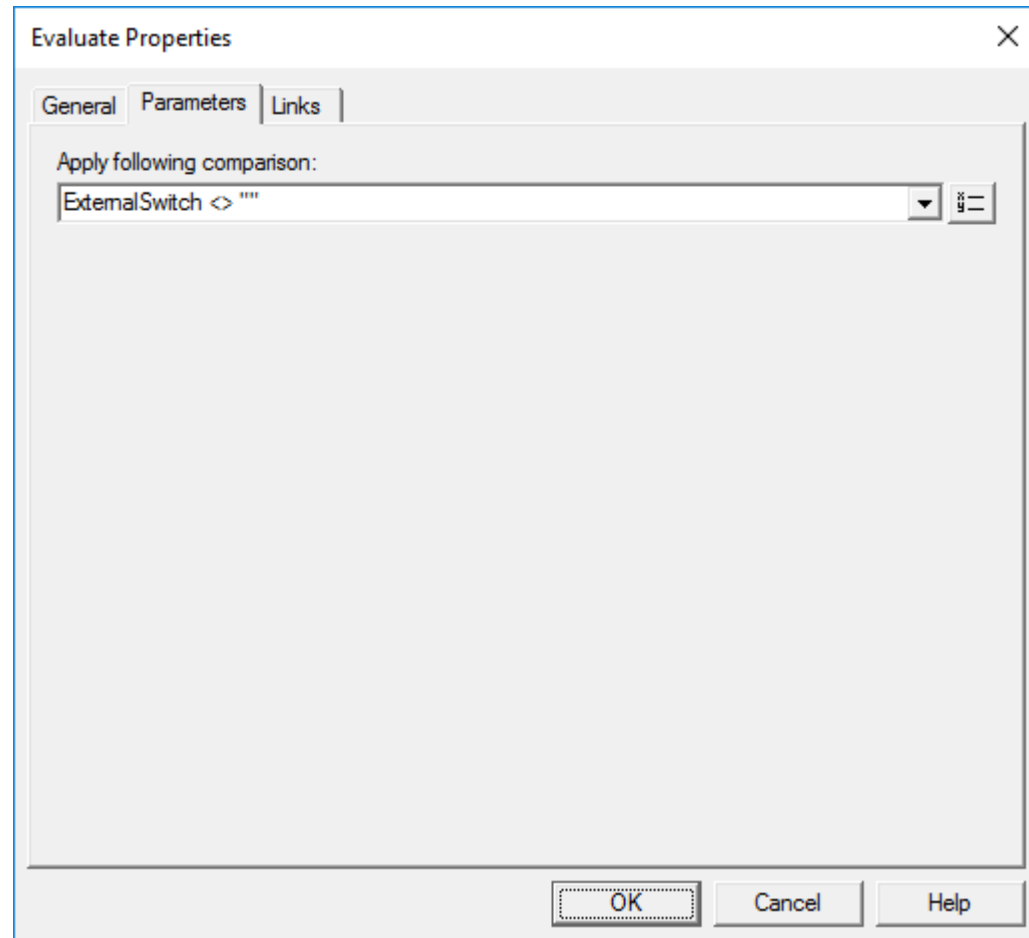
1. Initialize the **NightSwitch** persistent variable.



2. Store **post dialing digits** into own variable.



3. Do we have any **post dialing digits** ?



4. DTMF menu. If we have any post dialing digits within our own variable they will be taken as DTMF input. The block also returns the pressed DTMF key or the post dialing digit in another variable.

Get DTMF Char Properties

General Parameters Links

☒ Get input from variable: ExternalSwitch

☒ Play announcement during DTMF detection:
beep.wav (Template)

... [Play] [Stop] [Pause] [Reset] [Close]

Announcement repetitions:
0

Interval between repetitions:
1

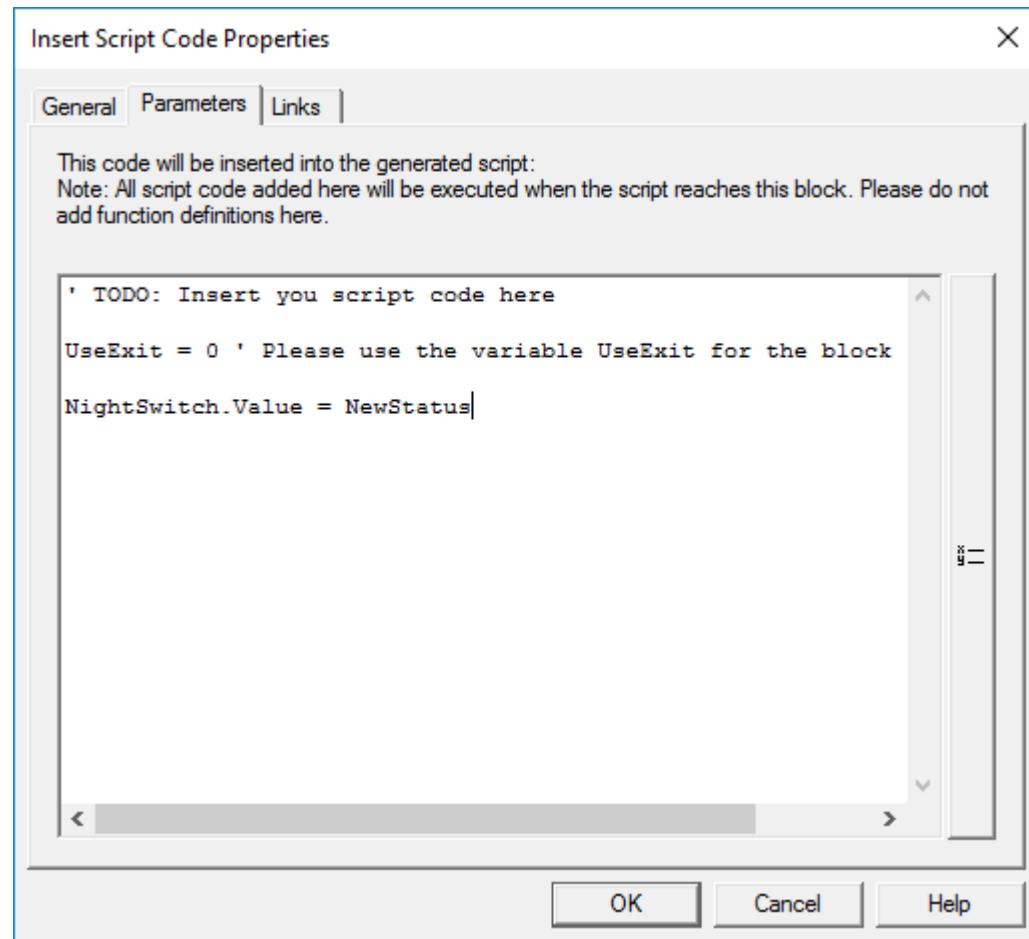
☒ Max detection time (starting after last announcement) in seconds:
10

☒ Save input in variable: NewStatus

☒ Replace variable content
☐ Append to variable content

OK Cancel Help

5. Store the selected new status of the night switch into the persistent variable.



To make use of the Night Switch in any call routing script see here: [A.2 - Night Switch enabled call routing script](#)

Hint: the Persistent Variables can be used with **ANY** SwyxWare version. The included examples and the GSE Action itself however might be stored with the most recent SwyxWare version of the time the Persistent Variable version was released. This means that you might need to use the latest SwyxWare version to be able to open the included GSE Rules (.rse files) and the GSE Action (.ase file) within your GSE.



By Tom Wellige
May 14, 2022

 Share

Next Page >

A.2 - Night Switch enabled call routing script

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - A.2 - Night Switch enabled call routing script

Followers

0

VBScript

This script is an example to show how to use the **Night Switch** in any call routing script of any user.

How to install it

- Make sure you have the Persistent Variable extension properly installed
- Open **Call Routing Manager** of the user you want a Night Switch enabled call routing script for
- Create a **new rule** using the GSE
- **Import** the file **NightSwitchEnabledScript.rse** from the download package
- **Save** the rule and exit the GSE

4.1 - Name

4.2 - Value

4.3 - Default

4.4 - Scope

4.5 - Namespace

4.6 - UserID

4.7 - GroupID

4.8 - CallerName

4.9 - CallerNumber

4.10 - Created

4.11 - CreatedByName

4.12 - CreatedByNumber

4.13 - Modified

4.14 - ModifiedByName

4.15 - ModifiedByNumber

4.16 - Forget

4.17 - LatestError

4.18 - LatestDescription

SCOPES

5.1 - SCOPE_NAMESPACE

5.2 - SCOPE_USER

5.3 - SCOPE_GROUP

5.4 - SCOPE_GLOBAL

MISCELLANEOUS

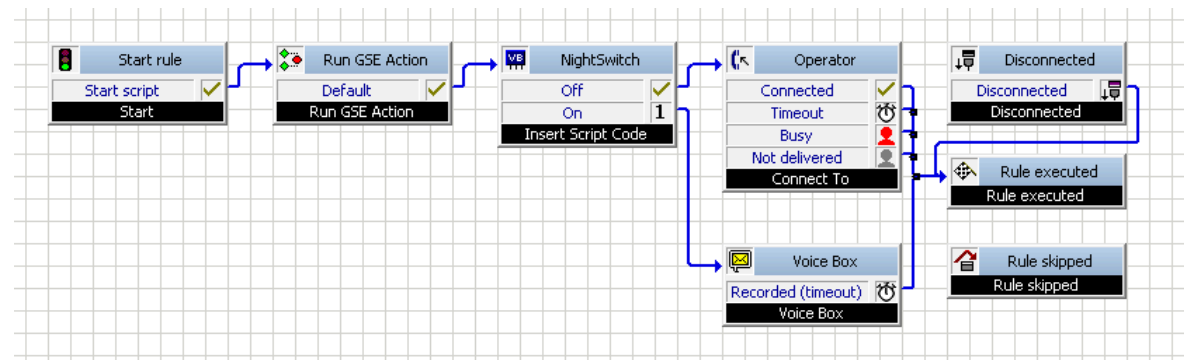
6.1 - Database field definitions

6.2 - Master Standby Installations

How to use it

Just call the user. If the night switch is off the call will be connected to the user. If the night switch is on you will end up in his voicemail.

How it works



The persistent variable **NightSwitch** will be initialized. It's global scope makes sure that it can be accessed from any user. The content of the NightSwitch variable will be passed to the exits of the block. If the content is 0 the default exit will be taken, meaning the night switch is off. If the content is 1 the "1" exit will be taken, meaning the night switch is on.

The default value of the night switch is 0, meaning if the NightSwitch variable hasn't been set by any other script before (i.e. the [A.1 - Night Switch Manager](#) script) the night switch is off.

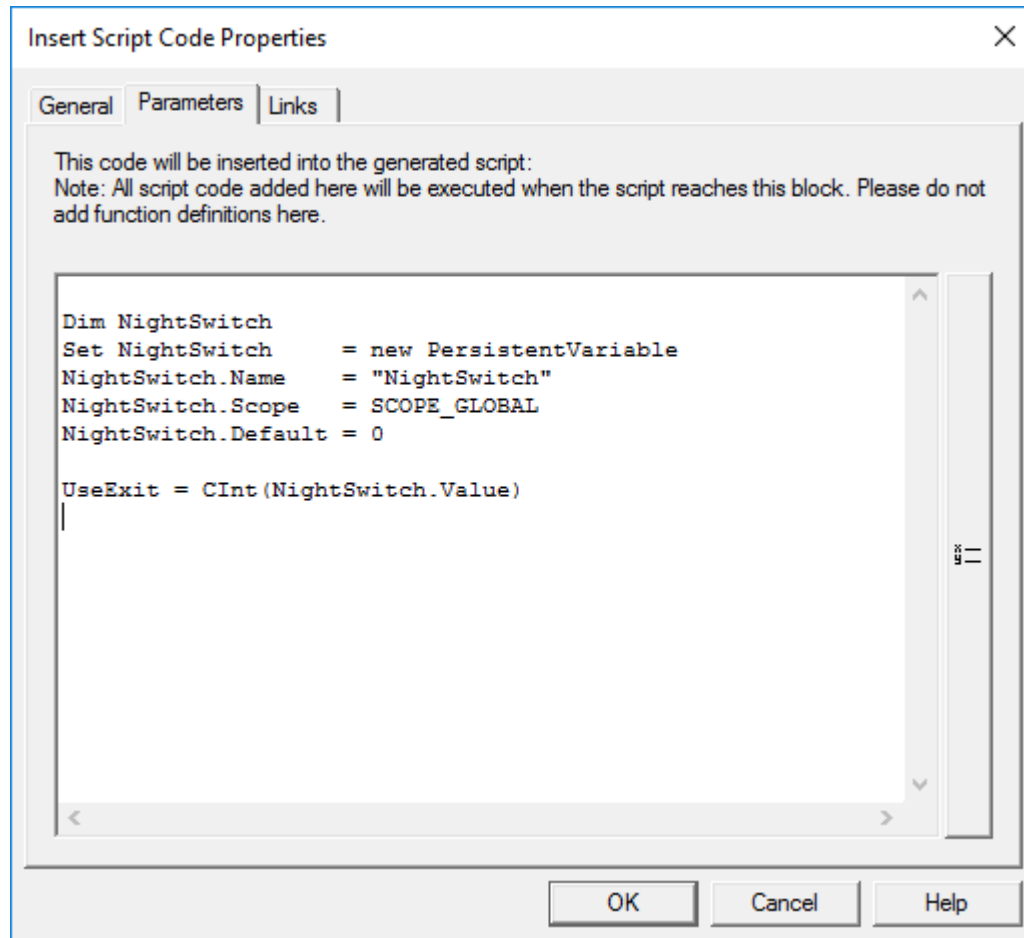
- 6.3 - Use SQL Server on different machine
- 6.4 - Use database with different name (other than default name)
- 6.5 - Use different database (other than MS SQL Server, e.g. MySQL or Oracle)
- 6.6 - Usage of Persistent Variables outside of a call routing script

APPENDIX A

- A.1 - Night Switch - Night Switch Manager
- A.2 - Night Switch enabled call routing script

APPENDIX B

- B.1 - Usage outside call routing - Simple ASP webpage
- B.2 - Usage outside call routing - Simple VBS/WSF script
- B.3 - Version History



Please note that the **value** of the persistent value is converted into an integer using **CInt** before being passed into the **UseExit** variable. This is just to make sure that regardless of what datatype has been passed into the persistent variable, it is used here in the correct data type.

The above mentioned script to manage the status of the night switch can be found here: [A.1 - Night Switch Manager](#)

Hint: the Persistent Variables can be used with **ANY** SwyxWare version. The included examples and the GSE Action itself however might be stored with the most recent SwyxWare version of

the time the Persistent Variable version was released. This means that you might need to use the latest SwyxWare version to be able to open the included GSE Rules (.rse files) and the GSE Action (.ase file) within your GSE.



By Tom Wellige
May 14, 2022

 Share



Previous Page

A.1 - Night Switch - Night Switch Manager

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - B.1 - Usage outside call routing - Simple ASP webpage

Followers

0

VBScript

1.1.0

This example demonstrates how to make use of Persistent Variables in an **ASP** (Active Server Page) web page.

You will find it in the download package in the following folder

- \examples\External\asp

This little web page is meant to be used within a **WebExtension** in a SwyxIt! skin.

It displays the current status of the installed **Night Switch** example (see chapters A.1 and A2).

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

By clicking into the web page the status of the night switch will be toggled. The page also refreshes itself every 2 seconds, so if the night switch is getting switched from somewhere else, this page will also automatically shows the new status.

Default.asp

```
<meta http-equiv="refresh" content="2;url=<%=Request.ServerVariables("SCRI

<style>
    body.on { background: #FFFF00; }
    body.off { background: #C0C0C0; }
    a      { padding:    48% 50%; }
</style>

<!-- #include file = "PersistentVariables.inc"-->

<%
    ' configure the complete db connect string
    g_sPersistentVariableConnectionString = _
        "Provider=sqloledb;" & _
        "Data Source=WS-WELLIGE04;" & _
        "Initial Catalog=" & PERSISTENT_VARIABLE_DATABASE & ";" &
        "User Id=PersistentVariables;" & _
        "Password=PersistentVariables"

    ' initialize persistent variable
    Dim NightSwitch
    Set NightSwitch = new PersistentVariable
    NightSwitch.Name = "NightSwitch"
    NightSwitch.Default = 0
```

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

```
' evaluate url parameter
sUrlSwitch = Request.QueryString("switch")
if (sUrlSwitch <> "") and IsNumeric(sUrlSwitch) then NightSwitch.V

' set display styles
if NightSwitch.Value = 0 then
    sStyle = "off"
    sSwitch = "1"
else
    sStyle = "on"
    sSwitch = "0"
end if

Set NightSwitch = Nothing

sURL = Request.ServerVariables("SCRIPT_NAME") & "?switch=" & sSwit

%>

<body class="<%=sStyle%>">
    <a href="<%=sURL%>"> </a>
</body>
```

You have to make sure, that the **PersistenVariables.inc** file is in the same folder as the **default.asp** file.

Please find a complete explanation of the usage of persistent variables outside of call routing scripts here:

- [6.6 - Usage of Persistent Variables outside of a call routing script](#)



By Tom Wellige

May 14, 2022

 Share

Next Page

[B.2 - Usage outside call routing - Simple VBS/WSF ...](#)



Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

```
1 NightSwitch.vbs
2
3 Dim NightSwitch
4 Set NightSwitch = new PersistentVariable
5 NightSwitch.Name = "NightSwitch"
6
7 ' the NightSwitch can be accessed by any user
8 NightSwitch.Scope = SCOPE_GLOBAL
9
10 ' by default the NightSwitch is off ( 0 - off / 1 - on )
11 NightSwitch.Default = 0
12
13 ' pass status of NightSwitch directly on block exists
14 UseExit = CInt(NightSwitch.Value)
```

Persistent Variables - B.2 - Usage outside call routing - Simple VBS/WSF script

Followers

0

VBScript

1.1.0

This example demonstrates how to make use of Persistent Variables in a WSF (Windows Scripting Host File) file, i.e. a VBScript file.

You will find it in the download package in the following folder

- \examples\External\vbs

This little script is meant to be connected with a shortcut SwyxIt! skin element. By clicking on that shortcut the .wsf file will be started.

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

By starting this .wsf file the current status of the installed **Night Switch** will be toggled.

Toggle.wsf

```
<package>
<job id="Toggle_Example">
<script language="VBScript" src="PersistentVariables.vbs"/>
<script language="VBScript">

' configure the complete db connect string
g_sPersistentVariableConnectionString = _
    "Provider=sqloledb;" & _
    "Data Source=WS-WELLIGE04;" & _
    "Initial Catalog=" & PERSISTENT_VARIABLE_DATABASE & ";" & _
    "User Id=PersistentVariables;" & _
    "Password=PersistentVariables"

' initialize persistent variable
Dim NightSwitch
Set NightSwitch      = new PersistentVariable
NightSwitch.Name     = "NightSwitch"
NightSwitch.Default  = 0

if NightSwitch.Value = 0 then
    NightSwitch.Value = 1
else
    NightSwitch.Value = 0
end if

Set NightSwitch = Nothing
```

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History

```
WScript.Quit  
</script>  
  
</job>  
</package>
```

You can call this .wsf file directly from a command prompt:

```
C:\PersistentVariables> Toggle
```

You have to make sure, that the **PersistenVariables.vbs** file is in the same folder as the **Toggle.wsf** file.

Please find a complete explanation of the usage of persistent variables outside of call routing scripts here:

- [6.6 - Usage of Persistent Variables outside of a call routing script](#)



By Tom Wellige
May 14, 2022

Share

[Previous Page](#)
B.1 - Usage outside call routing - Simple ASP webp...

[Next Page](#)
B.3 - Version History

Theme ▼

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

Persistent Variables

INTRODUCTION

0 - Introduction

PREPARATIONS

1 - Preparations

SETUP

- 2.1 - Setup Database
- 2.2 - Setup Database Table
- 2.3 - Setup GSE Action
- 2.4 - Setup an updated version

USAGE

- 3.1 - Create a persistent variable
- 3.2 - Store a value
- 3.3 - Retrieve a value
- 3.4 - Simple Script - Store Variable
- 3.5 - Simple Script - Retrieve Variable

PROPERTIES & METHODS

Persistent Variables - B.3 - Version History

Followers

0

Version v1.0.0, Released 25.01.2011

Initial Release

Version v1.0.1, Released 14.03.2017

This version fixes 2 problems:

- MSSQL EXPRESS databases are not detected automatically
- Runtime error in Master-Standby setup (user id too large)

Version v1.1.0, Released 27.11.2017

Persistent Variables can be used now also outside of call routing scripts

Examples for external usage have been added:

- ASP (Active Server Page) web page
- WSF (Windows Scripting Host File) file, i.e. VBScript file

- 4.1 - Name
- 4.2 - Value
- 4.3 - Default
- 4.4 - Scope
- 4.5 - Namespace
- 4.6 - UserID
- 4.7 - GroupID
- 4.8 - CallerName
- 4.9 - CallerNumber
- 4.10 - Created
- 4.11 - CreatedByName
- 4.12 - CreatedByNumber
- 4.13 - Modified
- 4.14 - ModifiedByName
- 4.15 - ModifiedByNumber
- 4.16 - Forget
- 4.17 - LatestError
- 4.18 - LatestDescription

SCOPES

- 5.1 - SCOPE_NAMESPACE
- 5.2 - SCOPE_USER
- 5.3 - SCOPE_GROUP
- 5.4 - SCOPE_GLOBAL

MISCELLANEOUS

- 6.1 - Database field definitions
- 6.2 - Master Standby Installations

Added UserID as new property (needed for usage scenarios outside of call routing scripts)
The NightSwitch example which was available as single download is now also part of this download

Version v1.2.0, Released 18.01.2018

This version fixes a bug in the detection of what SQL server variant is installed (full or express) which caused after some windows updates an up to 20 seconds delay when accessing the content of a persistent variable.

Version v1.2.1, Released 14.04.2023

This version fixes 2 problems:

- A theoretical security vulnerability in the parameters of the GSE action.
- Runtime error when using a self defined database connection string in the parameters of the GSE action.

Version v1.3.0, Released 25.02.2024

This version adds the following features:

- New read-only properties "Created", "CreatedBy", "Modified" and "ModifiedBy" added.
- New method "Forget" added to forget a variable. This deletes the variable from the database.

Version v1.4.0, Released 29.02.2024

This version adds the following features and fixes:

- Added new read-only properties "CreatedByName", "CreatedByNumber", "ModifiedByName" and "ModifiedByNumber".
- Removed previously added properties "CreatedBy" and "ModifiedBy" again.
- Added new property "GroupID" for better handling of group call routing scripts.
- Added new scope "SCOPE_GROUP" for better handling of group call routing scripts.
- Fixed a problem with group call routings where user and group scopes could interfere.
- Fixed a problem with tracing which was most likely cause by the VBScript Scripting Engine at a 29th of February.

6.3 - Use SQL Server on different machine

6.4 - Use database with different name
(other than default name)

6.5 - Use different database (other than MS
SQL Server, e.g. MySQL or Oracle)

6.6 - Usage of Persistent Variables outside of
a call routing script

APPENDIX A

A.1 - Night Switch - Night Switch Manager

A.2 - Night Switch enabled call routing script

APPENDIX B

B.1 - Usage outside call routing - Simple
ASP webpage

B.2 - Usage outside call routing - Simple
VBS/WSF script

B.3 - Version History



By Tom Wellige

October 27, 2024

Share



Previous Page

[B.2 - Usage outside call routing - Simple VBS/WSF ...](#)

Theme ▼

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