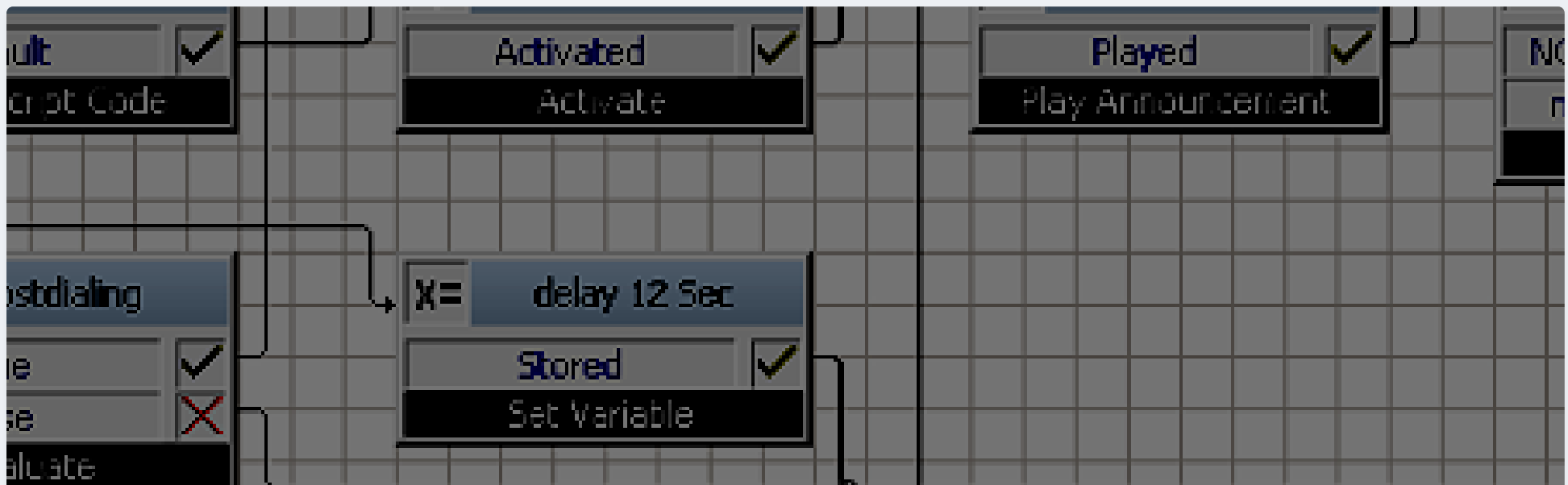


## The Call Routing Guy

A blog by Tom Wellige in General

Followers

1



## #23: Persistent Variables in the cloud (SwyxON)



Entry posted by Tom Wellige in VBScript February 19  
890 views

[Share](#)

Followers

1

Here on Swyx Forum you can find the [Persistent Variables](#), which are variables that keep their content persistently instead of losing it once a call routing script ends. Although they keep their content in a database you are never getting in touch with a database, for you they are more or less plain VBScript variables. That makes them really easy to use.

As long as you have a database at hand which can be used for the persistent variables everything is just fine. But what if you don't have a database at hand, for example you are running your SwyxWare in the cloud, e.g. on the SwyxON platform?

The first thought is of course that you can forget about the persistent variables again and have to look for some other solutions to keep some own content somewhere.

I know that there are some solutions in the SwyxON environment making use of some local storage on the cloud server, e.g. the file system or the Windows registry. While they work on the first glance ok, they all come with a severe hook: your SwyxON tenant can get at any time redeployed into a totally new VM, without any warning upfront and without any notification afterwards. It is totally up to the SwyxON platform management, which is doing such redeployments fully automated when it sees the need for it. There is no way for you to prevent that.

So, if you use some local storage on your SwyxON tenant, it can get at any time lost. In this case your own application hopefully has at least some meaningful default handling if the locally stored information isn't available anymore.



I strongly have to emphasize not to make use of any local storage (file system, registry, ...) on a SwyxON tenant for your own stuff, e.g. in call routings!

But what else possibilities are there to setup for example some simple night switches within the SwyxWare, like they are a real piece of cake when using the persistent variables?

Well, as a matter of fact you can still use the persistent variables! Don't believe me? Read on! It's more easy than you think.

All cloud systems come with the limitation that you don't have access to local resources. The answer to this problem is always the same: keep the information somewhere else, and connect everything with web request. That's the reason why all cloud based solutions usually come with so called REST APIs, i.e. web interfaces, to be able to connect to them easily.

So the answer to our problem is:

1. use a local resource on your premise to keep the information, in this case:  
an own database and own web server (IIS), along with a small ASP website, which uses the **persistent variables**
2. use web requests from within the call routing in your SwyxON tenant to access those "distant" persistent variables

There are already a ton of examples here on Swyx Forum ([Open ECR Extensions](#)) in how to gain access to REST API based systems from within the SwyxWare call routing.

And as a matter of fact the other part (the web server directly using the persistent variables) is also already available for many years. Its even well documented, the included example is a web page to be used in a web extension in your SwyxIt! client.

So all what's needed to be done is a little alteration of the provided example code. I will stick with the standard "night switch" example of the persistent variables (a simple on/off switch).

You will see that I do not hard code the name of the persistent variable "NightSwitch", just to keep this more flexible.

Lets do this step by step...

### **Step 1 - web server / web application using the persistent variables**

1.1 You need to setup an IIS webserver on your premise and make it publicly available. There is a ton of information available on this on the internet, so I take the liberty to divert you to Google in case you need some help on how setup and configure an IIS.

1.2 You also need a database somewhere in your network. This could be an MS SQL Express Server which you can install right from the [SwyxWare complete download package](#). In there you need to setup the [persistent variables database](#) and [persistent variables table](#).

1.3 Setup either a web application or a virtual path within the IIS.

1.4 Store the following code as "**default.asp**" file into that application/virtual path. Place the "**\examples\External\asp\PersistentVariables.inc**" file from the persistent variable [download file](#) into the very same folder. You also need to modify the connect string (**g\_sPersistentVariableConnectionString**) right at the beginning of the code. It must reflect on how and where you have installed the persistent variables database.

▼ Code

```
<!-- #include file = "PersistentVariables.inc"-->
<%

    ' configure the complete db connect string
    g_sPersistentVariableConnectionString = _
        "Provider=sqloledb;" & _
```

Expand ▼

As you can see, the persistent variables still keep all database stuff away from you. It's just that they are now on the web server and not within the call routing anymore.

## Step 2 - use web requestes within call routing to access persistent variable

As the persistent variables are now used on the web server and not within your SwyxWare call routing, you don't need to install them into the SwyxWare anymore.

You just need to **copy & paste** the following code into the **Start** block of your call routing rule. Furthermore you need to update the first line, where the **URL\_BASE** is defined, by the address of your web server.

▼ **Code**

```
Const URL_BASE      = "https://my-pv-domain/pv/"

Const URL_GET_PV     = "default.asp?type=get&name=%name%"
Const URL_SET_PV     = "default.asp?type=set&name=%name%&value=%value%"
Const URL_TOGGLE_PV = "default.asp?type=toggle&name=%name%"
```

Expand ▼

The code provides three simple functions to **get**, **set** or **toggle** a persistent variable (on/off), i.e. the night switch.

Coming back to the night switch example in the download package of the persistent variables. Here you need to make the following modifications:

- [A.1 - Night Switch - Night Switch Manager](#)

- Remove the **"GSE Action"** block, the first **"Insert Script Code"** block (**Init NightSwitch**) and the **"Insert Script Code"** block (**Set new Status**) completely.
- Within the **"Say Number"** Block (**Current Status**) you replace the current **"String of digits"** by

```
= GetPersistentVariable("NightSwitch")
```

- Within the **"Say Number"** Block (**New Status**) you replace the current **"String of digits"** by

```
= SetPersistentVariable("NightSwitch", NewStatus)
```

- A.2 - Night Switch enabled call routing script
  - Remove the "**GSE Action**" block completely.
  - Replace the code in the "**Insert Script Code**" block by

```
UseExit = CInt(GetPersistentVariable("NightSwitch"))
```

Of course two updated .rse files (**NightSwitchManager\_Web.rse** and **NightSwitchEnabledScript\_Web.rse**) containing these both example are included in the download link below as well.

 Don't forget to update the **URL\_BASE** constant in the first line of the code within the **Start** block of these examples!

Although this blog article turned out to be a little bit lengthy, the principle behind is quite easy, as you hopefully grasped.

In future updates of the persistent variables I will also add all the above stuff directly into the download package.

In the meantime you can download everything from here:



**PersistentVariables\_SwyxON.zip**

13.74 kB · 27 downloads

Enjoy!

PS: don't miss to take a look into the [ECR Useful Link Collection](#).



Previous entry

#22: Global call routing rules? Meet the "PreProces...

## 0 Comments

There are no comments to display.

## Create an account or sign in to comment

You need to be a member in order to leave a comment

### Create an account

Sign up for a new account in our community. It's easy!

Register a new account

### Sign in

Already have an account? Sign in here.

Sign In Now

Theme ▼

Copyright (c) 2007-2025 by Tom Wellige

Powered by Invision Community