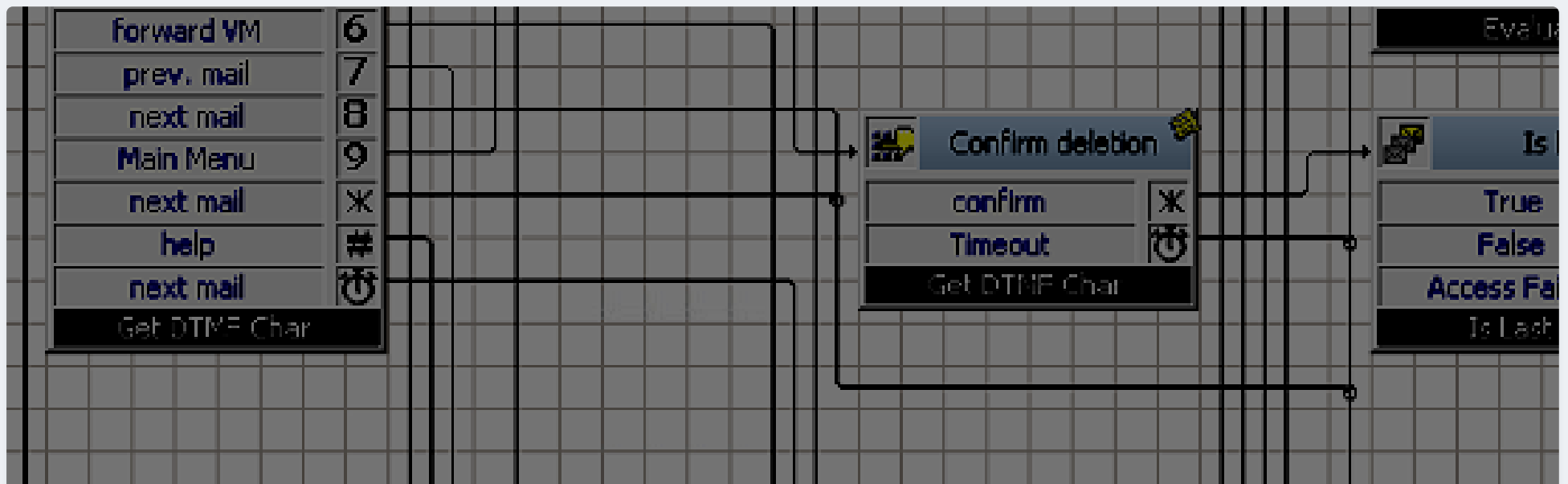


The Call Routing Guy

A blog by Tom Wellige in General

Followers

1



#12: Rotating call distribution to engineers



Entry posted by Tom Wellige in VBScript June 15, 2020
1,583 views

[Share](#)

Followers

1

VBScript

Inspired by a [question](#) in the forum I spent a little bit more time not just giving a few hints as answer in the forum, but instead made a complete call routing script from it.

Task: an incoming call should be provided to 4 different engineers, while in week 1 engineer 1 will get the call, in week 2 engineer 2 will get the call, a.s.o.

In the following I will describe the my approach solving this problem and also how to use the resulting call routing script.

VBScript provides a function **DatePart** which can be used to evaluate the current week of the year. With that number it should be possible to identify a responsible engineer.

There are two approaches possible:

- somehow calculate the engineer from the week of the year number
- use a list of assigned weeks per engineer to identify the engineer from

I have decided to use a list of assigned weeks per engineer as this opens a very easy way of changing the call distribution in case of e.g. annual leave of one engineer.

```
Dim AssignedWeeks(4)
AssignedWeeks(1) = "1,5,9,13,17,21,25,29,33,37,41,45,49"
AssignedWeeks(2) = "2,6,10,14,18,22,26,30,34,38,42,46,50"
AssignedWeeks(3) = "3,7,11,15,19,23,27,31,35,39,43,47,51"
AssignedWeeks(4) = "4,8,12,16,20,24,28,32,36,40,44,48,52"
```

With having that the rest of the implementation is easy. I have implemented a function **GetAssignedEngineer** which simply returns the extension of an engineer (which are set via rule parameters in the final script).

This function gets the week of the year number by the already above mentioned **DatePart** function and then loops through all 4 engineers if this number is part of their assigned weeks list.

```
Function GetAssignedEngineer

    PBXScript.OutputTrace "----> GetAssignedEngineer"

    Dim sReturn
    sReturn = sDefaultExtention

    Dim nCurrentWeek
    nCurrentWeek = DatePart("ww", Now)
    PBXScript.OutputTrace "nCurrentWeek = " & nCurrentWeek

    Dim i
    For i = 1 To 4

        If IsInAssignedWeeksList(i, nCurrentWeek) Then
            sReturn = Engineer(i)
        End If

    Next

    GetAssignedEngineer = sReturn

    PBXScript.OutputTrace "sReturn = " & sReturn
    PBXScript.OutputTrace "<---- GetAssignedEngineer"

End Function
```

To check for each engineer if the current week is in his list of assigned weeks I have implemented a function **IsInAssignedWeeksList** which splits the above defined list of weeks into an array and then simply loops through this array to see if the current week is in it.

```
Function IsInAssignedWeeksList ( nEngineer, nCurrentWeek )

    PBXScript.OutputTrace "----> IsInAssignedWeeksList"
    PBXScript.OutputTrace "nEngineer = " & nEngineer
    PBXScript.OutputTrace "nCurrentWeek = " & nCurrentWeek

    Dim bReturn
    bReturn = False

    Dim aWeeks
    aWeeks = Split(AssignedWeeks(nEngineer), ",")

    Dim y, nWeek
    For y = LBound(aWeeks) To UBound(aWeeks)
        nWeek = CInt(aWeeks(y))
        PBXScript.OutputTrace "nWeek = " & nWeek
        If nWeek = nCurrentWeek Then bReturn = True
    Next

    IsInAssignedWeeksList = bReturn

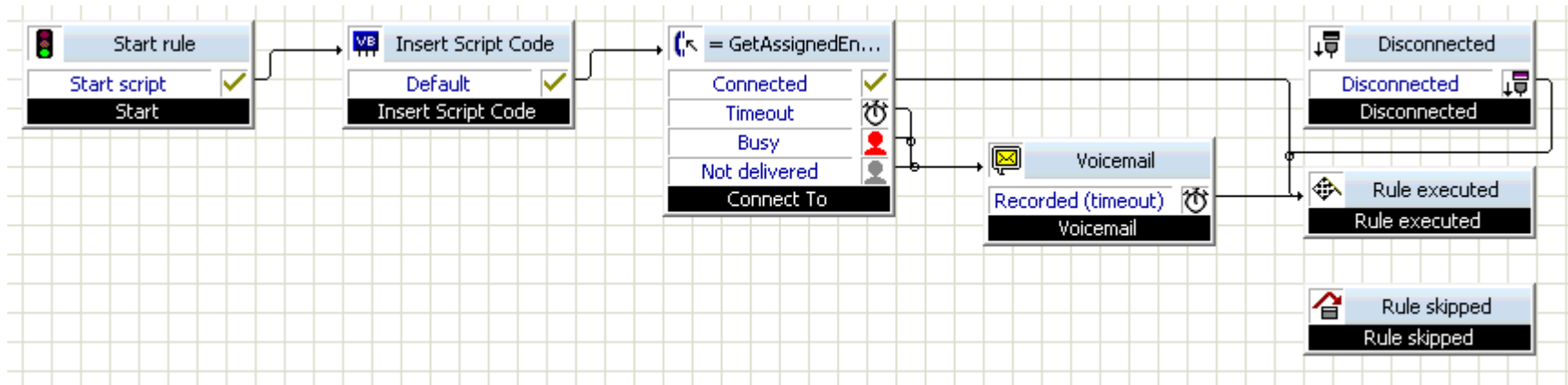
    PBXScript.OutputTrace "bReturn = " & bReturn
    PBXScript.OutputTrace "<---- IsInAssignedWeeksList"

End Function
```

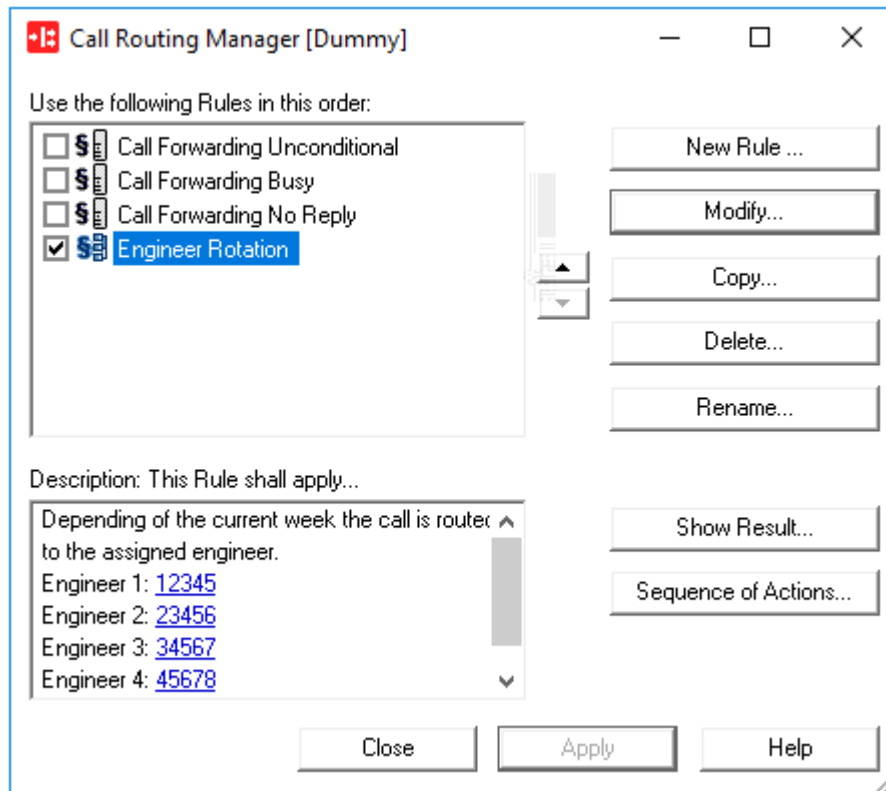
Please note, that in case the current week can not be found in any of the lists, the function **GetAssignedEngineer** returns a default destination. This will make sure that if the lists are getting modified and a week will be forgotten to be added, the function will provide a valid destination to connect

the call to.

The call routing script becomes straight forward with having all the logic implemented in VBScript code:



All VBScript code can be found in the **Start** block. The **Insert Script Code** block is just used to copy the extensions from each engineer which can be comfortable configured as rule parameters into the VBScript code part of the call routing script.



There is also a 5th parameter which defines the default number the call should be delivered to in case a week can't be found in any of the week lists.

To connect the call to an engineer the **Connect To** block directly calls the **GetAssignedEngineer** function:

The screenshot shows the 'Connect To Properties' dialog box with the 'General' tab selected. The 'Connect call for' field is set to '15' seconds. The 'Destination' dropdown is set to '= GetAssignedEngineer', which is highlighted with a yellow box. The 'Include SwyxIt! Mobile devices' checkbox is checked. The 'Use calling party number' dropdown is set to 'Original Calling Party Number'. The 'Alert sound' dropdown is set to 'Play system alert sound'. The 'DTMF interruption' section is expanded, showing a 'Mask' dropdown set to 'All'. The 'Save input in variable' checkbox is unchecked, and the 'Replace variable content' radio button is selected. The 'Proceed with destination's call routing' checkbox is unchecked. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

Connect To Properties

General Parameters Links

Connect call for 15 seconds to:

☒ = GetAssignedEngineer

☐ original destination

☒ Include SwyxIt! Mobile devices

Use calling party number: Original Calling Party Number

Alert sound:

☒ Play system alert sound

☐ [Empty dropdown]

☐ DTMF interruption

Mask: All

☐ Save input in variable:

☒ Replace variable content

☐ Append to variable content

☐ Proceed with destination's call routing

OK Cancel Help

If the call can't be delivered I have just added a **Voicemail** option to play it simply here.

And this sums the entire call routing up. One might say its overdone, but this is simply my style of call routing scripts, even including tracing information in case one needs to debug the script.

You can use the following link to download the script (please note, that you need to be a logged in user of this forum in order to be able to download any file from here):



EngineerRotation.rse

7.41 kB · 128 downloads

To use this file simply open a new empty call routing rule with the GSE and the use **File | Import...** to import the above .rse file.
A more complete explanation on how to get call routing functionality from A to B can be found in this blog article:

- [#4: Take me to your leader!](#)

In case you want to learn all the details behind the used **rule parameters** in this script, the following blog article will enlighten you:

- [#6: Make it easy!](#)

Enjoy!

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



Previous entry

#11: What time is it?

Next entry

#13: Lua is coming



5 Comments



DNW66 ● 0

Posted October 5, 2020

Hi Tom,

How does the script determine which day (Sunday or Monday) is the first day of the week? Is there a way to change this?



Tom Wellige +85

Posted October 5, 2020

✓ On 10/5/2020 at 5:12 PM, DNW66 said:



Hi Tom,

How does the script determine which day (Sunday or Monday) is the first day of the week? Is there a way to change this?

Within the function "GetAssignedEngineer" the VBScript build-in function "DatePart" is used. Per default it uses the first day of week setting from the Windows regional settings. If needed that can be overwritten with an optional third parameter. For details check here:

- https://www.w3schools.com/Asp/func_datepart.asp



MarcoFF ● 0

Posted December 7, 2022

Hello Tom,

thanks for that script and the work.

Even if we would raise the 4 engineers to a slightly higher number, I've tried to use the routing.

It looks like that I'm missing a step to get it running - the incoming call is shown in active calls but goes instant to the voicebox message and is ended then. If I remove the voicebox block, there's no incoming call shown in the active calls and (provider side i think) it tells that the called number is not reachable.

It feels like that the steps from script are not passed.

You've added numbers to the engineers (12345, 23456, ...) I tried to replace them with the user who should receive the call - internal number, external number and even user name. But nothing changes.

The button "Show results" in the Call Routing Manager will only show "???" behind all engineers. So what am I doing wrong?



Tom Wellige +85

Posted December 7, 2022

▼ On 12/7/2022 at 11:15 AM, MarcoFF said:



You've added numbers to the engineers (12345, 23456, ...) I tried to replace them with the user who should receive the call - internal number, external number and even user name. But nothing changes.

The button "Show results" in the Call Routing Manager will only show "???" behind all engineers. So what am I doing wrong?

You should check the server trace file for what your modified call routing script is doing.

Details on how to find and read the server trace file from a call routing perspective can be found here:

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



MarcoFF ● 0

Posted December 8, 2022

Hello Tom,

thanks for your reply.

As we're not running the server on premise, I'll have to make some calls to find out how to get my hands on the logfiles. The only logs we have are the mmc logs from the machine the swyxware administration is running on.

And my usual called tech guy seems to be in holidays already 😊 I'll come back as soon as I get any info - when ever that will be.

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

