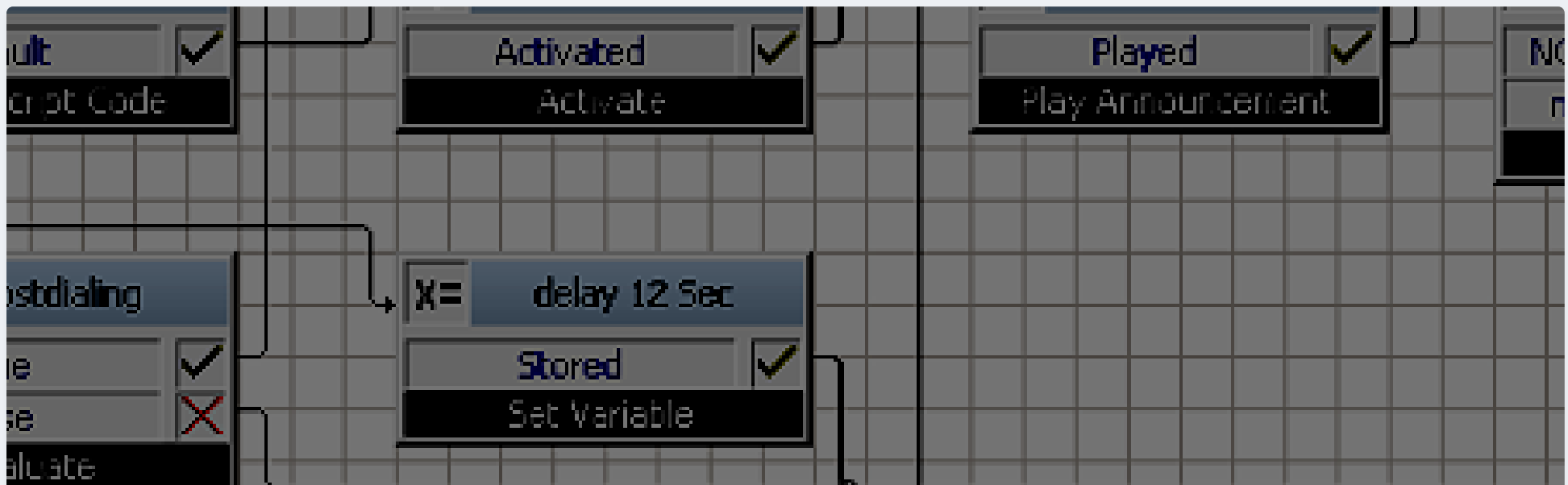


The Call Routing Guy

A blog by Tom Wellige in General

Followers

1



#19: We are many



Entry posted by Tom Wellige in General July 11, 2024
513 views

[Share](#)

Followers

1

Yesterday I was asked an interesting call routing question: is it possible to route a call to different groups, depending on the amount of users currently logged in into one of those groups?

More precisely: if in **Group A** are more than one users logged in, route all incoming calls to this group. If there is no or just one user logged in, route the call to **Group B**.

So the question that needs be answered is how many users are currently logged in into a given group?

With a little bit of scripting code this can be answered quickly.

The **Server Script API** ([VBS/Lua](#)) provides a function **PBXConfig.GetUserByAddress** ([VBS/Lua](#)) which can be used to enumerate all users of a group. Per user a set of properties is available:

- UserID
- Name
- EMailAddress
- **State**
- Numbers
- NumberOfNewVoicemails

The **State** property is the current state of this user. This is a numeric property and can hold any of the following values:

- User state not available
UserStateUnavailable = **0**
- User is logged off
UserStateLoggedOff = **1**

- User is logged on
UserStateLoggedIn = **2**
- User is speaking, at least one call has an external origination or destination
UserStateSpeakingExternal = **3**
- A call is currently alerting at one of the user's devices
UserStateAlerting = **4**
- User is speaking, none of the calls has an external origination or destination
UserStateSpeakingInternal = **5**
- User is away (since SwyxWare 2011)
UserStateAway = **6**
- User has do not disturb status (since SwyxWare 2011)
UserStateDoNotDisturb = **7**
- Busy indication from external presence source (e.g. MS Teams) (since SwyxWare 13)
UserStateActive3rdParty = **8**

Any number above **1** indicates that the user is logged on (regardless if he is free, busy, away, etc.).

So what is needed to do is to count all users in the group who have a **State** value above **1**.

The following is a simple function that sums all such users up and returns the sum.

VBScript

```

'-----
' NumberOfLoggedInUsersInGroup
'
' Returns the number of users which are currently logged in into the given group.
' The users need to have a higher status as "1", i.e. being logged in (regardless if free or busy).
'
' Parameter:
'     sGroup      name or number of group
'
' return value:
'     integer     number of logged in users
'-----

```

```
Function NumberOfLoggedInUsersInGroup ( sGroup )
```

```
    PBXScript.OutputTrace("-----> NumberOfLoggedInUsersInGroup ( sGroup = " & sGroup & " )")
```

```
    Dim nReturn
```

```
    nReturn = 0
```

```
    Dim Users
```

```
    Set Users = g_PBXConfig.GetUserByAddress(sGroup)
```

```
    Dim User
```

```
    For Each User In Users
```

```
        PBXScript.OutputTrace("Found user " & User.Name & " with current state " & User.State)
```

```
        If User.State > 1 Then nReturn = nReturn + 1
```

```
    Next
```

```
NumberOfLoggedInUsersInGroup = nReturn
```

```
PBXScript.OutputTrace("nReturn = " & nReturn)
```

```
PBXScript.OutputTrace("<----- NumberOfLoggedInUsersInGroup")
```

```
End Function
```

Lua

```
-----  
-- NumberOfLoggedInUsersInGroup  
--  
-- Returns the number of users which are currently logged in into the given group.  
-- The users need to have a higher status as "1", i.e. being logged in (regardless if free or busy).  
--  
-- Parameter:  
--      sNumber      uname or number of group  
--  
-- Return:  
--      integer      number of logged in users  
-----  
function NumberOfLoggedInUsersInGroup( sGroup )  
  
    PBXScript.OutputTrace ("-----> NumberOfLoggedInUsersInGroup ( sGroup = " .. sGroup .. " )")  
  
    local nReturn = 0  
    local oUsers  = nil  
  
    oUsers = PBXScript.GetUserByAddress(sGroup)  
    if (oUsers ~= nil) then
```

```

    for i = 1, #oUsers do

        PBXScript.OutputTrace ("Found user '" .. oUsers[i]:Name() .. "' with current state '" .. oUsers[i]:State()
        if (oUsers[i]:State() > 1) then nReturn = nReturn + 1

    end

end

PBXScript.OutputTrace ("nReturn = " .. tostring(nReturn))
PBXScript.OutputTrace ("<----- NumberOfLoggedInUsersInGroup")

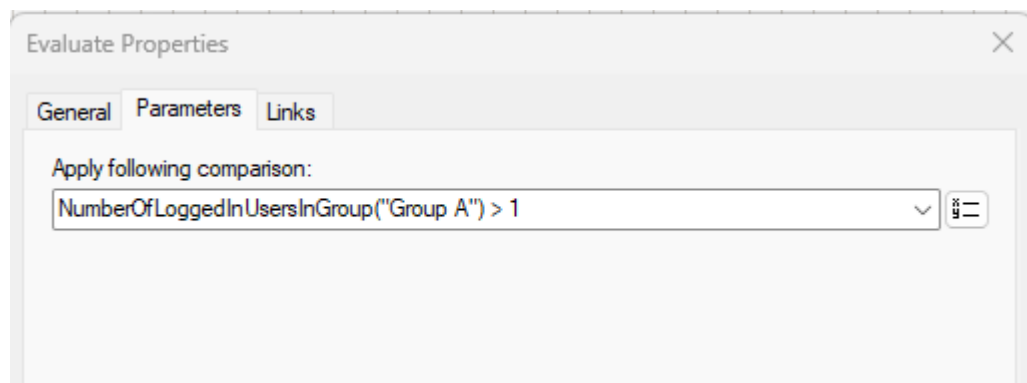
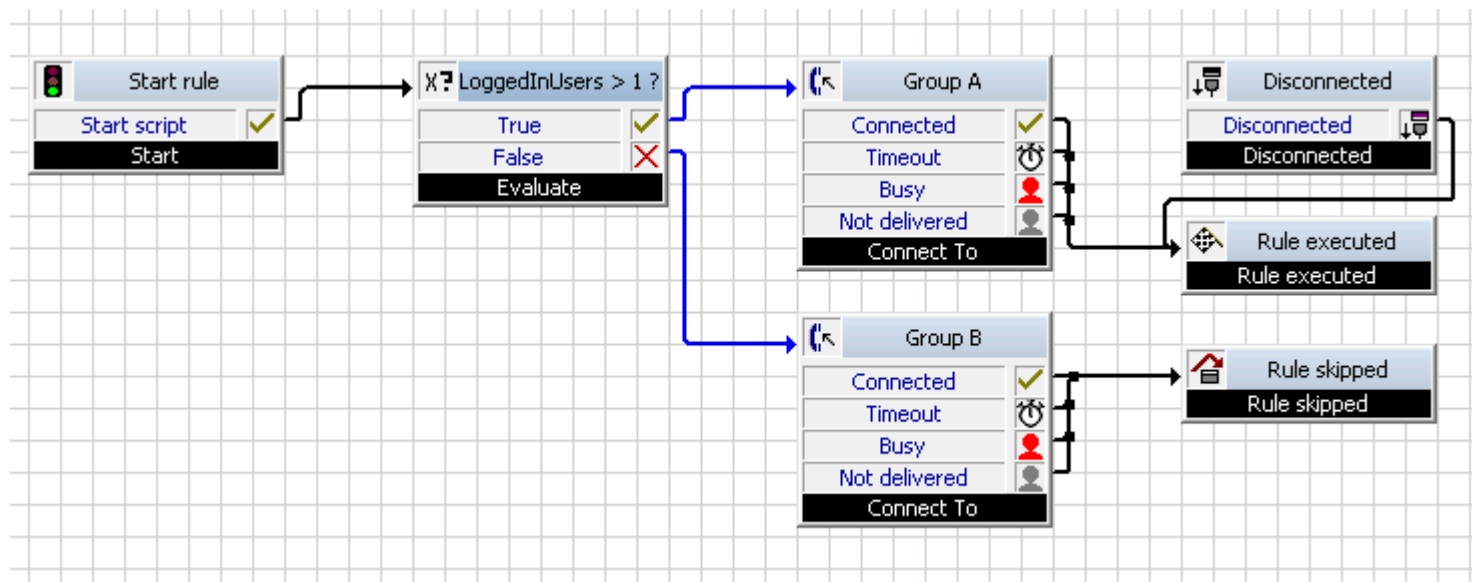
return bReturn
end

```

With placing this code into the **Start** block of a GSE script (as explained [here](#) (VBS) or [here](#) (Lua)) it is possible to use an **Evaluate** block to define a simple condition for routing the calls like needed:

- **NumberOfLoggedInUsersInGroup("Group A") > 1**

The entire GSE script could then look like this:



The above given function is also part of the Function Collection here on the forum page:

- [Function Collection \(VBScript\) - NumberOfLoggedInUsersInGroup](#)
- [Function Collection \(Lua\) - NumberOfLoggedInUsersInGroup](#)

Enjoy!

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

< Previous entry
#18: Record and replace Announcements

Next entry >
#20: How to place your own VBScript/Lua code int...

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