

THE SAP GUI SCRIPTING API

PA251

Exercises / Solutions

CHRISTIAN COHRS, GISBERT LOFF, SAP AG

SCRIPT RECORDING AND PLAYBACK

The examples in this workshop were designed for SAP GUI for Windows 6.20 patch level 34. If you have an older version of SAP GUI then please update the installation. The latest patch is always available at service.sap.com/patches.

Exercise Description

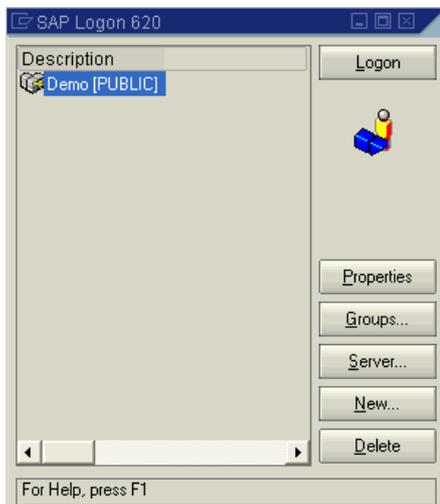
The first exercise will help you to get acquainted with SAP GUI and the sample recorder.

Task1: Connect to the SAP system using SAP GUI for Windows

Double click the SAPLogon icon on your desktop to start SAP GUI for Windows. The icon looks like this:



The SAPLogon dialog will be opened like this:



Now you can connect to the system by double clicking on the entry *“Demo [Public]”* and entering your login data. The speakers will provide you with the required login information. At the time this document was created the name of the system to use was not yet available. Please replace the system name *“Demo [Public]”* wherever it is used in this document by the name given to you by the speakers.

Task 2: Start a transaction

Transactions can be executed from the command field in the system toolbar:

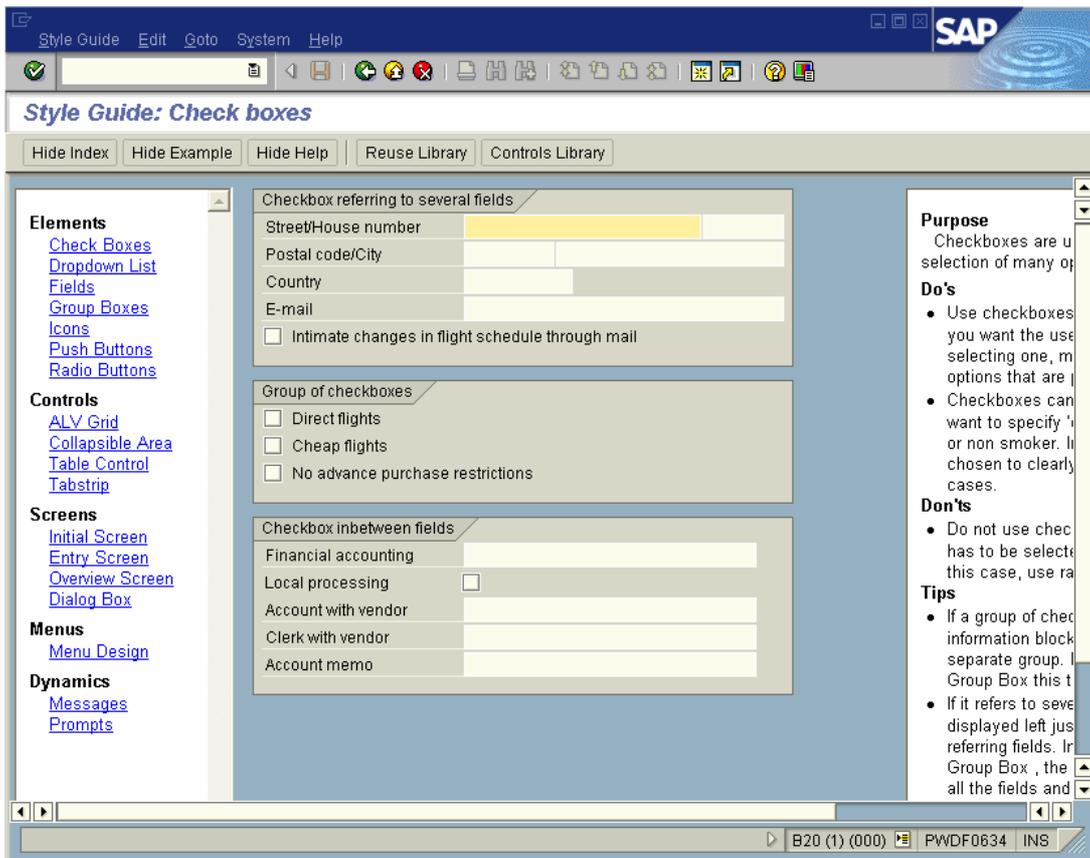


Note: If the command field is closed you can open it by clicking on expand icon displayed its place:



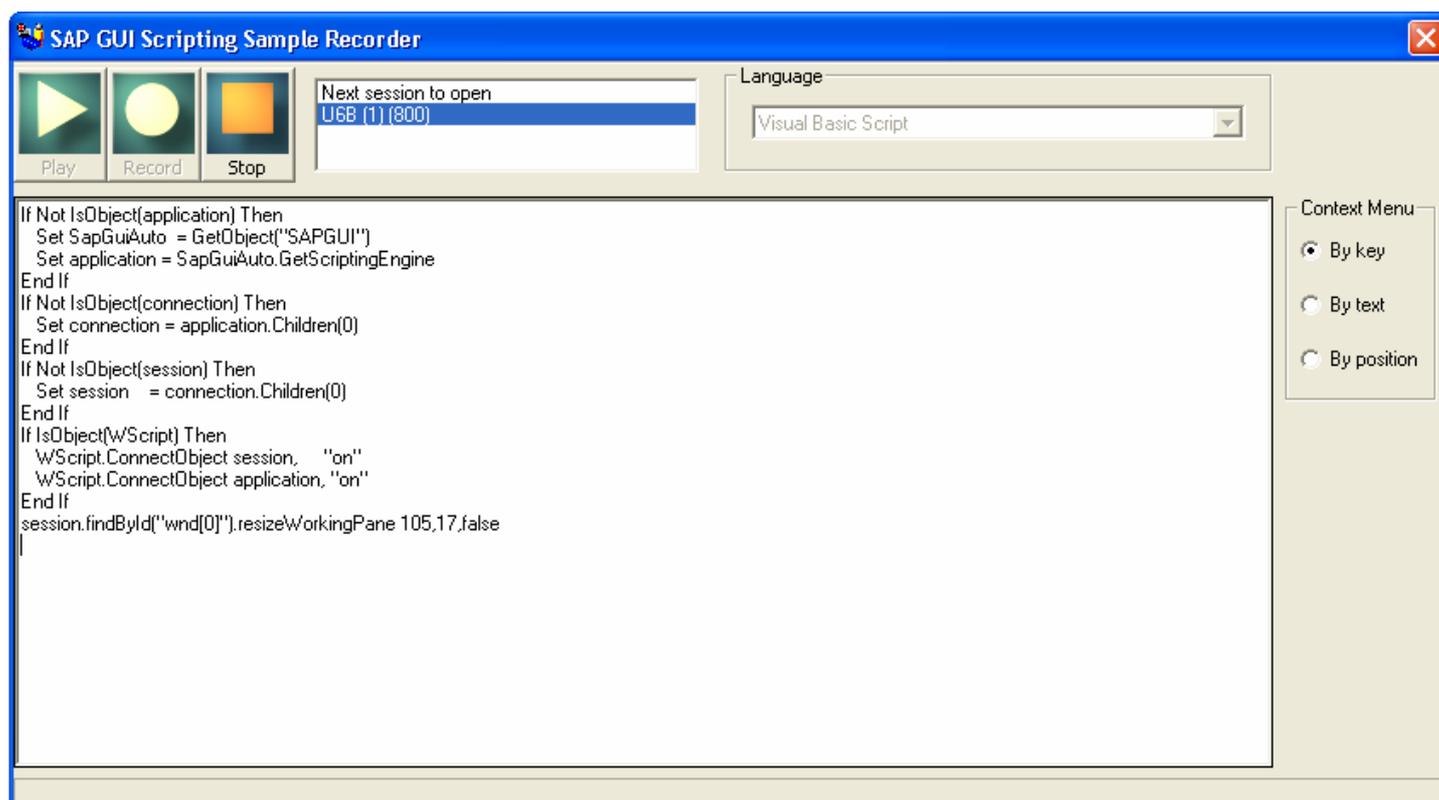
A new transaction can be started by entering the text `/n` followed by the transaction name into the command field. In this workshop we will use the simple `bibs` transaction, which was designed to test elements of the user interface. This allows us to focus on the scripting functionality rather than the requirements for a business process.

Now start transaction `bibs` by entering the text `/nbibs` into the command field and pressing the `Enter` key. The following screen will be displayed:



Task 3: Record a script using the Sample Recorder

The sample recorder may be run from the MS Windows start menu by going to **Programs → SAP Front End → SAP Sample Recorder** program group.



The Sample Recorder records changes made to a screen whenever the changes are sent to server, or when the user stops the recording. All changes made since the last server communication are sent.

Verify this by doing the following tests.

Test 1

- Run the recorder and start recording by pressing the record button while you are on the screen you navigated to in the second task. Make sure the session you want to record is marked in the list of session next to the buttons, and that the language is set to Visual Basic Script. A short scripting header will be generated into the script being recorded.
- Change the values of several fields, such as *Street/House number* or *Country* and press the *Enter* key in the end.
Note: The *bibs* transaction does not save data on the server. After the data has been sent to the server the screen will be cleared.

Notice that the changes are recorded the very moment you press the *Enter* key. Stop the recorder by clicking the Stop button and cancel the file dialog so that the file is not saved.

Test 2

- Change the values of some fields on the current screen **without** pressing the *Enter* key.

- Start the recorder by pressing the record button.
- Press the *Enter* key.

Notice that your changes are recorded even though you made them before starting the recorder. Stop the recorder without saving the file.

Test 3

- Start the recorder by pressing the record button.
- Enter data into some of the fields **without** pressing the *Enter* key.
- Stop the recorder and save the file to the desktop as task3.vbs.

Notice that the changes are recorded when you stop the recorder.

When you close a SAP GUI session while it is being recorded, the recorder will stop recording and open the dialog to save the file. The SAP GUI window will close as soon as you close the recorder dialog by either saving the script or pressing the cancel button.

Task 4: Run a script

Test the script you saved in the previous test by doing the following steps:

- Restart transaction *bibs* by entering the text */nbibs* in the command field and pressing *Enter*.
- Double-Click the script file on the desktop.

Notice that your changes are applied to the screen.

Task 5: View and modify a script

Open the script file in notepad by right clicking it on the desktop and selecting the *Edit* menu item. Look for lines like

```
session.findById("...").text = "your text"
```

For example:

```
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF1").text = "Hello"
```

If you want the script to write a different text to the screen, then change the text on the right side of the assignment to something else. Change the script, save it and run it again by double-clicking it. Notice how the changed text is written to the screen.

Note: SAP GUI Scripting only allows you to enter data that the end user can enter. If your text is longer than the text field you will get an 'Invalid Argument' error, which is raised by the Scripting API.

Task 6 (Optional): Record a script execution

The recorder does not differentiate between manual changes to a screen and changes done by a script. Verify this:

- Go to transaction *bibs* by entering */nbibs* into the command field and pressing the *Enter* key.
- Start the recorder by pressing the record button.
- Double-click the script on the screen.
- Set the focus to SAP GUI.
- Press *Enter*.

Notice both the changes done by the script and the changes done by you are recorded into the new script. Stop the recording without saving the file.

OPEN A CONNECTION USING A SCRIPT

In this section you will create a script, which can log you into the system, take you to a given transaction and pre-set some of the values.

Task 7: Record the steps you want to automate

If you are still connected to the system, then close the connection now.

- Start the connection from the SAPLogon dialog, but do not login at this point.
- Start the recording by pressing the record button of the recorder.
- Enter your login data and press the *Enter* key.
- Go to the transaction *bibs* by entering */nbibs* in the command field and pressing the *Enter* key.
- Write text into some of the text fields **without** pressing the *Enter* key.
- Stop the recording by pressing the stop button of the recorder and save the file as *task7.vbs* to your desktop.

Now open the script in notepad by right-clicking it on the desktop and selecting the *Edit* menu item. Identify the different steps you made in the script:

- Setting the username and password
- Going to transaction *bibs*
- Entering text in transaction *bibs*

You will notice that the password is not recorded. You will however see a line `session.findById("wnd[0]/usr/pwdRSYST-BCODE").text = "*****"` in the beginning of the script.

To successfully connect to system you need to replace the text "*****" with your real password.

While you store a password in a script as part of this task, you should think twice before doing this in a productive environment. Anybody who can read the file will see your password and might use it. In Task 9 you will see how to query the password from the user when executing the script instead of writing it into the script.

When you have edited the script, save it and close the connection.

Now open the connection again without entering the login data and run the script.

Task 8: Open the connection automatically

Currently you still have to open the connection manually by double-clicking it in the SAPLogon dialog. In this task you will learn how to open a connection from the script.

- Open the script **task7.vbs** by right-clicking the file on the desktop selecting *Edit* and save the file as **task8.vbs**
- Replace the line
`Set connection = application.Children(0)`

- by
Set connection = application.OpenConnection("Demo [PUBLIC]")
- Save the script, close any open connection and then run the script.

Task 9: Query information from the user while running a script

The capability to display an input dialog is not implemented within the SAP GUI Scripting object. Instead we rely on the scripting language from which SAP GUI is accessed to provide us with this functionality.

In Visual Basic Script[®] the function *InputBox* is available.

Open the script task8.vbs and save it as task9.vbs. Replace the line that sets the password by the following:

```
Dim MyPassword
MyPassword = InputBox ("Enter your password:", "SAP GUI Scripting Workshop")
session.findById("wnd[0]/usr/pwdRSYST-BCODE").text = MyPassword
```

Finally, save the script, close any open connection and run the script. Please note that the input dialog might not appear in the foreground because of the MS Windows focus handling.

Of course, there is still room for improvement, as the *InputBox* function displays the text you type on the screen for everybody to read. So depending on the level of security you require you might replace the limited *InputBox* function by using a more sophisticated ActiveX control for querying the password, or even read the password from a Single-Sign-On repository.

Task 10 (Optional): Handle multiple Logins gracefully

You may have noticed that the script fails if you try to run it while there is still a connection open. The reason for this is the additional 'Multiple Logon' dialog that pops up after you login. As the script does not handle this dialog it fails on the line following the login.

You may handle this problem by checking whether the dialog appears, and selecting the 'Continue' option from the script.

Open a connection to the system while there is still another connection open. Start the recorder and record the second connection. Now login on this connection, select the 'Continue' option of the following dialog and start the transaction *bibs*. Stop the recording and save the file as task10.vbs.

Now edit the script and replace the "*****" text with your password, and the line *Set connection = application.Children(0)* by *Set connection = application.OpenConnection("Demo [PUBLIC]")*.

Close the second connection while leaving the first connection open, and run the script to see if it successfully handles the dialog.

This script handles all but the first login successfully. There it fails because the additional dialog is not displayed. As a final step, only release the "Multiple Logon" dialog if it appears. The following code checks if a modal window is open.

```
If session.children.count > 1 Then
...
End If
```

Change the script by moving the code to release the “Multiple Logon” dialog into the if-statement:

```
If session.children.count > 1 Then
  session.findById("wnd[1]/usr/radMULTI_LOGON_OPT2").select
  session.findById("wnd[1]/usr/radMULTI_LOGON_OPT2").setFocus
  session.findById("wnd[1]/tbar[0]/btn[0]").press
End If
```

This approach is sufficient for our workshop. In real life you might want to distinguish the ‘Multiple Logon’ dialog from other modal dialogs such as system messages and handle each of them separately.

USING SAP GUI SCRIPTING FOR MONITORING PERFORMANCE

As the scripts run synchronously they will only end when all the steps have been executed by the system. You may therefore check the response time of the system by checking the time it takes to run a script

Task 11: Check the response time once by running a script

Close all open connections, then open one connection to the system and login. Start the recording and do the following steps:

- Start transaction *bibs* by entering the text */nbibs* into the command field and pressing *Enter*.
- Start transaction *dwdm* by entering the text */ndwdm* into the command field and pressing *Enter*.
- Stop recording and save the script as **task11.vbs**

The script you created should look like this in notepad:

```

If Not IsObject(application) Then
    Set SapGuiAuto = GetObject("SAPGUI")
    Set application = SapGuiAuto.GetScriptingEngine
End If
If Not IsObject(connection) Then
    Set connection = application.Children(0)
End If
If Not IsObject(session) Then
    Set session = connection.Children(0)
End If
If IsObject(WScript) Then
    WScript.ConnectObject session, "on"
    WScript.ConnectObject application, "on"
End If
session.findById("wnd[0]").resizeWorkingPane 139,33,false
session.findById("wnd[0]/tbar[0]/okcd").text = "/nbibs"
session.findById("wnd[0]").sendVKey 0
session.findById("wnd[0]/tbar[0]/okcd").text = "/ndwdm"
session.findById("wnd[0]").sendVKey 0

```

You can check the response time of the system by using a timer. Apply the following changes to your script and save it.

```

If Not IsObject(application) Then
    Set SapGuiAuto = GetObject("SAPGUI")
    Set application = SapGuiAuto.GetScriptingEngine
End If
If Not IsObject(connection) Then
    Set connection = application.Children(0)
End If
If Not IsObject(session) Then
    Set session = connection.Children(0)
End If
If IsObject(WScript) Then
    WScript.ConnectObject session, "on"

```

```

WScript.ConnectObject application, "on"
End If
session.findById("wnd[0]").resizeWorkingPane 139,33,false

```

Dim StartTime, EndTime, TimeIt

StartTime = Timer

```

session.findById("wnd[0]/tbar[0]/okcd").text = "/nbibs"
session.findById("wnd[0]").sendVKey 0
session.findById("wnd[0]/tbar[0]/okcd").text = "/ndwdm"
session.findById("wnd[0]").sendVKey 0

```

EndTime = Timer

TimeIt = EndTime – StartTime

MsgBox "Response time: " & TimeIt

This script will check how long it takes to switch between the two transactions.

Task 12 (Optional): Monitor the system performance over time

You usually do not want to monitor a system manually. Instead the monitor should run automatically over a long time and notify you when a problem is detected.

Add the following lines to the script while removing the *MsgBox* line in the end, and save the script as **task12.vbs**.

```

If Not IsObject(application) Then
    Set SapGuiAuto = GetObject("SAPGUI")
    Set application = SapGuiAuto.GetScriptingEngine
End If
If Not IsObject(connection) Then
    Set connection = application.Children(0)
End If
If Not IsObject(session) Then
    Set session = connection.Children(0)
End If
If IsObject(WScript) Then
    WScript.ConnectObject session, "on"
    WScript.ConnectObject application, "on"
End If
session.findById("wnd[0]").resizeWorkingPane 139,33,false

```

For i=1 to 5

Dim StartTime, EndTime

StartTime = Timer

```
session.findById("wnd[0]/tbar[0]/okcd").text = "/nbibs"  
session.findById("wnd[0]").sendVKey 0  
session.findById("wnd[0]/tbar[0]/okcd").text = "/ndwdm"  
session.findById("wnd[0]").sendVKey 0
```

```
EndTime = Timer  
TimeIt = EndTime - StartTime
```

```
If TimeIt > 10 Then  
  MsgBox "Response time alert: " & TimeIt  
End If
```

```
WScript.Sleep(2000)  
Next
```

The *Sleep* call is required as your script would otherwise constantly put load onto the system. With the Sleep call it will only check the response time every 2 seconds.

It is now up to you to decide

- Which steps you want to check
- How often you want to check
- Which threshold you set for raising an alert

SOLUTIONS

Task3.vbs

```

If Not IsObject(application) Then
    Set SapGuiAuto = GetObject("SAPGUI")
    Set application = SapGuiAuto.GetScriptingEngine
End If
If Not IsObject(connection) Then
    Set connection = application.Children(0)
End If
If Not IsObject(session) Then
    Set session = connection.Children(0)
End If
If IsObject(WScript) Then
    WScript.ConnectObject session, "on"
    WScript.ConnectObject application, "on"
End If
session.findById("wnd[0]").resizeWorkingPane 139,33,false
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF1").text = "Hello"
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF2").text = "World"
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF2").setFocus
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF2").caretPosition = 5

```

Task7.vbs

```

If Not IsObject(application) Then
    Set SapGuiAuto = GetObject("SAPGUI")
    Set application = SapGuiAuto.GetScriptingEngine
End If
If Not IsObject(connection) Then
    Set connection = application.Children(0)
End If
If Not IsObject(session) Then
    Set session = connection.Children(0)
End If
If IsObject(WScript) Then
    WScript.ConnectObject session, "on"
    WScript.ConnectObject application, "on"
End If
session.findById("wnd[0]/usr/txtRSYST-MANDT").text = "<Client>"
session.findById("wnd[0]/usr/txtRSYST-BNAME").text = "<Your Username>"
session.findById("wnd[0]/usr/pwdRSYST-BCODE").text = "<Your Password>"
session.findById("wnd[0]/usr/txtRSYST-LANGU").setFocus
session.findById("wnd[0]/usr/txtRSYST-LANGU").caretPosition = 0
session.findById("wnd[0]").sendVKey 0

```

```

session.findById("wnd[0]/tbar[0]/okcd").text = "/nbibs"
session.findById("wnd[0]").sendVKey 0
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF1").text = "Hello"
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF2").text = "World"
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF2").setFocus
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF2").caretPosition = 5

```

Task8.vbs

```

If Not IsObject(application) Then
    Set SapGuiAuto = GetObject("SAPGUI")
    Set application = SapGuiAuto.GetScriptingEngine
End If
If Not IsObject(connection) Then
    Set connection = application.OpenConnection("Demo [PUBLIC]")
End If
If Not IsObject(session) Then
    Set session = connection.Children(0)
End If
If IsObject(WScript) Then
    WScript.ConnectObject session, "on"
    WScript.ConnectObject application, "on"
End If
session.findById("wnd[0]/usr/txtRSYST-MANDT").text = "<Client>"
session.findById("wnd[0]/usr/txtRSYST-BNAME").text = "<Your Username>"
session.findById("wnd[0]/usr/pwdRSYST-BCODE").text = "<Your Password>"
session.findById("wnd[0]/usr/txtRSYST-LANGU").setFocus
session.findById("wnd[0]/usr/txtRSYST-LANGU").caretPosition = 0
session.findById("wnd[0]").sendVKey 0
session.findById("wnd[0]/tbar[0]/okcd").text = "/nbibs"
session.findById("wnd[0]").sendVKey 0
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF1").text = "Hello"
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF2").text = "World"
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF2").setFocus
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF2").caretPosition = 5

```

Task9.vbs

```

If Not IsObject(application) Then
    Set SapGuiAuto = GetObject("SAPGUI")
    Set application = SapGuiAuto.GetScriptingEngine
End If
If Not IsObject(connection) Then

```

```

Set connection = application.OpenConnection("Demo [PUBLIC]")
End If
If Not IsObject(session) Then
    Set session = connection.Children(0)
End If
If IsObject(WScript) Then
    WScript.ConnectObject session, "on"
    WScript.ConnectObject application, "on"
End If
session.findById("wnd[0]/usr/txtRSYST-MANDT").text = "<Client>"
session.findById("wnd[0]/usr/txtRSYST-BNAME").text = "<Your Username>"

```

```

Dim MyPassword
MyPassword = InputBox ("Enter your password:", "SAP GUI Scripting Workshop")
session.findById("wnd[0]/usr/pwdRSYST-BCODE").text = MyPassword

```

```

session.findById("wnd[0]/usr/txtRSYST-LANGU").setFocus
session.findById("wnd[0]/usr/txtRSYST-LANGU").caretPosition = 0
session.findById("wnd[0]").sendVKey 0
session.findById("wnd[0]/tbar[0]/okcd").text = "/nbibs"
session.findById("wnd[0]").sendVKey 0
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF1").text = "Hello"
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF2").text = "World"
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF2").setFocus
session.findById("wnd[0]/usr/subSA_0100_1:SAPLEXAMPLE_ENTRY_SCREEN:0200/subSA_200_2:SAPLEXAMPLE_ENTRY_SCREEN:1200/txtF2").caretPosition = 5

```

Task10.vbs

```

If Not IsObject(application) Then
    Set SapGuiAuto = GetObject("SAPGUI")
    Set application = SapGuiAuto.GetScriptingEngine
End If
If Not IsObject(connection) Then
    Set connection = application.openConnection("Demo [PUBLIC]")
End If
If Not IsObject(session) Then
    Set session = connection.Children(0)
End If
If IsObject(WScript) Then
    WScript.ConnectObject session, "on"
    WScript.ConnectObject application, "on"
End If
session.findById("wnd[0]").resizeWorkingPane 139,33,false
session.findById("wnd[0]/usr/txtRSYST-MANDT").text = "<Client>"
session.findById("wnd[0]/usr/txtRSYST-BNAME").text = "<Your Username>"

```

```

session.findById("wnd[0]/usr/pwdRSYST-BCODE").text = "<Your Password>"
session.findById("wnd[0]/usr/txtRSYST-LANGU").setFocus
session.findById("wnd[0]/usr/txtRSYST-LANGU").caretPosition = 0
session.findById("wnd[0]").sendVKey 0
If session.children.count > 1 Then
    session.findById("wnd[1]/usr/radMULTI_LOGON_OPT2").select
    session.findById("wnd[1]/usr/radMULTI_LOGON_OPT2").setFocus
    session.findById("wnd[1]/tbar[0]/btn[0]").press
End If
session.findById("wnd[0]/tbar[0]/okcd").text = "/nbibs"
session.findById("wnd[0]").sendVKey 0

```

Task11.vbs

```

If Not IsObject(application) Then
    Set SapGuiAuto = GetObject("SAPGUI")
    Set application = SapGuiAuto.GetScriptingEngine
End If
If Not IsObject(connection) Then
    Set connection = application.Children(0)
End If
If Not IsObject(session) Then
    Set session = connection.Children(0)
End If
If IsObject(WScript) Then
    WScript.ConnectObject session, "on"
    WScript.ConnectObject application, "on"
End If
session.findById("wnd[0]").resizeWorkingPane 139,33,false

```

```

Dim StartTime, EndTime, TimeIt
StartTime = Timer

```

```

session.findById("wnd[0]/tbar[0]/okcd").text = "/nbibs"
session.findById("wnd[0]").sendVKey 0
session.findById("wnd[0]/tbar[0]/okcd").text = "/n"
session.findById("wnd[0]").sendVKey 0

```

```

EndTime = Timer
TimeIt = EndTime – StartTime

```

```

MsgBox "Response time: " & TimeIt

```

Task12.vbs

```

If Not IsObject(application) Then
    Set SapGuiAuto = GetObject("SAPGUI")
    Set application = SapGuiAuto.GetScriptingEngine
End If

```

```
If Not IsObject(connection) Then
  Set connection = application.Children(0)
End If
If Not IsObject(session) Then
  Set session = connection.Children(0)
End If
If IsObject(WScript) Then
  WScript.ConnectObject session, "on"
  WScript.ConnectObject application, "on"
End If
session.findById("wnd[0]").resizeWorkingPane 139,33,false

For i=1 to 5
  Dim StartTime, EndTime, TimeIt
  StartTime = Timer

  session.findById("wnd[0]/tbar[0]/okcd").text = "/nbibs"
  session.findById("wnd[0]").sendVKey 0
  session.findById("wnd[0]/tbar[0]/okcd").text = "/ndwdm"
  session.findById("wnd[0]").sendVKey 0

  EndTime = Timer
  TimeIt = EndTime - StartTime

  If TimeIt > 10 Then
    MsgBox "Response time alert: " & TimeIt
  End If

  WScript.Sleep(2000)
Next
```

Copyright 2003 SAP AG. All Rights Reserved

- No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.
- Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.
- Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of Microsoft Corporation.
- IBM®, DB2®, DB2 Universal Database, OS/2®, Parallel Sysplex®, MVS/ESA, AIX®, S/390®, AS/400®, OS/390®, OS/400®, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere®, Netfinity®, Tivoli®, Informix and Informix® Dynamic Server™ are trademarks of IBM Corporation in USA and/or other countries.
- ORACLE® is a registered trademark of ORACLE Corporation.
- UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of the Open Group.
- Citrix®, the Citrix logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® and other Citrix product names referenced herein are trademarks of Citrix Systems, Inc.
- HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.
- JAVA® is a registered trademark of Sun Microsystems, Inc.
- JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.
- MarketSet and Enterprise Buyer are jointly owned trademarks of SAP AG and Commerce One.
- SAP, SAP Logo, R/2, R/3, mySAP, mySAP.com, xApps, mySAP Business Suite, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies.

SAP assumes no responsibility for errors or omissions in these materials.