**VILT**

**SAP Testing with UFT**

[Assumptions 3](#_Toc95744541)

[Introductory Comments 3](#_Toc95744542)

[Lab Overview 4](#_Toc95744543)

[Start-up / Before You Record 4](#_Toc95744544)

[Setting up git and Using git with UFT 4](#_Toc95744545)

[Launching SAP and UFT 5](#_Toc95744546)

[Finish UFT install on Chrome 6](#_Toc95744547)

[Record VA01 with SAPGui 8](#_Toc95744548)

[VA01 Fiori Recording with AI 16](#_Toc95744549)

[VA01 Fiori Recording with “Traditional” Object Identification 20](#_Toc95744550)

[Add VA02 to Fiori (Traditional) 21](#_Toc95744551)

[Add VA02 to SAP GUI 24](#_Toc95744552)

## Assumptions

1. You are familiar/comfortable with using UFT against web sites
2. You have attended the introductory sessions on SAP
3. You have attended the introductory session on the new, enablement SAP environment
4. You are familiar with NimbusClient
5. You are familiar with Blueshift

## Introductory Comments

I recommend that you do all your work in a git/github directory. This is the easiest way to make a “permanent” copy of your work even after the Blueshift order expires. It will also be available for other orders, or for work on NimbusClient. If you are unfamiliar with UFT and git, I recommend you watch this 10-minute video. <https://www.youtube.com/watch?v=t_H5VVigO1o>

When working with SAP, the most difficult thing is often finding/identifying the correct data. SAP has 100s (1000s?) of business rules. Because of this, we will provide the data to be used for the exercises. Please use this data exactly.

When working with SAP, on the screen you will often see fields like this:

Rectangle

Description automatically generated with medium confidence

i.e., to the right of the field, there is a small “double square”. Clicking on this control will bring up a page that allows you search/select for possible values. **DON’T DO IT!**

As mentioned above – SAP data has many rules, and you will almost certainly not be able to identify values that will work.

Clicking on such controls adds many steps to the test but provide little value.

“Real” SAP users rarely if ever use such controls. They know their data and enter values into the edit field.

Similarly, **don’t** click on calendar objects. They are difficult to implement within UFT, and simply typing in a date (usually in the future, so that you test will work going forward) is far better.

Two other things:

If in your recorded test you may see a lot of .Sync method calls, they almost certainly can be deleted, which makes for a much cleaner/easier to read test. However, since one or more \_may\_ be required, comment them out with a ‘, ‘ then when replay succeeds, delete the lines.

If you repeat the business process several times, some fields may be “pre-populated”. This is a problem when recording with UFT, as UFT only records an edit field if a **change** is made in the field. For example, you may well find the Sales Organization pre-populated to 1710. If this is the case:

Change the value to 1711 <tab> or click outside the edit (you should see an event in the recording tool bar)

Change the value back to 1710

After recording, delete the line setting the incorrect value.

Note that this behavior is NOT unique to SAP. It is how UFT works for all edit fields, for all add-ins. It just so happens that it occurs very often with SAP, and rarely with other add-ins.

## Lab Overview

We will be focusing on create a Sales Order, which is the SAP t-code (transaction code) VA01, a very common transaction. After creation we will modify the Sales Order with VA02. (This is common in SAP: 01 -> Create, 02 -> Modify, 03 -> Display). Note: SAP SMEs refer to this by saying: “V A Zero 1”.

We will do this three times. First we will be working in the SAPGui interface, which is the traditional interface that has existed for decades and is still widely used by clients. If you have used WPF against FlightsGUI you will find a lot of the look and feel to be quite similar.

Second, we will repeat this in a browser interface (SAP Fiori), using AI.

Third, repeat with object property/value pairs, which you will find is similar to working with any browser based application.

You may not complete the exercises unless you are extremely skilled with UFT, but I hope you will complete at least the first two exercises. After that, you can complete the exercise(s) that are of the most interest to you, in any order. To summarize:

* Do the first exercise – VA01 with SAPGui
* Do the second exercise – VA01 with AI
* Do the third exercise, VA01 with Fiori (web)
* Then – as time permits work on adding the VA02 t-codes to exercise 1 and 3.

Lastly – feel free to reach out to me ([ron.sercely@microfocus.com](mailto:ron.sercely@microfocus.com)) anytime in the next few weeks, if you want to keep working and need help.

Also, completed scripts are available from my github repository:

<https://github.com/rsercely/sap-training.git>

I also placed this document, and a text file with the values to use for recording VA01 in the repository.

## Start-up / Before You Record

## Setting up git and Using git with UFT

If not familiar with this, watch the video referred to above, but do this AFTER this training session while I am available to answer SAP questions. If you are familiar:

* 1. Create a github repository
  2. Clone that as a local repository
  3. Create a new UFT test within the repository

## Launching SAP and UFT

|  |  |  |
| --- | --- | --- |
|  | In the ADM Solutions folder (on the desktop), click on SoftEther VPN Client Manger |  |
|  | You should have already imported the VPN connection. Make sure you are connected. |  |
|  | From ADM Solutions – launch UFT |  |
|  | Select all SAP and Web add-ins (Web for later so that Fiori will work) |  |

### Finish UFT install on Chrome

You might see the following after launching Chrome within the Blueshift Order. (Other browsers *might* have similar pop-ups. If you do not see any pop-ups like this do not worry)

Note: you will only see the orange “Error” oval the first time you ever launch Chrome within this order. Clicking on the oval launches the text “New extension added….”.

1. Click on the “New extension added…” text

Graphical user interface, text, application, website

Description automatically generated

1. Click on Enable extension

Graphical user interface, text, application, email, website

Description automatically generated

#### Explanation of Above

For UFT to recognize objects within a browser window, an extension or agent must be installed. Details are a bit different for each browser. Usually, after a UFT installation, this is all done automagically, but in this Blueshift Offering, we have found that it is sometimes necessary to do this manual extension enablement process.

This **never** has to be done more than once per order/UFT installation.

#### Rarely, UFT might never appear. To fix this:

1. Open task manager; Sort tasks by name

Note at the bottom of the screen capture, there are two different processes associated with UFT, and you might find multiple copies of either or both. Kill them all. Then re-launch UFT

Graphical user interface, text, application

Description automatically generated

With that, UFT should be open on your desktop.

Unrelated to the launching of UFT, if at any time you see this pop-up – well, it happens. Simply save your test; Click on OK; close UFT; relaunch UFT

Graphical user interface, text, application, email

Description automatically generated

# Record VA01 with SAPGui

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Discussion | | Screenshot | |
|  | Add a new GUI test  Name as you see fit  If using git, be sure to set the Location to be the git repository directory on your filesystem. If needed  Change the location with the CHANGE LINK | |  | |
|  | If in a git repository directory, you will see the red dots in Solution Explorer, which is an indication that there are non-committed changes.  Click the Record button | |  | |
|  | Use these settings for the SAP tab. Most users of UFT with SAPGui use this “auto-login” option. The other tabs don’t matter  Note that the Password value is Password1  But UFT “hides” that with the \*\*\*\*\*\*\* | |  | |
|  | Where did these values come from, from this window (which is available in the ADM Solutions folder on the desktop:  For the User – you just have to know which user has access to which t-codes | |  | |
|  | In the SAP Easy Access window, enter VA01 as the t-code. <Enter>  Note – within this interface, at any time, from any window, if you enter /n as the t-code – the current process is ended and you get back to this screen | |  | |
|  | I strongly urge you to copy the “VA01 Steps.txt” file into the VM, and arrange your windows like this, so that you can easily see both the SAPGui and the values that you need.  Note that I entered 1711 (not 1710) so that UFT would learn this field. In SAP, the step count does NOT increment until you click Continue, then a step for each edit field is noted. | |  | |
|  | This is typical SAP. The area at the bottom is the “Status Bar” and will show warnings and errors.  Note that the UFT Step count incremented by 7.  This error was noted because 1711, is wrong, so correct it 1710, then  Click: Continue  (After stopping recording, change setting Sales Org value from 1711 to 1710, and delete the extra setting of value and Continue). | |  | |
|  | After correcting the Sales Organization and clicking Continue, you should see the  Create Standard Order Overview | |  | |
|  | Enter the Header info as shown.  Note: values in edit fields are CaSe inSENSItive.  You can <tab> or use the mouse to move to different fields.  After entering the header, Click in the area shown, so that you can enter the Item Details. | | |  | |
|  | Enter the Item details.  Click on Save | | |  | |
|  | In a few (or maybe even 10) seconds, the form should clear, and the Order Number should appear in the status bar | | |  | |
|  | In the Recording Toolbar, choose:  Standard Output Value  Click on the Order Number in the status bar.  Click OK | | |  | |
|  | This is one of the capabilities unique to SAPGui add-in.  Note that “allitems” is the string as displayed in the GUI.  But below that, note that there is item1 to item8. SAP parses the string, and knows which part is a fixed string that never changes (item1:=Standard Order), followed by a dynamic value (item2:=4236, which is what we want). This parsing of static/dynamic value continues for the entire string.  This eliminates the need to write VBScript code to parse the Status! Very cool!  I recommend that you click Modify | | |  | |
|  | DataTable -> Global sheet is fine  But change the Parameter Name to a good value  Click OK | | |  | |
|  | Create a checkpoint.  I did mine against the text, with a regular expression replacing the order number with \d+ | | |  | |
|  | Complete the business process and go back to the starting page.  In the Command Bar, type in:  /n<Enter>  You should be back on the starting Screen. | | |  | |
|  | Stop Recording.  Save Test.  Clean up test. In my case, I had:  I would delete the resizes steps and SetFocus steps  I would also delete the setting of the incorrect Sales Organization. After doing that, my code is in the next step. | SAPGuiSession("Session").SAPGuiWindow("SAP Easy Access").Resize 131,30  SAPGuiSession("Session\_2").SAPGuiWindow("SAP").Resize 131,31  SAPGuiSession("Session\_2").SAPGuiWindow("SAP").Close  SAPGuiSession("Session").SAPGuiWindow("SAP Easy Access").SAPGuiOKCode("OKCode").Set "VA01"  SAPGuiSession("Session").SAPGuiWindow("SAP Easy Access").SendKey ENTER  SAPGuiSession("Session").SAPGuiWindow("Create Sales Documents").SAPGuiEdit("\*Order Type").Set "OR"  SAPGuiSession("Session").SAPGuiWindow("Create Sales Documents").SAPGuiEdit("Sales Organization").Set "1711"  SAPGuiSession("Session").SAPGuiWindow("Create Sales Documents").SAPGuiEdit("Distribution Channel").Set "10"  SAPGuiSession("Session").SAPGuiWindow("Create Sales Documents").SAPGuiEdit("Division").Set "00"  SAPGuiSession("Session").SAPGuiWindow("Create Sales Documents").SAPGuiEdit("Division").SetFocus  SAPGuiSession("Session").SAPGuiWindow("Create Sales Documents").SendKey ENTER  SAPGuiSession("Session").SAPGuiWindow("Create Sales Documents").SAPGuiStatusBar("StatusBar").Sync  SAPGuiSession("Session").SAPGuiWindow("Create Sales Documents").SAPGuiEdit("Sales Organization").Set "1710"  SAPGuiSession("Session").SAPGuiWindow("Create Sales Documents").SendKey ENTER | | | |
|  | Here is my clean code.  Note: I used Edit -> Format-> Apply “With” to make the code fit better.  Save.  In the code, note that the item data appears in a table (All Items), and that first value is the row, second is the title of the column, third is the value.  In Fiori – this is different. The second value is also just a number – the number of the column! | | | With SAPGuiSession("Session")  With .SAPGuiWindow("SAP Easy Access")  .SAPGuiOKCode("OKCode").Set "VA01"  .SendKey ENTER  End With  With .SAPGuiWindow("Create Sales Documents")  .SAPGuiEdit("\*Order Type").Set "OR"  .SAPGuiEdit("Sales Organization").Set "1710"  .SAPGuiEdit("Distribution Channel").Set "10"  .SAPGuiEdit("Division").Set "00"  .SendKey ENTER  End With  With .SAPGuiWindow("Create Standard Order:")  .SAPGuiEdit("Cust. Reference").Set "450000019998"  .SAPGuiEdit("Cust. Ref. Date").Set "11/02/2021"  .SAPGuiEdit("Sold-To Party").Set "EWM17-CU02"  .SAPGuiEdit("Ship-To Party").Set "EWM17-CU02"  .SAPGuiTable("All Items").SetCellData 1,"Item","10"  .SAPGuiTable("All Items").SetCellData 1,"Material","ewms4-01"  .SAPGuiTable("All Items").SetCellData 1,"Order Quantity","1"  .SAPGuiTable("All Items").SetCellData 1,"Un","pc"  .SAPGuiTable("All Items").SelectCell 1,"Un"  .SendKey F11  .SAPGuiStatusBar("StatusBar").Sync  .SAPGuiStatusBar("StatusBar").Output CheckPoint("StatusBar")  .SAPGuiStatusBar("StatusBar").Check CheckPoint("StatusBar\_3")  .SAPGuiOKCode("OKCode").Set "/n"  .SendKey ENTER  End With  End With | |
|  | Run.  In this pop-up (which you will only see the first time you run), expand the Write results to and select Temporary.  When developing tests, there is no benefit of filling your disk with lots of run results. That is really for “production” | | | Graphical user interface, text, application, email  Description automatically generated | |
|  | Verify that you captured the order. | | |  | |
|  | Remember to switch to the Global sheet. | | |  | |

# VA01 Fiori Recording with AI

|  |  |  |  |
| --- | --- | --- | --- |
|  | Add a New GUI Test | | Name might be Fiori AI VA01 – or whatever you like |
|  | For an AI recording, the Record and Run Settings are not too important, except to set Record and Run on any open browser. (Also set SAP to **not** open a new SAPGUI)  Open a Chrome browser, and navigate to  <https://sap-hana.mfdemoportal.com:44300/sap/bc/ui2/flp#Shell-home>  (IE doesn’t work too well with Fiori) | | Remember – when working with AI:  Browser zoom should always be 100%  Browser window should always be full screen |
|  | Start Recording  This pop-up appears:  Open a Chrome browser, and navigate to (IE doesn’t work too well with Fiori and AI):  <https://sap-hana.mfdemoportal.com:44300/sap/bc/ui2/flp#Shell-home> | | Because you used the taskbar (or some other method) to launch  Chrome, the UFT Recording bar changes to the above. After  Navigating to Fiori, click on SELECT AGAIN |
|  | If this occurs, we must work around the bad certificate. (Any real customer will have valid certificates. This problem is rather unique to Nimbus)  AI recording should still be active:  Click: More information | | Graphical user interface, text, application, email  Description automatically generated |
|  | Click: Go on to the webpage….. | | Graphical user interface, text, application, email  Description automatically generated |
|  | In the Fiori login window, record entering the same info that was used with SAPGui.  Click: Log On | | Graphical user interface, application  Description automatically generated |
|  | Wait for the Magnifying Glass to appear in the upper right corner. This takes a while. Be patient  Click on it | | Text, icon  Description automatically generated |
|  | Note that AI recognizes this as a Search field.  Click on Search  An edit field should appear. Enter the t-Code VA01, then click on the arrow  Note that the recorded method is Search. For edit fields – it is “Type”. Search types in the value, but then also presses the <Enter> key! | |  |
|  | This page should appear.  Record a synchronization on Apps:  Click: Create Sales Orders.  (AI does not recognize multi-line text) | | Graphical user interface, text, application  Description automatically generated |
|  | Record the same business process, with the same data used for creating VA01 with SAPGui, **but**, with AI, you have to enter the values from the “bottom up”, i.e.  First enter the Division  Second enter the Distribution Channel  Third enter the Sales Organization | |  |
|  | This reverse order is necessary, because of the “pop-up” that appears with Fiori. This pop-up covers the edit field below it, and this makes AI replay fail!  Working bottom up eliminates this problem | |  |
|  | Record the  Sold-To Party  Ship-To Party  Cust. Reference  Cust. Ref. Date  Again with the same values as SAPGui, but “reverse order”  Don’t enter the actual item data (AI has a problem entering the data).  Click on Save | | Note in the following snapshot,  that SAP indeeds allows an  order to be created  without any item info. |
|  | Record a synchronization point on the Status Bar  **And** on the Green checkbox | |  |
|  | To get back to the “home screen”, click on Exit  Near the upper right-hand corner | |  |
|  | Add this “boiler plate” code to the top of your test.  This is why the Record/Run settings don’t matter. With AI, it usually works best to close all browsers, then launch your own in code. | while Browser("CreationTime:=0").Exist(0) 'Loop to close all open browsers  Browser("CreationTime:=0").Close  Wend  SystemUtil.Run "chrome.exe","","","",3 'launch the chrome  Set AppContext=Browser("CreationTime:=0") 'Set the variable for what application (in this case the browser) we are acting upon  wait 5 ‘ I found this wait to be necessary. You may not :-)  URL = "https://sap-hana.mfdemoportal.com:44300/sap/bc/ui2/flp?sap-client=100&sap-language=EN#Shell-home"  AppContext.ClearCache 'Clear the browser cache to ensure you're getting the latest forms from the application  AppContext.Navigate URL 'Navigate to the application URL  AppContext.Maximize 'Maximize the application to give the best chance that the fields will be visible on the screen  AppContext.Sync 'Wait for the browser to stop spinning  AIUtil.SetContext AppContext 'Tell the AI engine to point at the browser | |
|  | The security certificate error seems to occur during the first run every day, so add this conditional code to handle that | if AIUtil.FindTextBlock("More information").Exist then ' occurs sometimes because of unsigned certificate  AIUtil.FindTextBlock("More information").Click  AIUtil.FindTextBlock("Go on to the webpage (not recommended)").Click  wait 5 ' if the certificate warning appears, a hard wait after dismissing it seems to be necessary  End If | |
|  | As you should have seen  while recording – Search  Take a long time to appear. I like to use code like this for things that take a long time to appear, instead of a single .exist. | For xxx = 1 To 10 Step 1 ' this is my favorite way to wait for a long time  if AIUtil("search").Exist (xxx) then ' not that i wait for the loop index, so I wait longer every time through the loop  Exit for ‘ without this code, the entering of the value VA01 below  end if ‘ may fail due to a timeout  Next  AIUtil("search").Search "va01" ' note - this does not seem to work with IE,  but does indeed work fine with Chrome  If AIUtil.FindTextBlock("Apps").Exist (120) then ' you have to sync on the Apps text in the left pane - not on the actual Create Sales Order  'using .exist insteach of checkExists so that I can specify a wait larger than the global timeout value  End If  AIUtil.FindTextBlock("Create Sales Orders").Click | |
|  | To get the Order Number out of the status bar, the recorded code needs to be replaced with this. This is why I had you record on the Green check\_box!!!  Basically this says that the text we want is the text between the check\_box and the Save button | AIUtil("check\_box", micAnyText, micWithAnchorOnRight, AIUtil("button", "Save")).CheckExists True  StatusBarText = AIUtil.FindTextBlock(micAnyText,  micWithAnchorOnLeft,  AIUtil("check\_box",  micAnyText,  micWithAnchorOnRight,  AIUtil("button", "Save"))).GetText  StatusBarArray = Split(StatusBarText," ") ‘ standard VBScript code to split a string  print "The Order number is " & StatusBarArray(2) | |
|  | Save and Run | | The Order Number should be printed in the Output window |

# VA01 Fiori Recording with “Traditional” Object Identification

|  |  |  |
| --- | --- | --- |
|  | Create a new test. Name might be VA01-Fiori, or whatever you like |  |
|  | Start Recording.  In Run – Record Settings, in the SAP tab, set SAP to “Run on any open window”, so that UFT does not launch a new SAPGui window every run.  In Web, Enter the Fiori URL. This is bookmarked in Chrome Usually I recommend you record against IE, so that you can work with the Active Screen, but with Fiori, for reasons I don’t understand, site response seems **much** faster with Chrome.. The URL is:  <https://sap-hana.mfdemoportal.com:44300/sap/bc/ui2/flp#Shell-home>  Be sure to replay on the browser that you recorded with. Some description/navigation steps are browser specific.  Also, I recommend you record into 3 Actions:  Login  VA01  Logout | Graphical user interface, application, email  Description automatically generated |
|  | If this occurs, we must work around the bad certificate. (Any real customer will have valid certificates. This problem is rather unique to Nimbus)  Click: More information | Graphical user interface, text, application, email  Description automatically generated |
|  | Click: Go on to the webpage….. | Graphical user interface, text, application, email  Description automatically generated |
|  | Enter the same info that was used with SAPGui.  Click: Log On | Graphical user interface, application  Description automatically generated |
|  | Wait for the Magnifying Glass to appear in the upper right corner. This takes a while. Record adding a synchronization point on this object.  Click on it | Text, icon  Description automatically generated |
|  | This is just a comment.  After recording, you might find either of these property/value pairs used for the Search button. What you get seems to be dependent upon the detail of exactly where you clicked, and even seems different when you click on the Search the first time vs. the second time. On replay, just adjust the settings until replay works. | Graphical user interface, text, application, email  Description automatically generated  Graphical user interface, text, application  Description automatically generated |
|  | The Edit field should appear.  Note: Don’t click on Create Sales Orders – VA01. UFT doesn’t find it on replay. Instead  click again on the Magnifying (Search) icon again | A picture containing graphical user interface  Description automatically generated |
|  | This page should appear.  Click: Create Sales Orders VA01.  Note: this object also is not reliably found on replay. But, by spying on the object later, a completely reliable object can be learned. This will be discussed later. | Graphical user interface, text, application  Description automatically generated |
|  | Record the same business process, with the same data used for creating VA01 with SAPGui |  |
|  | Record a synchronization point on the Status Bar | Graphical user interface, application  Description automatically generated |
|  | (After recording, open the OR and change the innertext value to a regular expression, so that object recognition will work for any Order Number). | Graphical user interface, text, application, email  Description automatically generated |
|  | Note that in Fiori, the web page does not allow you to capture just the order number, like we could in SAPGui. You could do the typical UFT processing with VBScript, which is to capture the entire string, split on the ‘ – ‘ character, and then the number is in the [2] value (zero based index).  Easier though is to record  Click: Display | Graphical user interface, text, application  Description automatically generated |
|  | On the next screen, record  Standard Output Value on the Order Number (rename it :-).  In SAPGui, we used the t-code /n to “go home”. In Fiori, Click on the SAP Icon :-) | Graphical user interface, application  Description automatically generated |
|  | Create a call to New Action: Logout. Record the steps to logout. Stop recording. |  |
|  | When you recorded clicking on the VA01 tile, something like the code to the right was recorded. This does not replay reliably. To fix:   1. Delete this line. 2. Get Fiori to open the window with the tile: 3. Open Spy, Spy until the entire text is highlighted 4. Click on the object 5. In Object hierarchy, navigate up to the WebButton and accept this. 6. Add to OR 7. Add this object into the test script | Browser("SAP Fiori").Page("Logon").SAPUIList("SAPUIList").Select "#1"  Graphical user interface, application, Word  Description automatically generated  Graphical user interface, text, application  Description automatically generated |
|  | Now save and replay, but first – Examine your script.  Comment out “unneeded” steps.  Modify the OR for the status bar to use a regex so that it can be found with changing numbers (as discussed above)  Also modify the description for the Display Order Page | Graphical user interface, application  Description automatically generated |
|  | Here is my competed, working code (after “withing”)  I had to add the wait and .exists statements in order to replay consistently.  Here is the synch on the Search icon:  This is the object using the spy:  I had to add this code to get VA01 to reliably replay. Note: this is complicated enough, that if demoing to a customer, I would create all of the code up to this point already within my test, then  Here is the code that actually created the Sales Order which is very straight forward.  other “security” pop-ups that might appear: | 'if IE - note – “This site isn’t secure” may or may not appear  With Browser("SAP Fiori")  if .Page("This site isn’t secure").Link("More information").Exist (3) then  .Page("This site isn’t secure").Link("More information").Click  .Page("This site isn’t secure").Link("Go on to the webpage (not").Click  End If  .Page("Create Sales Documents").SAPUIButton("Search").WaitProperty "visible", True, 20000  .Page("Logon").SAPUIButton("Search").Click  wait 5  .Page("Logon").SAPUITextEdit("Search").Set "va01"  wait 2  .Page("Logon").SAPUIButton("Search with Value in EditField").Click  if .Page("Logon").WebButton("Create Sales OrdersVA01").Exist (30) then  wait 2  end if  For xxx = 1 To 5 Step 1 ' I seem to need this to reliably replay. You may not  if not .Page("Create Sales Documents").SAPFrame("Create Sales Documents").SAPEdit("Order Type").Exist (1) then  wait 3  End If  If .Page("Create Sales Documents").SAPFrame("Create Sales Documents").SAPEdit("Order Type").Exist (2) Then  Exit for  End If  .Page("Logon").WebButton("Create Sales OrdersVA01").Click  wait xxx  Next  .Page("Create Sales Documents").SAPFrame("Create Sales Documents").SAPEdit("Order Type").Set "or"  .Page("Create Sales Documents").SAPFrame("Create Sales Documents").SAPEdit("Sales Organization").Set "1710"  .Page("Create Sales Documents").SAPFrame("Create Sales Documents").SAPEdit("Distribution Channel").Set "10"  .Page("Create Sales Documents").SAPFrame("Create Sales Documents").SAPEdit("Division").Set "00"  .Page("Create Sales Documents").SAPFrame("Create Sales Documents").SAPButton("Continue").Click  .Page("Create Sales Documents").SAPFrame("Create Standard Order:").SAPEdit("Sold-To Party").Set "EWM17-CU02"  .Page("Create Sales Documents").SAPFrame("Create Standard Order:").SAPEdit("Ship-To Party").Set "EWM17-CU02"  .Page("Create Sales Documents").SAPFrame("Create Standard Order:").SAPEdit("Cust. Reference").Set "450000019998"  .Page("Create Sales Documents").SAPFrame("Create Standard Order:").SAPEdit("Cust. Ref. Date").Set "11/02/2021"  .Page("Create Sales Documents").SAPFrame("Create Standard Order:").SAPTable("All Items").SetCellData 2,2,"10" ' item  .Page("Create Sales Documents").SAPFrame("Create Standard Order:").SAPTable("All Items").SetCellData 2,3,"EWMS4-01" ' material  .Page("Create Sales Documents").SAPFrame("Create Standard Order:").SAPTable("All Items").SetCellData 2,5,"1" ' quantity  .Page("Create Sales Documents").SAPFrame("Create Standard Order:").SAPTable("All Items").SetCellData 2,6,"PC" 'unit  .Page("Create Sales Documents").SAPFrame("Create Standard Order:").SAPButton("Save").Click  if .Page("Create Sales Documents").Frame("Frame").WebElement("ContentBlocked").Exist(5) then  .Page("Create Sales Documents").SAPUIButton("Back").Click  end if  if .Page("Review Availability Check").Frame("itshtmlvwrfn101").SAPUIButton("Apply").Exist (5) then  .Page("Review Availability Check").Frame("itshtmlvwrfn101").SAPUIButton("Apply").Click  end if  .Page("Create Sales Documents").SAPFrame("Create Standard Order:").WebElement("wnd[0]/sbar\_msg-txt").WaitProperty "visible", True, 10000  .Page("Create Sales Documents").SAPFrame("Create Standard Order:").SAPButton("Display").Click  .Page("Create Sales Documents").SAPFrame("Display Standard Order").SAPEdit("Standard Order").Output CheckPoint("Standard Order")  .Page("Create Sales Documents").Image("Company Logo").Click  .Dialog("Message from webpage").WinButton("OK").Click  End With |

# Add VA02 to SAPGui

|  |  |  |
| --- | --- | --- |
|  | Now let’s modify this SAPGui script to modify the newly created order, using VA02 |  |
|  | Rename Action1 to VA01 (Just for readability/clearness” |  |
|  | Right Click on the Test  Choose Add -> Call to New Action |  |
|  | Rename the new action to VA02  Be sure that Location is “At the end of the test” |  |
|  | Highlight the VA02 action  Before you start recording, be sure that you know the name of the Order that was just created from your VA01 run.  Start Recording |  |
|  | Enter VA02 in the Command Bar.  Note that SAP will populate the Order Number just created into the Order Number field, but we don’t want that. We want to set it. So do the trick of entering a number one less than the desired number.  Type: <Enter> |  |
|  | We are going to increment the existing quantity to show the power of UFT.  Right-click on the quantity, create an Output value. Note that this is a table, not an edit field. |  |
|  | Choose All Rows |  |
|  | We want only the Order Quantity value. So click on that cell,  then click on the Check – to select this row/column for output  Best practice – change the Parameter Name. |  |
|  | You should still be recording.  In the SAPGui, click on the quantity, change the value.  Click Save |  |
|  | Now let’s do a dynamic checkpoint.  Record creating a Standard Checkpoint on the Status Bar  Choose only the text attribute  Configure value to use a Parameter, not a Constant. (don’t worry too much about the string in the Parameter field. We will be changing it later).  Click OK  Finish VA02 – enter \n <Enter> in the Command Bar.  Stop recording. |  |
|  | Open the VA02 Local Object Repository (OR).  Select your Checkpoint (your name may differ)  Click on the icon to open Parameter properties |  |
|  | The prefix should be:  “Standard Order “ – case sensitive, note the ‘ ‘ at the end  Suffix should be:  “ has been saved.” – note the ‘ ‘ at beginning  Click OK |  |
|  | Here is a nice tip for working with SAP GUI.  Go to Edit -> Format -> Apply “With” to Script |  |
|  | Here is the result – makes the lines much shorter.  This is my final code. After Clean up.  I had recorded setting a specific Order Number. This was changed to the parameter from the DataTable (that is populated in VA01)  Your name may be different.  This Output, although the name is “All Items”, that is the name of the Table. In the output, it is configured to capture the current quantity, only  Here, I am reading the captured quantity, incrementing by 1, and setting the value  And this is the dynamic CheckPoint as discussed/shown above | With SAPGuiSession("Session")  With .SAPGuiWindow("SAP Easy Access")  .SAPGuiOKCode("OKCode").Set "VA02"  .SendKey ENTER  End With  With .SAPGuiWindow("Change Sales Documents").SAPGuiEdit("Order") \_  .Set DataTable("StatusBar\_item2\_out", dtGlobalSheet)  .SendKey ENTER  End With  With .SAPGuiWindow("Change Standard Order")  .SAPGuiTable("All Items").Output CheckPoint("All Items")  .SAPGuiTable("All Items").SetCellData 1, \_  "Order Quantity", DataTable.value("Origin QTY") +1  .SAPGuiTable("All Items").SelectCell 1,"Un"  .SendKey F11  End With  With .SAPGuiWindow("Change Sales Documents")  .SAPGuiStatusBar("StatusBar").Sync  .SAPGuiStatusBar("StatusBar").Check \_  CheckPoint("StatusBar")  .SAPGuiOKCode("OKCode").Set "/n"  .SendKey ENTER  End With  End With |
|  | Run. If there are any errors in the Test Results – debug until there are no errors. | If you get stuck, feel free to reach out to me.  This concludes our work with SAPGui. Onto Web (Fiori) |

# Add VA02 to Fiori (Traditional)

|  |  |  |
| --- | --- | --- |
|  | Now, like we did for SAPGui, lets add VA02 |  |
|  | Add a call to New Action, VA02, and record VA02  We could navigate as we did for VA01 by searching, but if you know exactly where to find it in Fiori, there is another possibility.  From the Fiori home page:  Click: Sales Order tab  Click: Change Sales Orders VA02 tile | Table  Description automatically generated |
|  | In VA02, enter one of the Order Numbers from running VA01.  Click: Continue | Graphical user interface, application  Description automatically generated |
|  | Using whatever technique you want (record a sync, output, use spy or OR) Add the table that has the order details to the OR.  Then capture the existing Order Quantity with a .SAPTable("All Items").GetCellData (2,5) into a VBScript variable  Note: 2 for second row (row 1 is the header), 5 the 5th column (As mentioned above – cell identification is not nearly as “user friendly” as SAPGui. | Graphical user interface, text, application  Description automatically generated |
|  | Use a .SetCellData(2,5) to update the value.  Click: Save  You might get this pop-up:  If so  Click: Apply | Graphical user interface, text, application, email  Description automatically generated |
|  | Record a synchronization on the Status Bar  Click: Continue | Graphical user interface, text, application, email  Description automatically generated |
|  | To verify that the new value was set:  Re-open the same order with VA02.  As above, capture the new Order Quantity, and create a custom checkpoint with code like this:  If nItems = cint(newVal) Then  reporter.ReportEvent micPass, "nValue Check", "Values updated successfully"  else  reporter.ReportEvent micFail, "nValue Check", "Values failed! Expected: " &nitems& " and found: " &newVal  End If | Graphical user interface, application  Description automatically generated  Note the  Cint() function call.  Without this Cint call, VBScript compares the  numeric(nItems)  with the string(newVal),  and will always evaluate to false. |
|  | Similar to the end of VA01,  Click: SAP Icon  To go back to Home page.  Dismiss any pop-ups that might occur  Stop Recording |  |
|  | Before Replaying, GoTo View->Test Flow  Make sure that the flow looks like this screen, i.e.,  Login  VA01  VA02  Logout | Graphical user interface  Description automatically generated |
|  | Similar to the VA01 link, the VA02 link object that was recorded is not very reliable. The OR shows that it was recorded as the 39th link in the table! Fix this as follows. | Graphical user interface, text, application, email  Description automatically generated |
|  | In Fiori, navigate back to the Sales Order Tab  Open the OR -> Spy  Spy on the VA02 button. When the entire button is highlighted:  Click on it  Add to repository  Add clicking on this new object to your script, in place of clicking on the recorded object | Graphical user interface, application  Description automatically generatedText  Description automatically generated with low confidence |
|  | Save  Replay  Look in the test results. You will find many instances of Smart Identification triggering within the VA02 Action. | This is because in VA02, the pages have the  Order Number as part of the description.  I recommend that after replaying, find all  occurrences of “Smart Identification”,  and modify those descriptions to  use regular expressions, ignoring the specifics of the order number. |

Congratulations! This completes the using UFT with SAP exercises. You are not an expert, but you are now a pretty component user of UFT against SAP.

If using github, I recommend you use git bash to add all files to the repository/commit/push. This way you will have this working code for future reference.