

SAP Automation - UiPath Academy

URL	2
Learning Objectives	2
Technical Prerequisites	2
Best Practices/Notes	3
SAP RPA and SAP Testing	5
SAP Scripting Interface	7
Certification	7
Supported SAP WinGUI versions	7
Basic SAP Window Elements	10
SAP user Interface	10
UiPath studio specific SAP activities	11
SAP configuration steps	11
Connecting to SAP WinGUI and Calling Transactions	11
Demo-Login, Logon & Call Transaction	12
Demo-Using Secure password to login to SAP	15
Creating a Purchase Order	18
Demo: Creating a purchase order	18
Importing Data From Excel to SAP	26
Creating an Invoice	26
Common Table Operations	30
Data Scraping in SAP Tables	32
Picture on screen and SAP calendar	35
SAP Context Menu and Pop-Ups	39

URL

<https://docs.uipath.com/studio/v2020.4/docs/about-sap-wingui-automation>

<https://docs.uipath.com/studio/v2020.4/docs/sap-configuration-steps>

<https://docs.uipath.com/studio/v2020.4/docs/supported-sap-elements>

Learning Objectives

At the end of this course you should be able to:

- Describe the differences and similarities between Robotic Process Automation and Test Automation;
- Prepare your SAP system for stable and reliable automation over the SAP Scripting interface;
- Perform different SAP automation techniques by following the best practices for SAP WinGUI automation;
- Automate any process related to SAP Robotic Process Automation, as well as SAP Test Automation, and work with any SAP element in UiPath Studio.

Technical Prerequisites

AP WinGUI 7.40 or higher;

UiPath Studio 19.10 or higher;

- **SAP WinGUI 7.40** or higher;
- **UiPath Studio 19.10** or higher;

The following UiPath Studio Packages Installed:

- **UI Automation Activity Package 20.4**;
- **UiPath Credentials Activities 1.0** or higher;
- **UiPath Testing Activity Package 1.0** or higher.

Best Practices/Notes

- RPA operates in **stable production environments**.
- Software Testing operates in **dynamic development and testing environments**.
- The point in which the **SAP Process Automation** and **SAP Testing Automation** disciplines meet is **Automation**.
- Both software testing, in our case **SAP Testing**, and **SAP RPA** are confronted with the task of **creating and maintaining resilient automation**.
- The **special SAP Logon activity** has been created in UiPath Studio to automate the process of logging in by one or multiple users at the same time.
- There are several methods of securely storing passwords, e.g. using Windows Credential Manager or storing the credentials as assets in Orchestrator.
- The **Get Secure Credential activity** can be used for automating the process of logging in with credentials stored in Windows Credential Manager.
- Calling a transaction in SAP can be done with a simple Type into activity, but using the **Call Transaction activity** is a more reliable option.
- Wildcards are useful when selectors frequently change - the screen ID from SAP is a good example. Adding a wildcard instead of the screen ID within the selector ensures that the selector will remain stable and valid throughout the automation.
- By using the **Select Item activity** on a tab within SAP, all of the individual tab options will become selectable directly from UiPath Studio.
- The **Services for Object button** can be fully manipulated through the use of the **Click Toolbar Button activity**.
- Through the use of the **SAP Menu activity**, the full SAP menu can be called, thus gaining access to each menu item.
- The **Read Statusbar activity** allows checking the statusbar properties and values.
- The **Verify Control Attribute activity** allows users to compare the real value of an attribute against the expected value for different

elements in the system. This is a Verification activity, part of the Testing Activities Package.

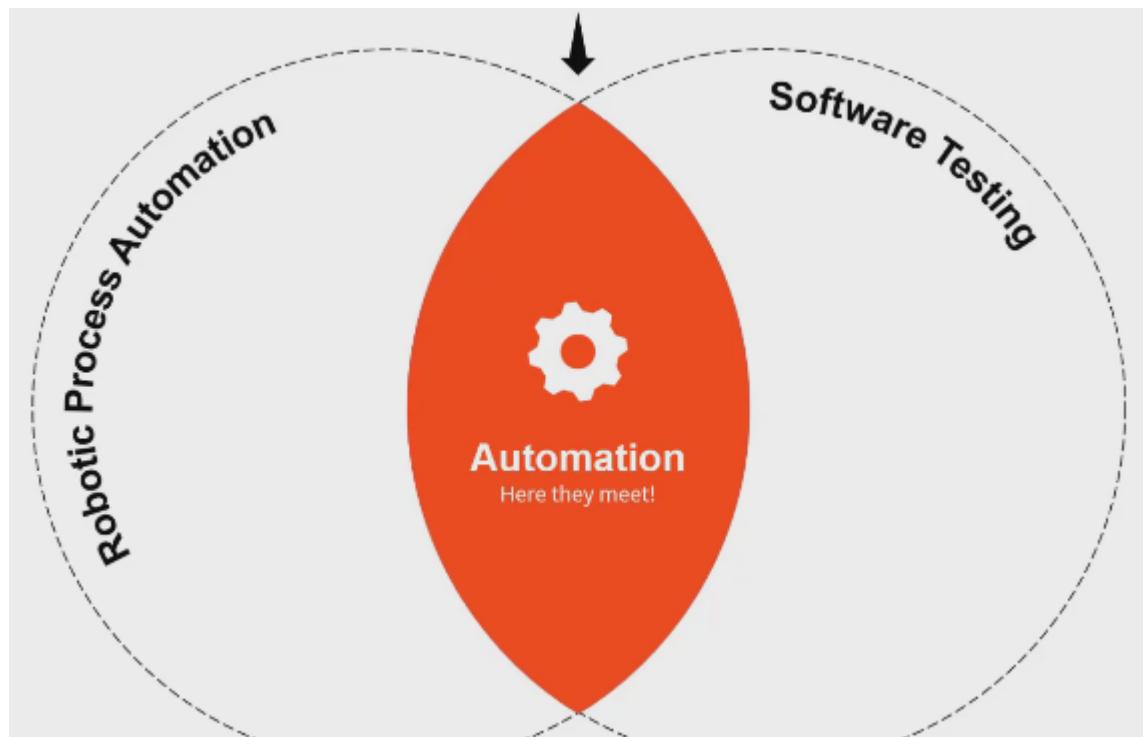
- In software testing, the process of verifying if software meets certain requirements and behaves as expected is the most important aspect. Test cases that do not perform any sort of verification add no value, as you would never know if they passed or failed, and if the system under the test behaves as expected.
- Best Practice is to clear the field (empty field) before entering any data in to the sap fields
- Reusing workflow components can be simply achieved through the use of the **copy and paste function** within UiPath Studio.
- In order to **add data from Excel to SAP**, the **Excel Application Scope** must be added to the workflow and the Excel file needs to be indicated.
- When **iterating through each Excel row** and then entering the data in SAP tables, a **For Each Row activity** should be used together with a **counter variable**.
- When **entering data into SAP fields**, it is a good practice to **clear the fields** if they are already populated. This can be achieved by checking the **empty field box** from the Type into activity Property panel. In addition, this can be achieved for the **entire project** and not just particular activities. The setup can be done from **Project Setting > UIAutomation > Keyboard Events**.
- The **SAP toolbar button menu** can be steered by using the **Click Toolbar Button activity**. This gains access to all of its **buttons and sub-menus of those buttons**.
- Even though the **SAP Tree** can be huge and dynamic, accessing a node or an item inside the tree can be done by using the **SAP Expand Tree activity**. It will **expand the parent tree** to an **active node** or an **active item**.
- Through the use of the **UI Explorer**, SAP elements, such as the **Traffic Light** (Ampel), can be identified and used as **indicators** to perform certain **branching activities**. In our case, we could tell if the balance was equal to zero or not, which in turn dictated if we could post the invoice. Depending on the elements state, a branching scenario can be formed.
- The **Verify Control Attribute activity** can be used to verify any kind of available element information recognized by UI Explorer.
- Simple **click activities** can manipulate the **elements** within a SAP Table.

- Once a **table element** is indicated, details such as **row** and **column** can be found in the **selector**.
- The **Click Toolbar Button activity** is used when dealing with **complex table menu buttons** that contain drop down options which could for example **set filters, extract data or change the table view**.
- To **select multiple rows or columns** at the same time, the Control key needs to be pressed while performing the selections. This can be achieved by selecting the Control key option from the **key modifiers** section within the **activity properties**. Once multiple rows or columns are selected, further **data manipulation** can be done prior to **data extraction**.
- When using Data Scraping the full table needs to be indicated.
- The desired **column name** (or source) can be indicated at extraction through the **XAML editor** within the **Data Scraping activity**. The column name can be either the **longest, shortest** or the **displayed** name at extraction time.
- Data can be **extracted** from SAP in **other mediums** besides Excel as well
- When using the **Click on Picture** activity, only the **SAP screen** in which the picture resides **must be indicated**. Afterwards, the **type of interaction** such as single or double clicking **must be indicated**. The interaction will be performed **regardless of the image's position** within the previously indicate screen.
- Through the use of the **Select Dates in Calendar** activity, **individual days, day ranges** or **specific weeks** can be selected once the SAP calendar has been indicated.
- When facing **pop-ups** or **context menus**, the simple **Click activity** is enough to manipulate them. This will become possible once the **identification framework** is changed by **pressing F4 when indicating the selector**.
- The 3 available **identification frameworks** are the default **SAP Scripting Interface, AA and UIA**.
- A common situation is selecting an **option nested in the drop down menu** of a button that also needs to be clicked. By **pressing F2**, the selection will be **delayed for 3 seconds, allowing to select the nested item** after expanding the drop-down.

SAP RPA and SAP Testing

Differences and similarities between SAP Process Automation and SAP Testing.

RPA	Software Testing
Cost reduction activity	Risk mitigation activity
Robotic Process Automation	Software Testing
Automate Human Activities	Automate System Checks
Event-Driven	Build-Driven
Production Environment	Development Environment
Stable Environment	Dynamic Environment
Low Change Frequency	High Change Frequency
Low Automation Artifact Quantity	High Automation Artifact Quantity
Uniform Application Landscape	Diverse Application Landscape
Strong Focus on Legacy Apps	Strong Focus on Modern Apps
Measures FTEs	Evaluates Software Quality
Uses Production Data	Uses Synthetic Test Data



Software testing is a risk mitigation activity. Risk mitigation activity is accelerated but not enabled by automation. In contrast, RPA is a cost reduction activity that is directly enabled by automation this means RPA cannot exist without automation but software testing can.

RPA operates in stable production environments where applications do not change often. Software testing operates in dynamic development environment where application receives updates frequently on an hourly or daily basis. This implies that the maintenance effort related to the test case automation is much higher than the maintenance effort related to rpa from a change frequency perspective.

In addition, typical software projects contain atleast several hundreds and up to multiple thousands of automated test cases where typical RPA projects contain 10s up to hundreds automated business processes. Once again this implies that the maintenance effort related to automate test cases is much higher than the maintenance effort related to automate business process from automation artifact quantity perspective.

Basically, the higher the change frequency, the higher the maintenance effort. To summarise, maintenance is a big challenge in RPA and even bigger challenge in software testing. The good news is that are UiPath products help overcome these maintenance challenges in various ways. The takeaway of this presentation would be that in both disciplines namely RPA and software testing, it is fairly easy to create automation but it is hard to maintain it.

The point in which these 2 disciplines meet is automation. Both RPA and software testing, in our case SAP testing, are confronted with the task of creating and maintaining a silent automation. During our UiPath studio SAP automation course, we will highlight automation techniques and best practices for automating SAP with UI path studio that are applicable for both words SAP robotic process automation as well as SAP test automation therefore lets

agree on the common word in which will be SAP automation weather mentioning SAP RPA or SAP testing.

SAP Scripting Interface

SAP WinGUI is the most commonly used interface for accessing SAP functionality in SAP applications, such as SAP ERP, SAP ECC, SAP S/4 HANA, and the family of SAP Business Suite applications, such as SAP BI/BW, SAP CRM, SAP SCM, SAP PLM, and others.

UI path studio and robots require SAP WinGUI scripting to be enabled on the local machine as well as on the server side.

Certification

UiPath has certified integration with SAP S/4 HANA and SAP ECC for SAP WinGUI automation.

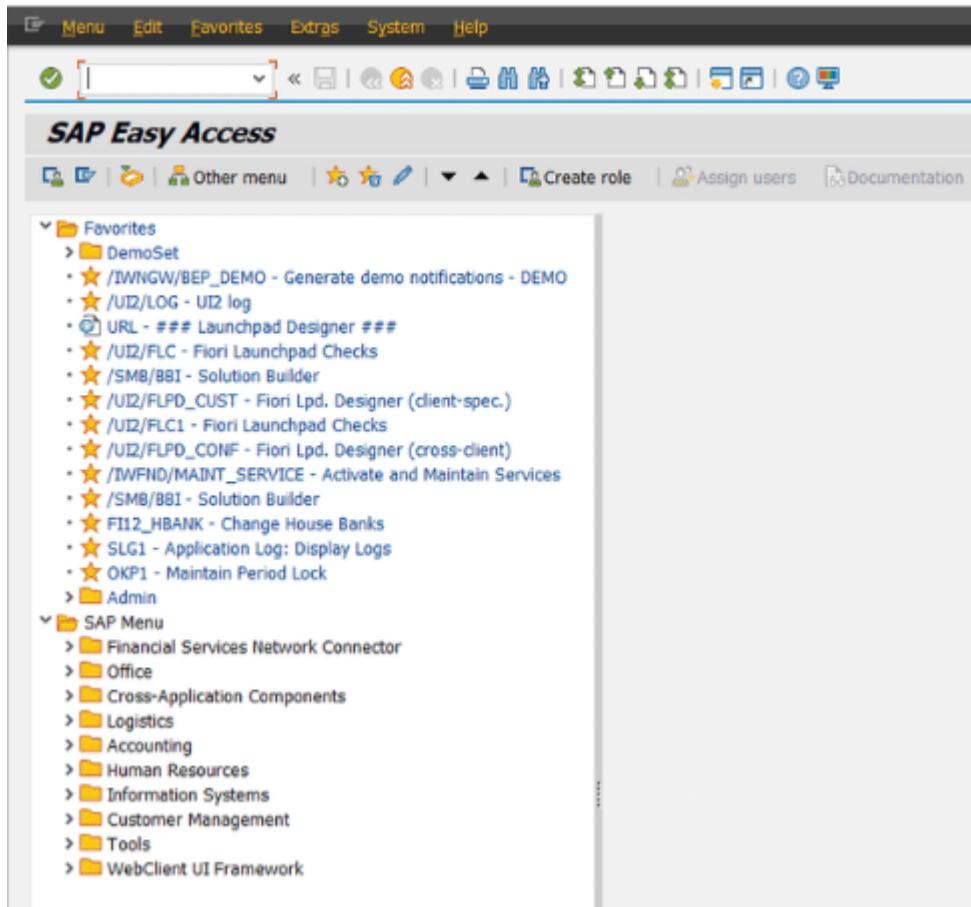
Supported SAP WinGUI versions

The below versions of SAP WinGUI are supported for Windows 7 and Windows 10, both architecture types (32-bit and 64-bit). **SAP Logon** and **SAP Logon Pad** are supported on all versions:

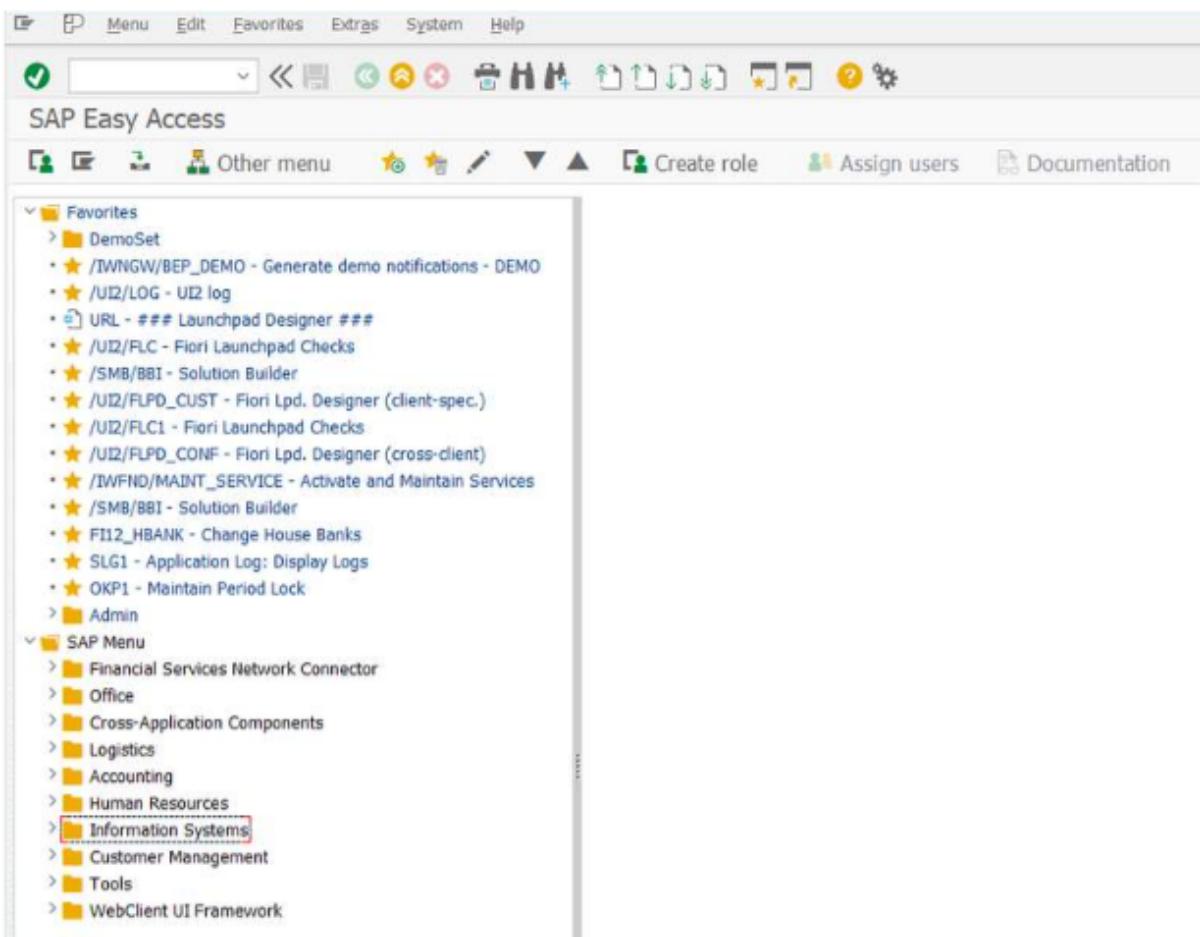
- SAP WinGUI 7.40
- SAP WinGUI 7.50
- SAP WinGUI 7.60

All SAP themes such as Belize, Blue Crystal, or Corbu can be used for SAP automation.

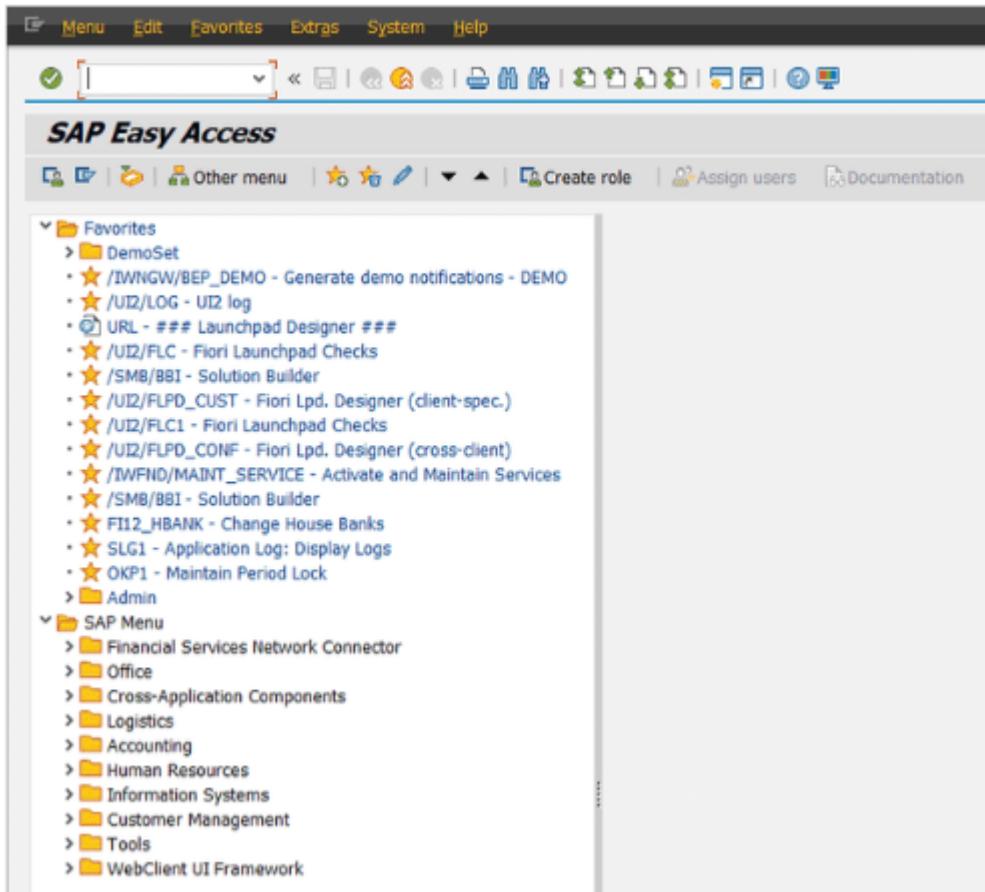
Belize Theme:



Blue Crystal theme



Corbu Theme

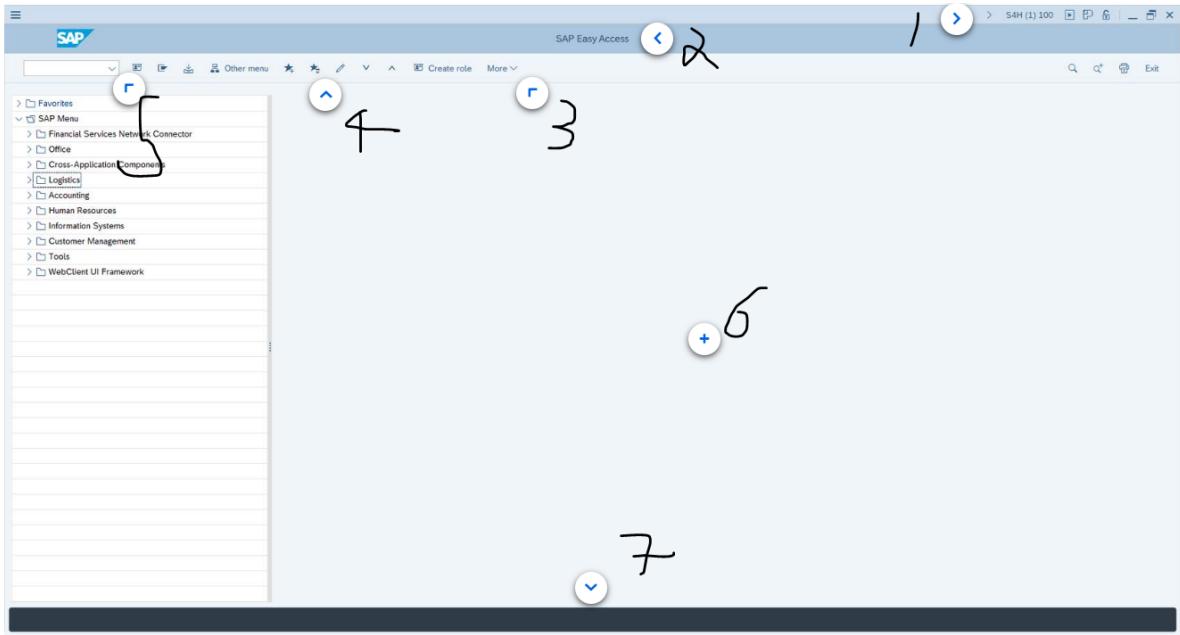


Basic SAP Window Elements

SAP user Interface

In order to work and create automation projects through UI path studio, understanding the SAP user interface and layout is crucial.

1. System information area: displays information about the current sap system and transaction or application.
2. Title bar: contains the name of the currently displayed application
3. Menu bar: contains different functions depending on the current user activities
4. Standard toolbar: consists of a range of icons with general GUI functions
5. Command field: used to enter a transaction code
6. Screen area: main user activity area
7. SAP Status bar: found at the bottom of the SAP screen. it displays important system



UiPath studio specific SAP activities

In addition to the activities from the UI automation activity package some special SAP activities have been introduced.

1. SAP call transaction
2. SAP click picture on screen
3. SAP click toolbar button
4. SAP expand tree
5. SAP login
6. SAP logon
7. SAP read statusbar
8. SAP select dates in calendar
9. SAP select menu item

SAP configuration steps

1. Enable SAP GUI Scripting
2. Enable a high speed connection for your SAP
3. Enable modal dialogue capabilities

Connecting to SAP WinGUI and Calling Transactions

1. Automate the process of login into your SAP client using securely stored credentials

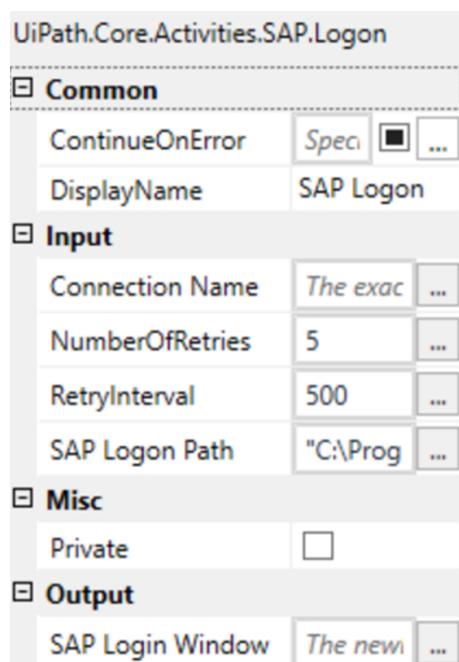
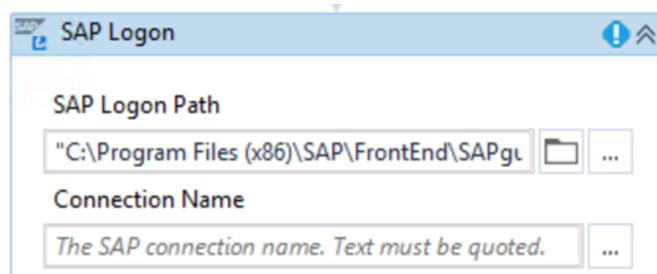
2. Automate the activity of calling transactions once logged in

Demo-Login, Logon & Call Transaction

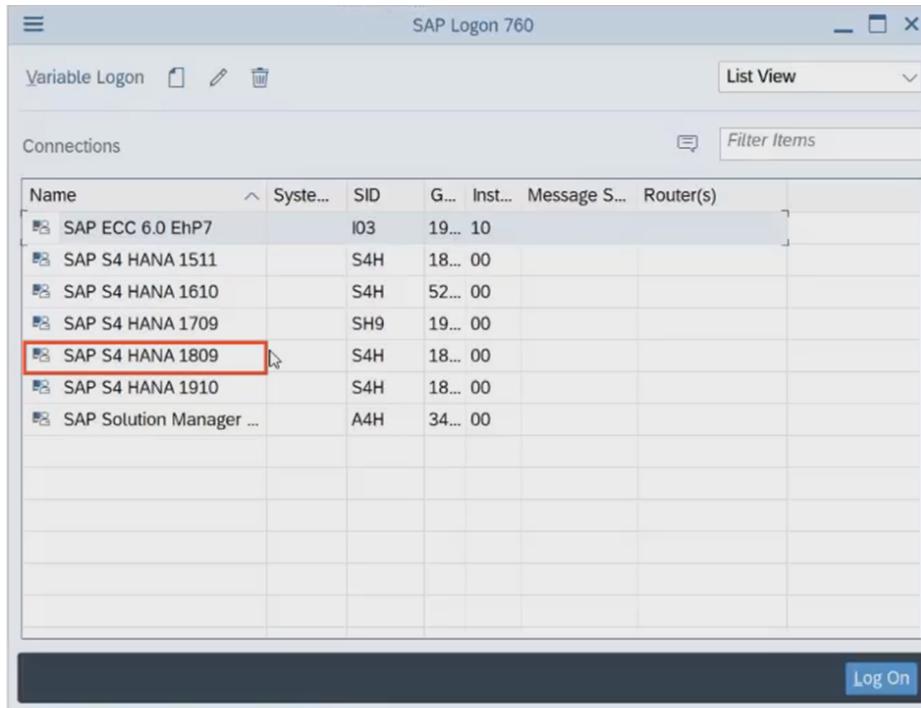
How to automate the login to SAP activity, using single and multiple users at the same time, then call a transaction

To log on to SAP, users need to firstly select a system from the SAP logon window which can potentially contain hundreds of systems. The appearance of the window can be different depending on the view the user is using but it does not affect the way in which the automation needs to be configured. So, let's start creating our SAP automation in UiPath Studio.

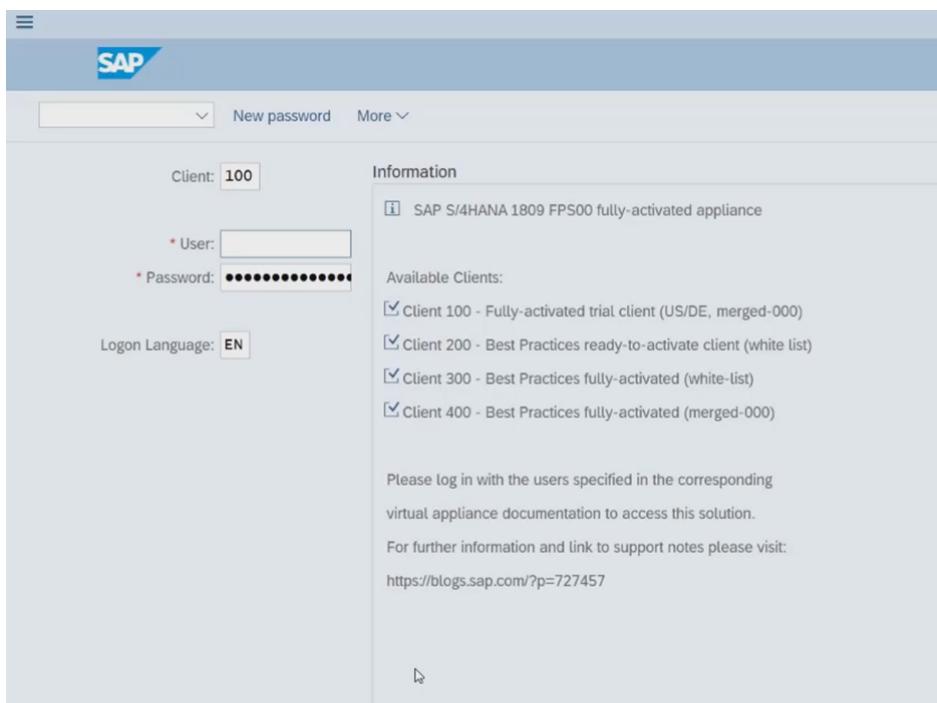
There's a special SAP activity designed for this purpose, and we can find it by navigating the activity panel to the SAP activity list or by searching for it in the same panel. We will add our SAP Logon activity, and we can see that the default path for our "saplogon.exe" file is already populated. In case SAP was installed in a different location, the "logon exe" file can be browsed for and have its location added here.



In the connection name field, the unique name of this system will need to be added as it is displayed in the connection list.

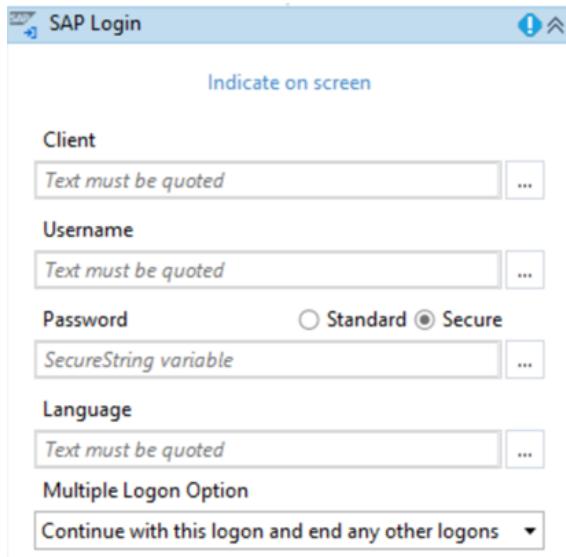


We will need to close the logon window and then run the workflow. We can see that the studio opened the login window and entered the system name. Now we are in the login screen where we will need to further add necessary information.

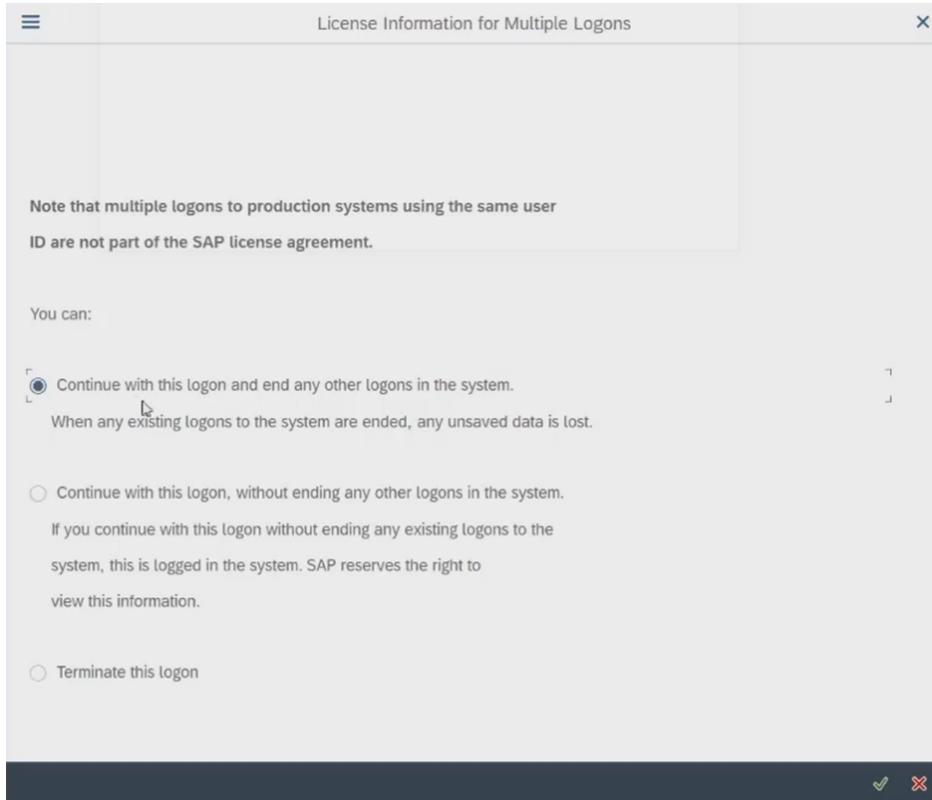


These fields can be filled in by using type into activities or we could use the special SAP activity which is login activity. By indicating the full login screen, we will gain access to all the

input fields. Next, we will add the client details, the username which will be demo in our case, the password and lastly the language English. Let's close all the SAP windows and run the workflow again to see what happens.

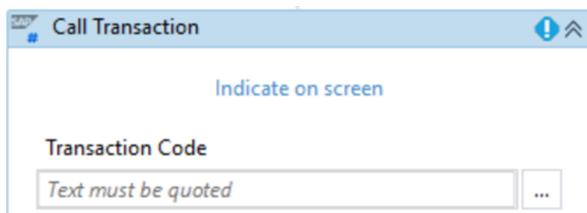


Login and logon details are added. Additionally, we are providing users the possibility to choose what happens in case a popup with multiple login options gets displayed. This popup shows up when users want to login twice in the same system using the same user. Let's perform this manually and see how the popup looks like.



We can either continue with this logon and end all other logons, continue with the same logon without sending any other logons in the system or terminate this logon. All 3 options are available directly from the SAP activity and you can easily use them from your automation project in the activity that is SAP Logon.

After logging in, the next logical step in SAP would be to call a new transaction. Transactions can be called either by using a simple type into activity in the transaction field or through the recommended way by using the special activity “Call transaction”. After adding this activity, we need to indicate the whole screen and if we go deeper within the selector, we can remove the title by unchecking it. By doing this, we ensure that the workflow will be reusable for all such screens. Let's also fill in the transaction code field by adding the transaction code. The transaction code must be added exactly as it is before we run the full automation.

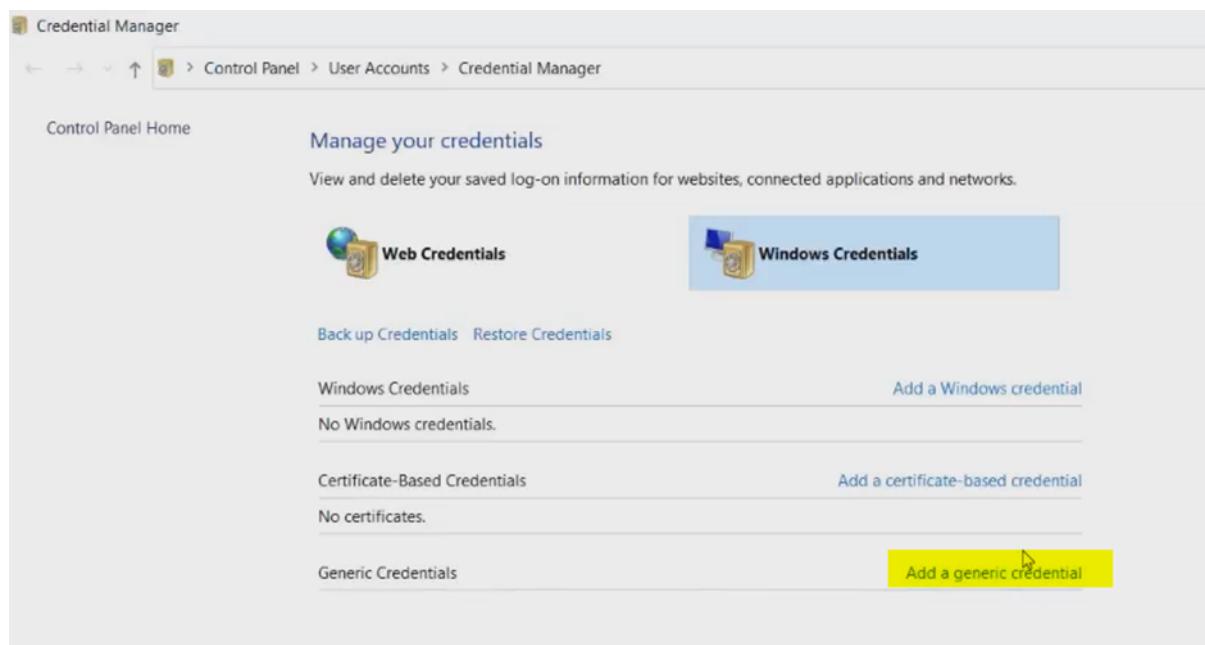


Basically what happens in the background when using the call transaction activity is sending a command into SAP that looks like this /nVA01 (transaction code with/n) which means that opening a transactions in the same window, entering the transaction number and then pressing enter. This optimization helps users to be more efficient when automating scenarios involving these activities without needing to think about all these details.

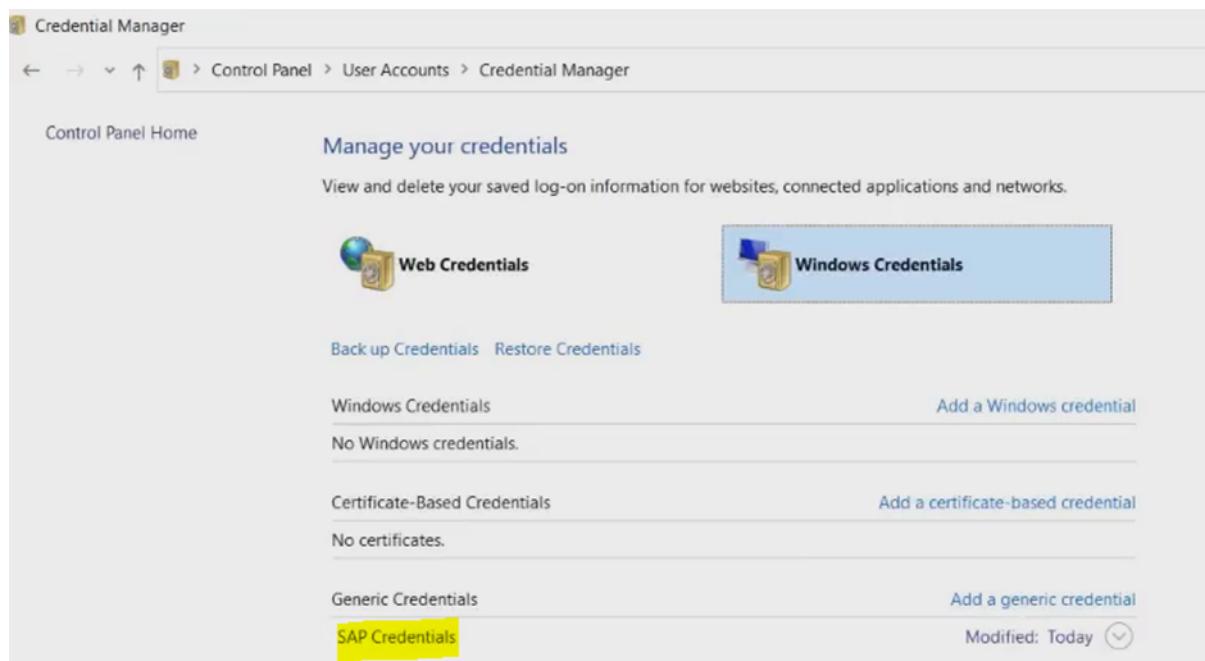
Demo-Using Secure password to login to SAP

There are several methods of storing credentials in RPA and business scenarios. The most common examples would be the windows credential manager storage and the orchestrator database.

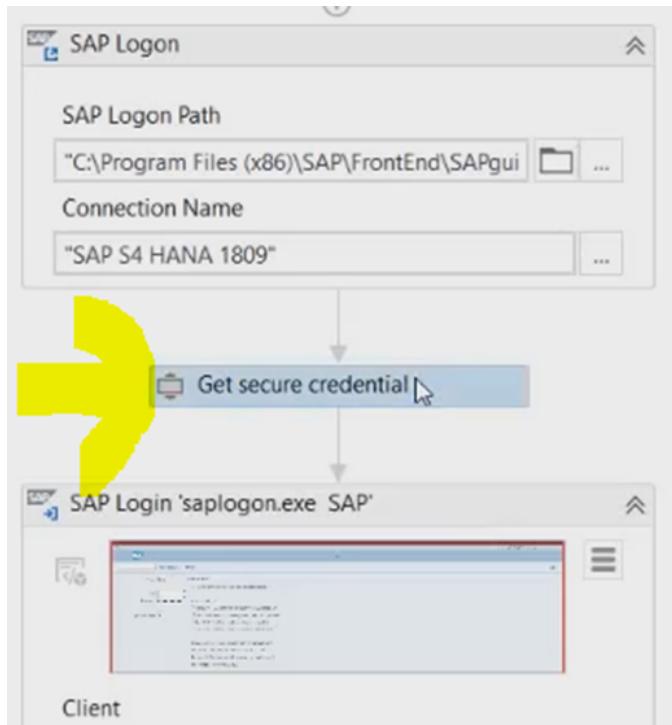
To demonstrate the first method, we will add a “Get secure credential” activity. This one can be found in the activities panel, under system, more specifically the credentials section. With this activity, our user ID and password will be retrieved from the windows credential manager. Let's open the windows credential manager and create a new generic credential.



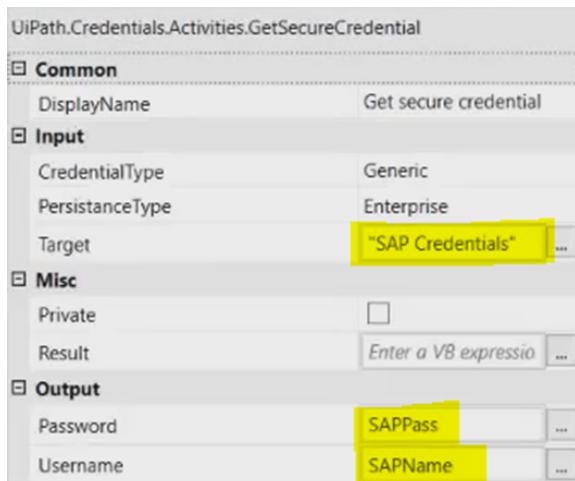
We can see the user has been created as below. We can now configure our activity in studio.



Use the below activity and specify parameters



We just need to mention the credential name we just created in Windows credential manager and create 2 outputs (for password and username)



Our automation will connect to the windows credential manager at runtime and use the named credentials stored now we can replace the username and password data from the sap login activity with the variables that we just created let's add the username first. Notice that the data type of the password variable is set to "SecureString". This will prevent having the actual password displayed in the workflow or in any other panels. We are updating the username and the password variable in the SAP login activity. With this approach we successfully automated the process of logging in by having the password securely sold for the other method that we have mentioned that is orchestrator we can visit the orchestrator for developer skills in uipath academy.

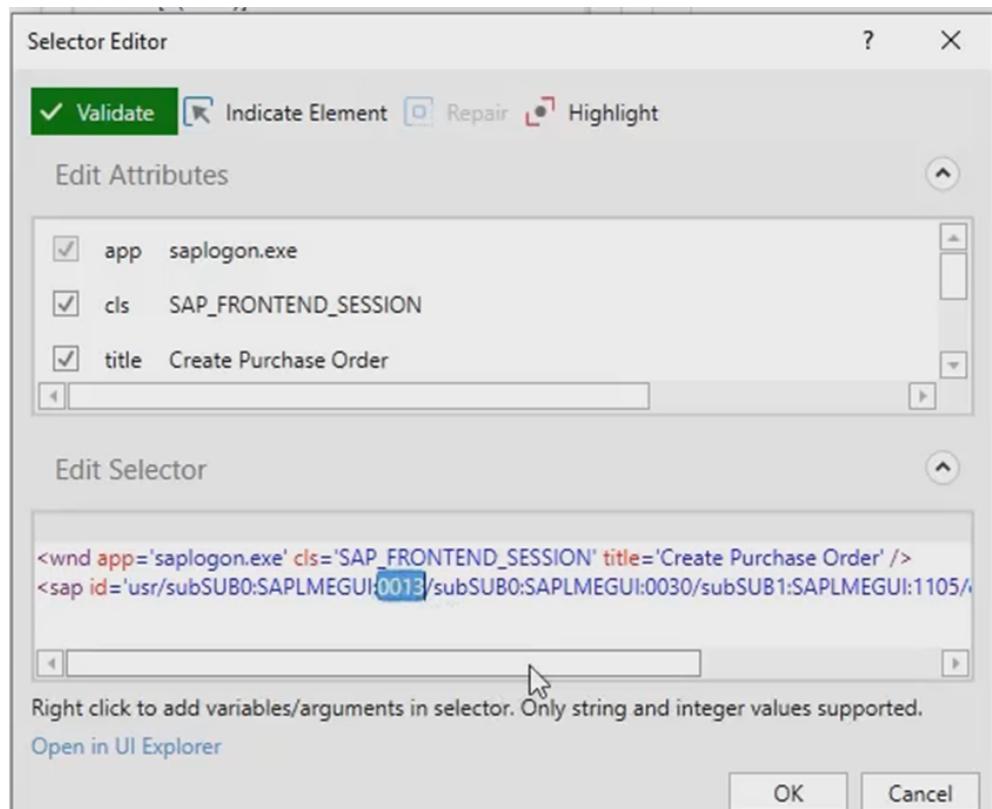
Creating a Purchase Order

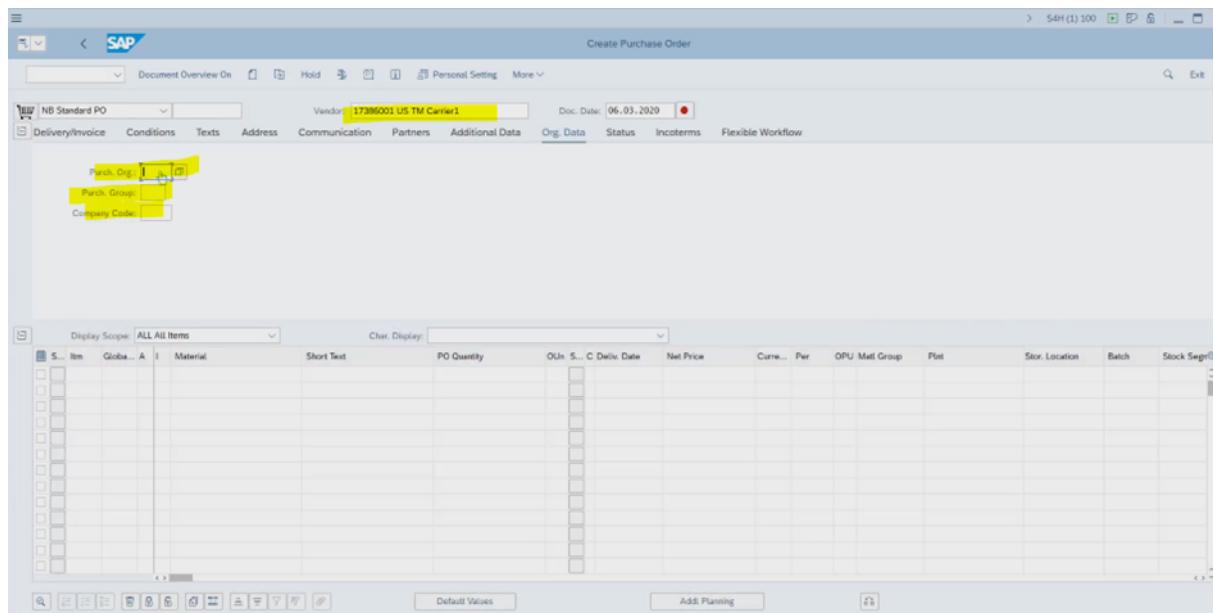
1. Automate basic SAP tasks, such as entering data into input fields and SAP Tables, using Checkboxes, Dropdown Menus and Popups
2. Use wildcards when automating dynamic elements from SAP
3. Work with pre-built SAP activities and automate SAP Menus, SAP Tabs and SAP Toolbars
4. Analyse SAP elements by using UI Explorer
5. Verify the purchase order status from the SAP Statusbar

Demo: Creating a purchase order

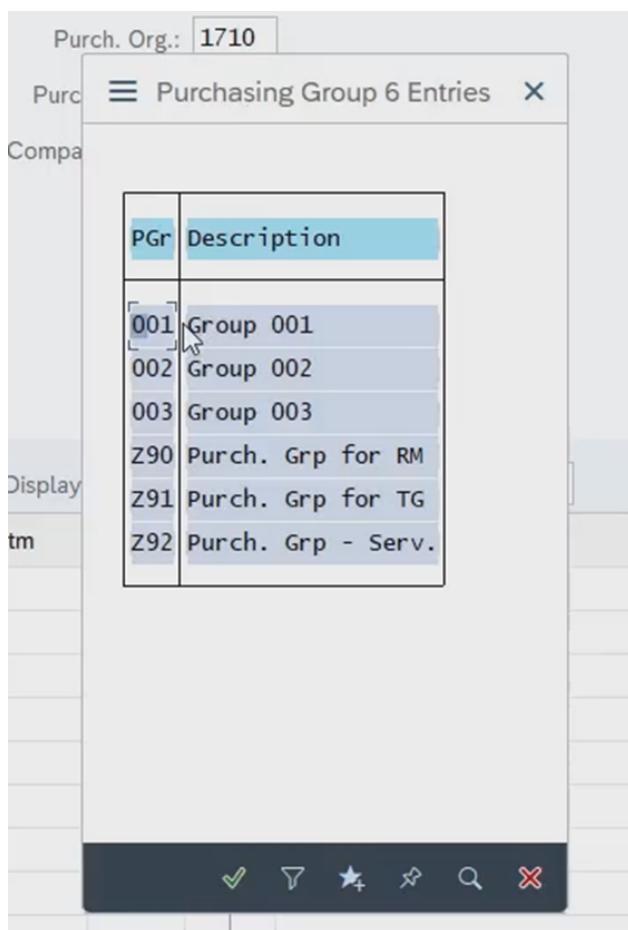
We will enter the connection, input the vendor name, purchase organisation purchase group and company code as shown below.

While inputting different vendor names, the selector will change and we have to use wildcards

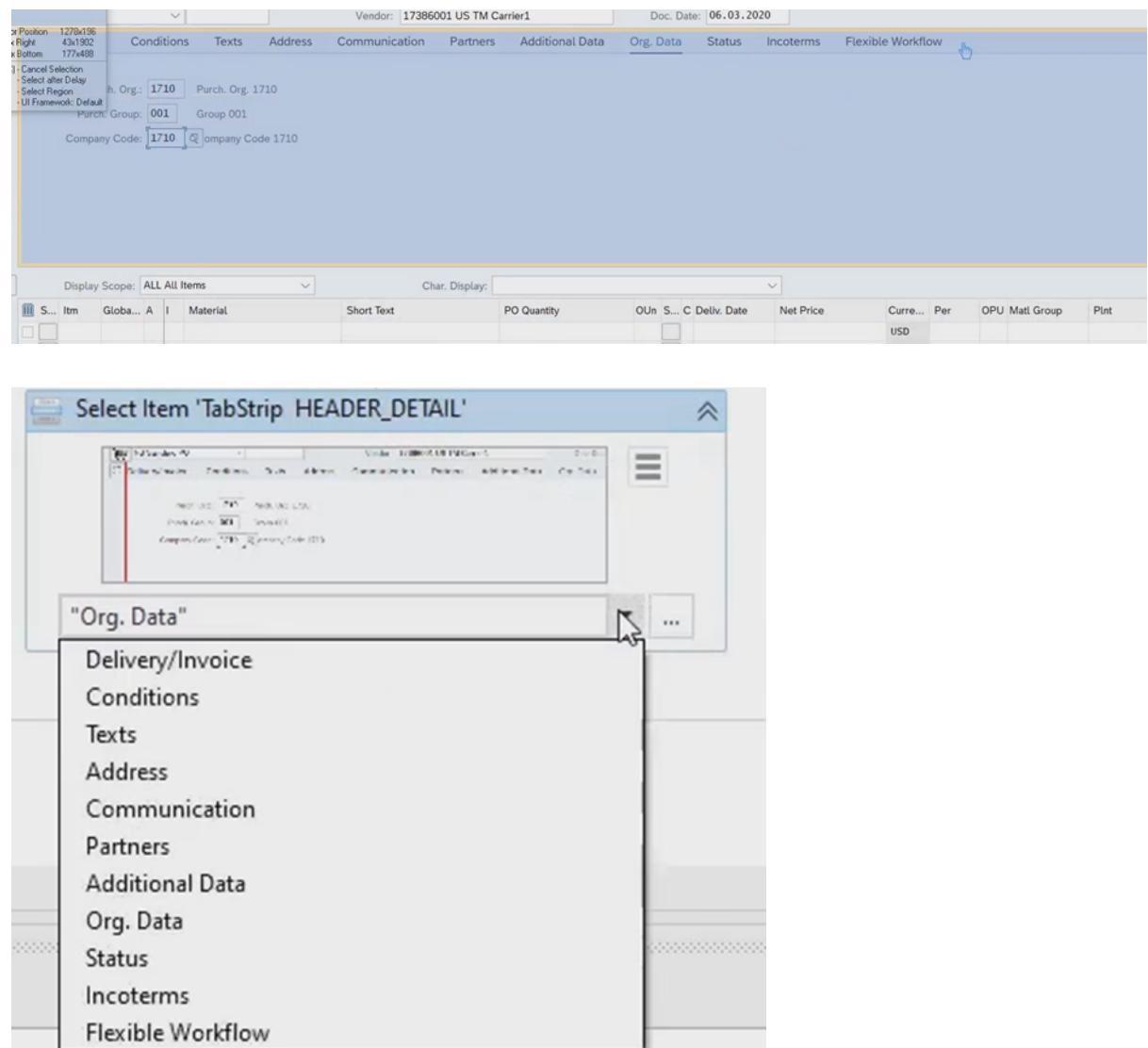




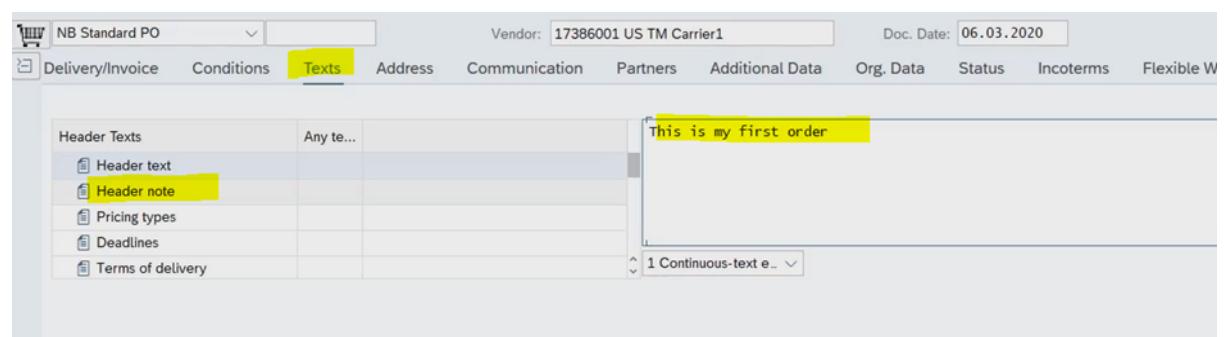
Purchase group will have multiple options and you will have to select the search help button to select the values out of it. in this case we pass the send hockey and then use click activity.



In the next step we will need to select this tab in Sep and due to the deep integration of studio with scp we can select any tab option from the drop down menu using select item activity and then choosing the value from the drop down as below



Add wildcards checking the selectors. Lets click on Texts tab, Header note field and add comments.



Using Type InTo activity, enter the data inside the table like below

Header Texts	Any to...	This is my first order
Header note	✓	
Pricing types		
Deadlines		
Terms of delivery		
Shipping instructions		

Display Scope: ALL All Items		Char. Display:															
S...	Item	Globa... A	I	Material	Short Text	PO Quantity	OUn	S...	C Deliv. Date	Net Price	Curre...	Per	OPU	Matl Group	PInt	Stor. Location	Batch
	10	TG11		Computer Mouse Black		2 PC		D	16.03.2020	13,00	USD	1	PC	Trading Materi...	Plant 1 US		

Now we have to navigate to another tab and so lets use “Select Item” activity and indicate the whole tab inorder to access the complete region texts like below

Display Scope: ALL All Items		Char. Display:																
S...	Item	Globa... A	I	Material	Short Text	PO Quantity	OUn	S...	C Deliv. Date	Net Price	Curre...	Per	OPU	Matl Group	PInt	Stor. Location	Batch	Stoc
	10	TG11		Computer Mouse Black		2 PC		D	16.03.2020	13,00	USD	1	PC	Trading Materi...	Plant 1 US			

Item: 1[10]TG11 , Computer Mouse Black	<input type="button" value="Default Values"/>	<input type="button" value="Add Planning"/>
<input checked="" type="checkbox"/> Inv. Receipt Tax Code: <input type="text"/> <input type="checkbox"/> Final Invoice Jurid. Code: <input type="text" value="7700000000"/>		
Material Data Quantities/Weights Delivery Schedule Delivery Invoice Conditions Texts Delivery Address Confirmations Condition Control Retail Spec2000 / SC Incoterms		

Select the required data from dropdown

↓

Select Item 'TabStrip ITEM_DETAIL'

Material Data

Quantities/Weights

Delivery Schedule

Delivery

Invoice

Conditions

Texts

Delivery Address

Confirmations

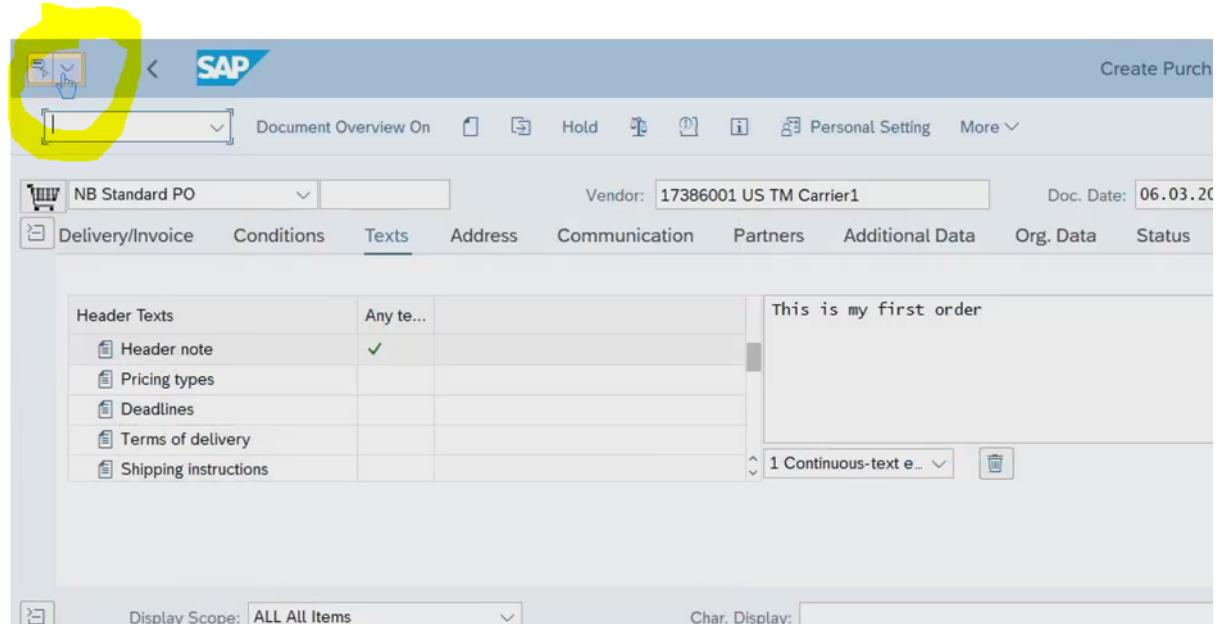
Condition Control

Retail

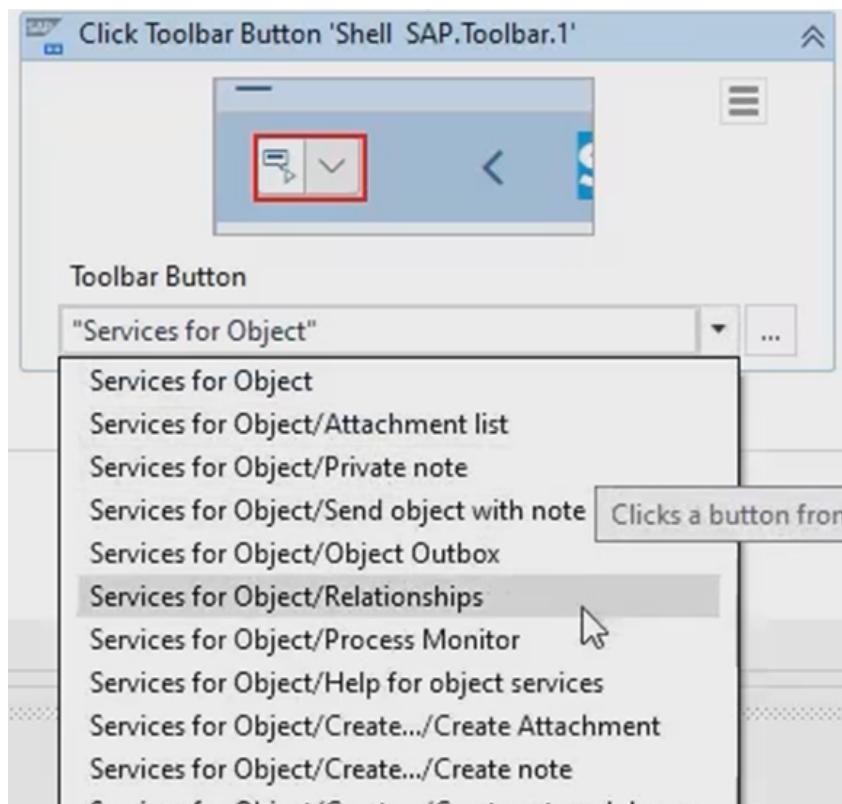
Spec2000 / SC

Incoterms

When we want to click on object button, we use “Click ToolBar Button” activity

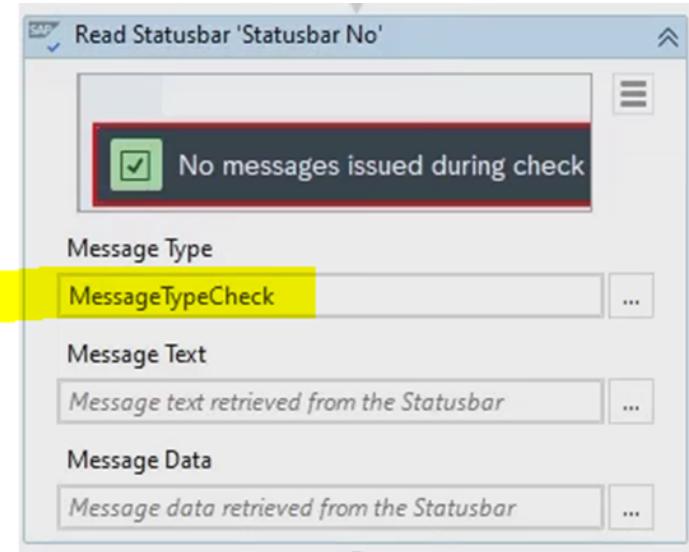


Select the create note option from dropdown to create notes

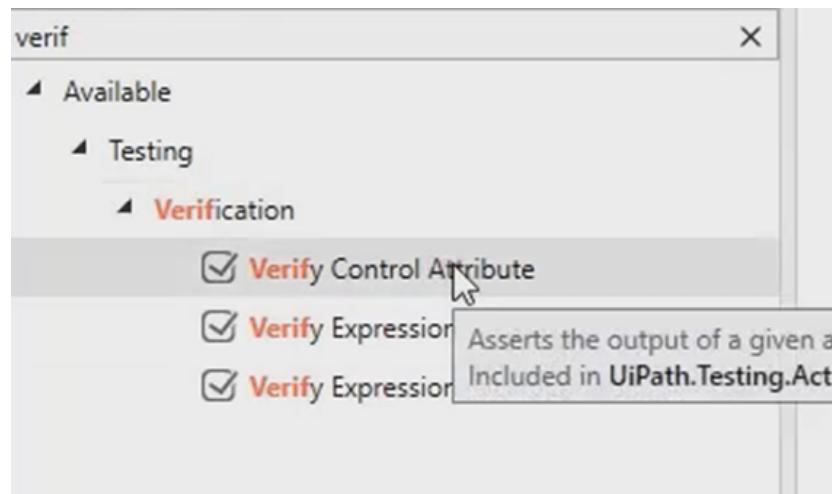


Next is to verify if everything has been entered. Let's check it from the application end. in the status bar at the bottom of the application we will be able to find the status. to read this we are using read status bar activity of SAP studio activity like below

Lets create a variable to store the output as below

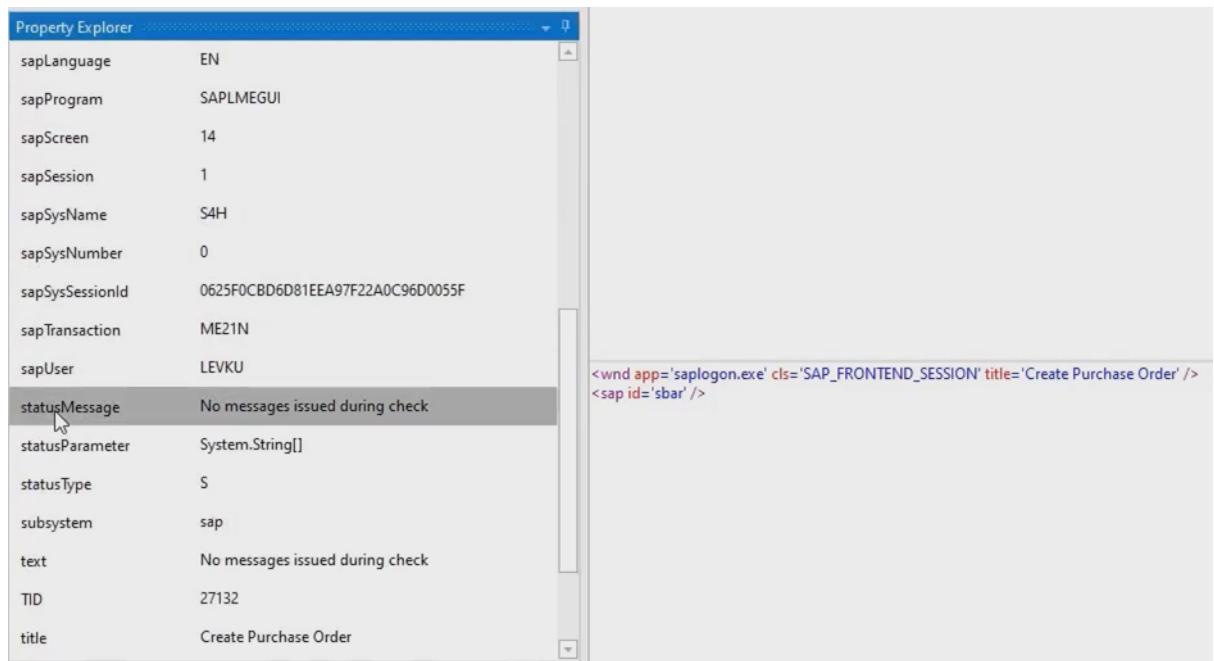
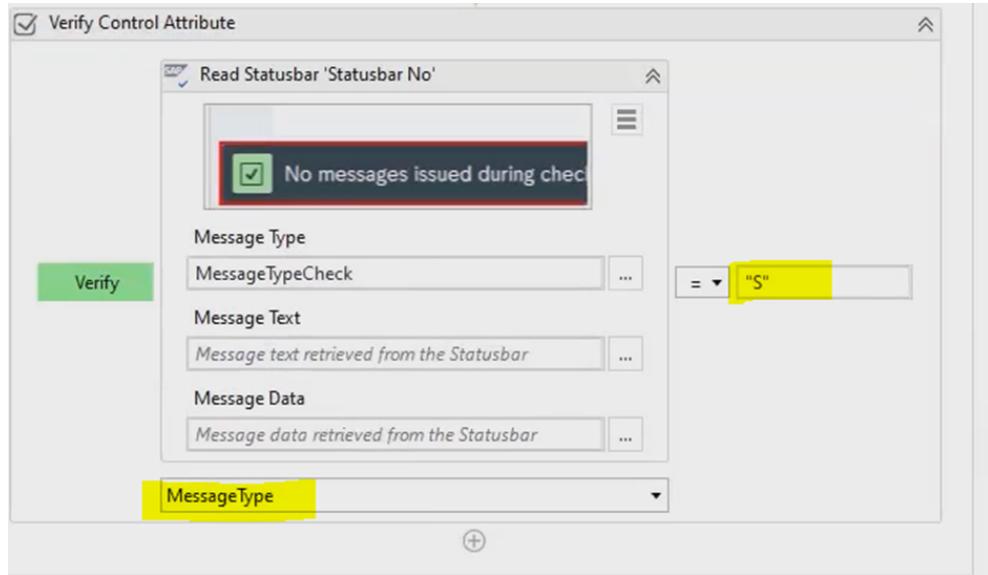


In order to verify this, we use the activity “Verify Control Attribute”

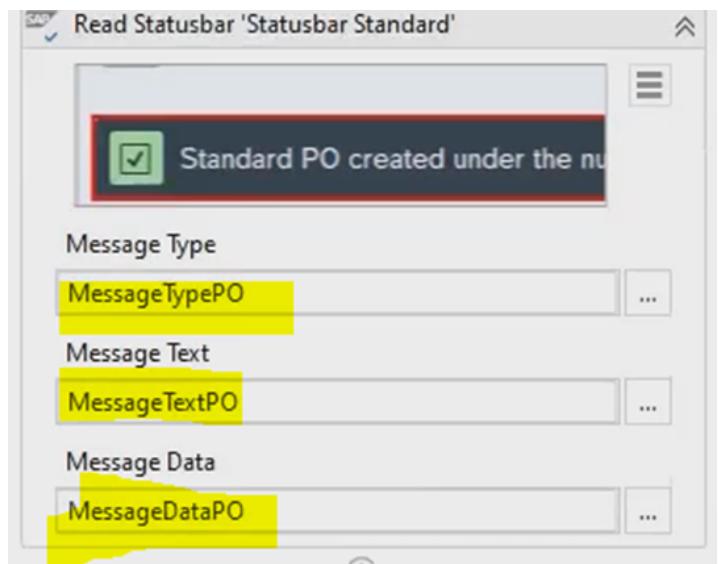


Place the “Read SttausBar” activity inside “Verify Control Attribute” like below

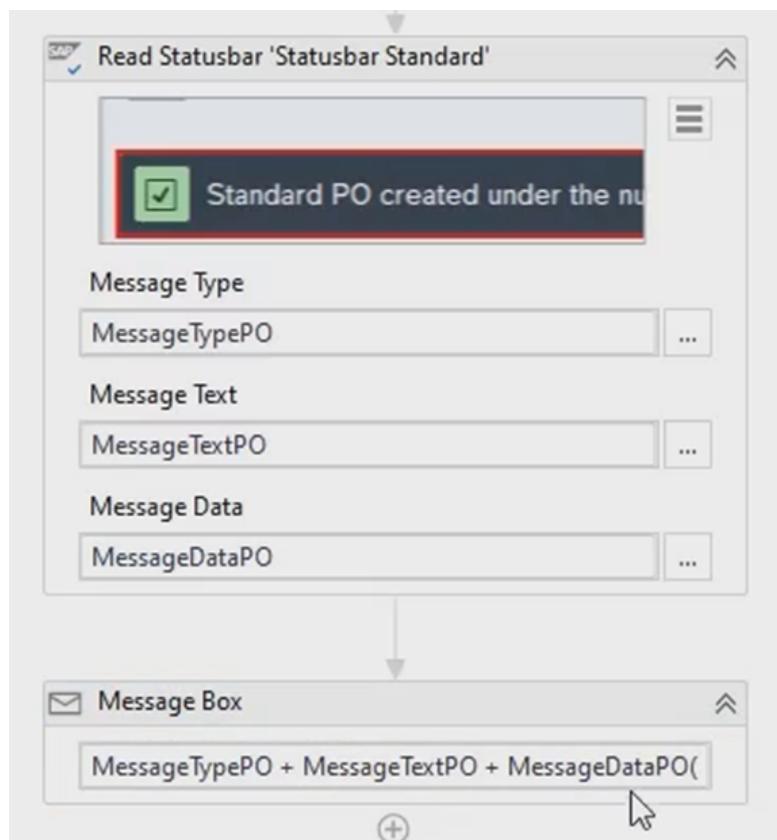
Check the attributes value from UIExplorer and mention the same (in our case it is S)



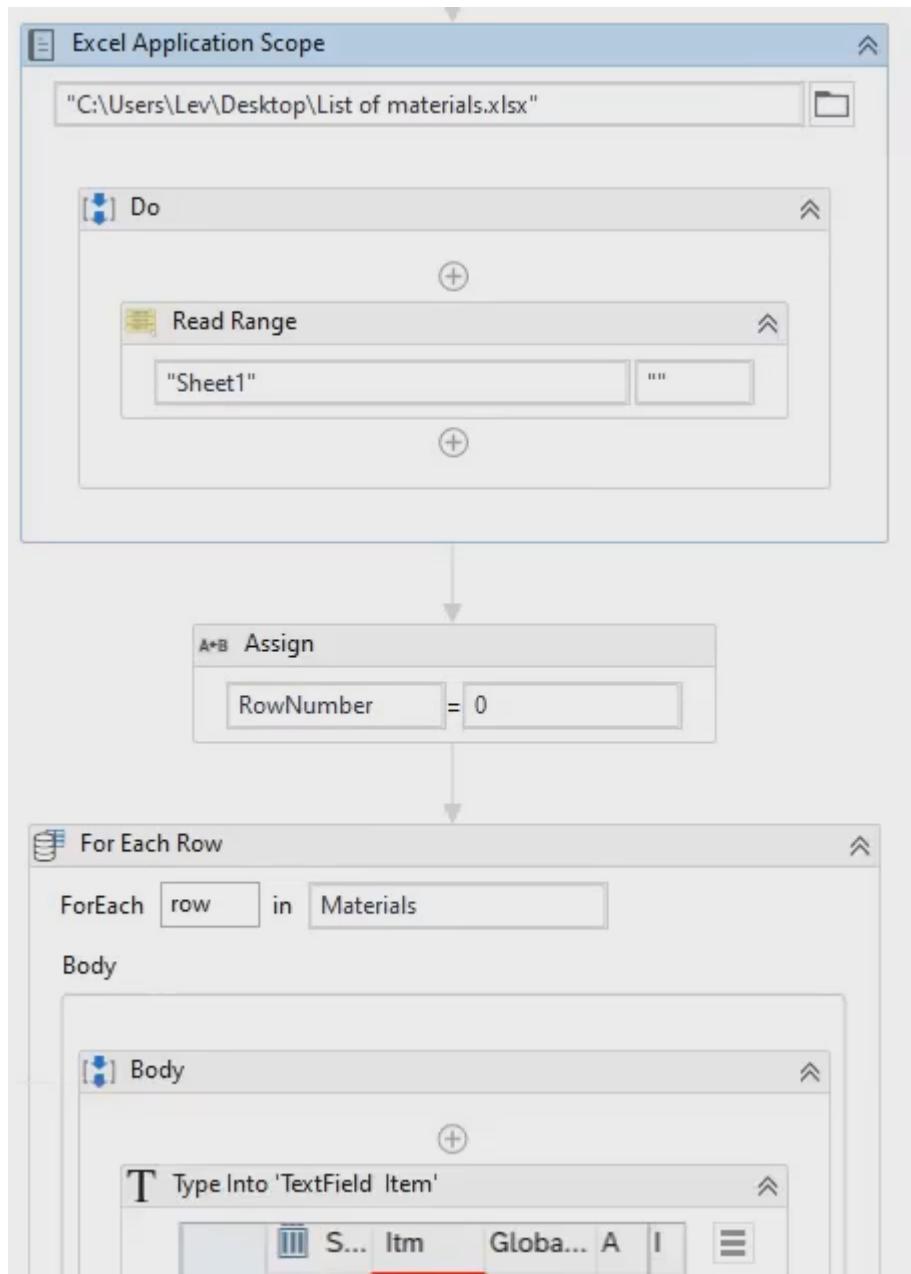
Once verification is done, click on save and get the status bar information using “Read StatusBar status”. I have created 3 variables



Lets display the output in message box



Importing Data From Excel to SAP

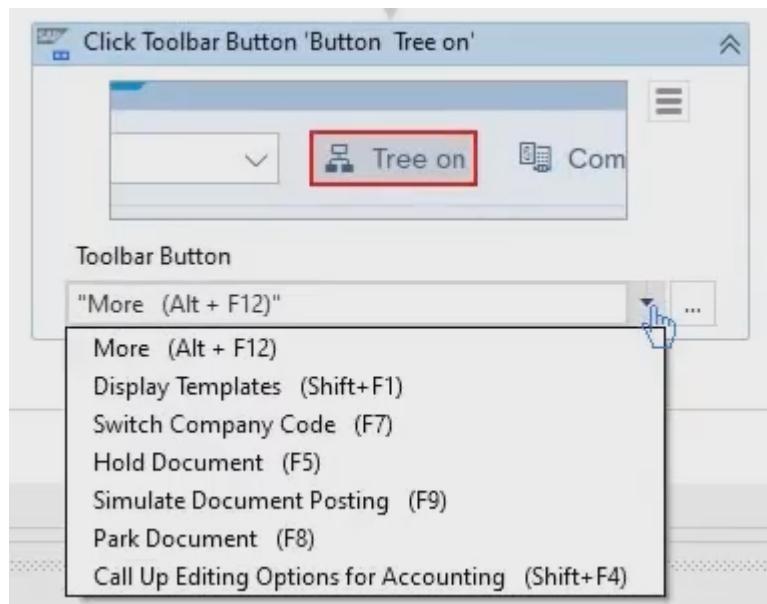


Best Practice is to clear the field (empty field) before entering any data in to the sap fields.

Creating an Invoice

Here we perform entering data into the SAP application fields and then selecting connections from TREE and entering data into the connection of a TREE.

After using few Type Into activities, we will need to select some variants from a tree. Before selecting tree variants, we should click on tree for which we use "Click ToolBar button" activity as the TREE option is present in toolbar.



Here we will be able to use any sub buttons of the toolbar button we selected.

St...	G/L acct	Short Text	D/C	Amount in doc.curr.	Loc.curr.amount	T...	Tax jurisdctn code	W Assignment
<input type="checkbox"/>		S Deb..	✓		0,00			<input type="checkbox"/>
<input type="checkbox"/>		S Deb..	✓		0,00			<input type="checkbox"/>
<input type="checkbox"/>		S Deb..	✓		0,00			<input type="checkbox"/>
<input type="checkbox"/>		S Deb..	✓		0,00			<input type="checkbox"/>

After clicking on TREE, we have to expand tree & select one of the options as below

Tree		Created on	Created by
✓	Screen variants for items		
	RE ABRECHNUNG_0100	RE settle	12.05.99
	RE OBJEKTE_0100	RE objects	12.05.99
	RE VERTRAEGE_0100	RE contract	12.05.99
	STANDARD 1_0100	Standard 1	10.05.99
	STANDARD 2_0100	Standard 2	10.05.99
	STANDARD 3_0100	Standard 3	14.05.99
	STANDARD 4_0100	Standard 4	05.08.99
	Account assignment templates for items		
	Held documents		

SAP trees can be huge & dynamic & will be loading during scrolling but with UiPath studio, we can handle all such uncertainties.

For this we use “Expand tree” activity by selecting the whole tree, all the elements will be accessible.

The screenshot shows the 'Expand Tree' activity configuration in UiPath Studio. The tree structure is expanded, and the 'Tree Item' dropdown shows the selected item: 'Screen variants for items/STANDARD 4_0100'. This item is highlighted with a gray background and has a cursor pointing at it.

We will double click once selecting the tree variant. For double clicking we can indicate whole tree & then select the options or indicate the variant alone

The screenshot shows a SAP Fiori application window titled "Expand Tree 'Tree SAP.TableTreeCont...'". Below the title bar is a toolbar with icons for search, refresh, and other functions. The main area displays a table with columns: Item, Created on, and Created by. There are five rows in the table. One row, "STANDARD 4_0100", is highlighted with a red border. Below the table is a section labeled "Tree Item" containing a dropdown menu with the text "Screen variants for items/STANDARD 4_0100". A downward arrow points from this section to another window below.

Double Click

"STANDARD 3_0100"
"STANDARD 4_0100" (highlighted with a red border)
> Account assignment templates fo

Use the activity “Select Menu Item” and turn off tree.

The screenshot shows a SAP Fiori application window titled "Select Menu Item 'saplogon.exe Enter'". Below the title bar is a toolbar with icons for search, refresh, and other functions. The main area displays a menu structure. A specific menu item, "Document/Change", is highlighted with a red border. Below the menu structure is a section labeled "Menu Item" containing a dropdown menu with the text "Document/Change". A button labeled "Select a Menu" is located to the right of the dropdown.

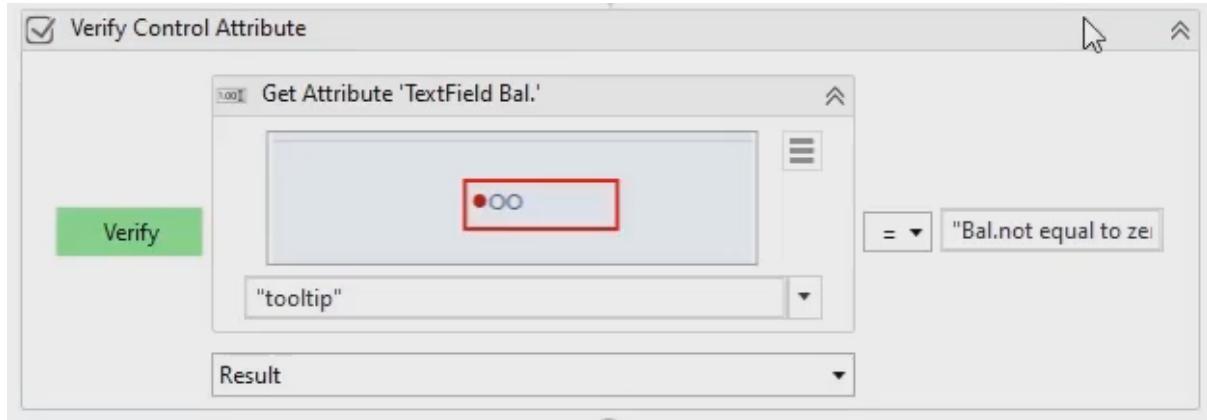
Menu Item
"Document/Change"

Use the activity “Select Item” to select data from SAP dropdowns.

The screenshot shows a SAP Fiori application window titled "Select Item 'ComboBox D/C'". Below the title bar is a toolbar with icons for search, refresh, and other functions. The main area displays a dropdown menu with two options: "S Debit" and "S Credit". The "S Debit" option is highlighted with a red border. Below the dropdown is a section labeled "Debit" containing a dropdown menu with the text "Debit". A button labeled "Select a Menu" is located to the right of the dropdown.

Short Text
D/C
Amount in do
S Debit...
S Credit...
Debit

Atlast we will verify certain parameters. In order to verify this, we use the activity “Verify Control Attribute”

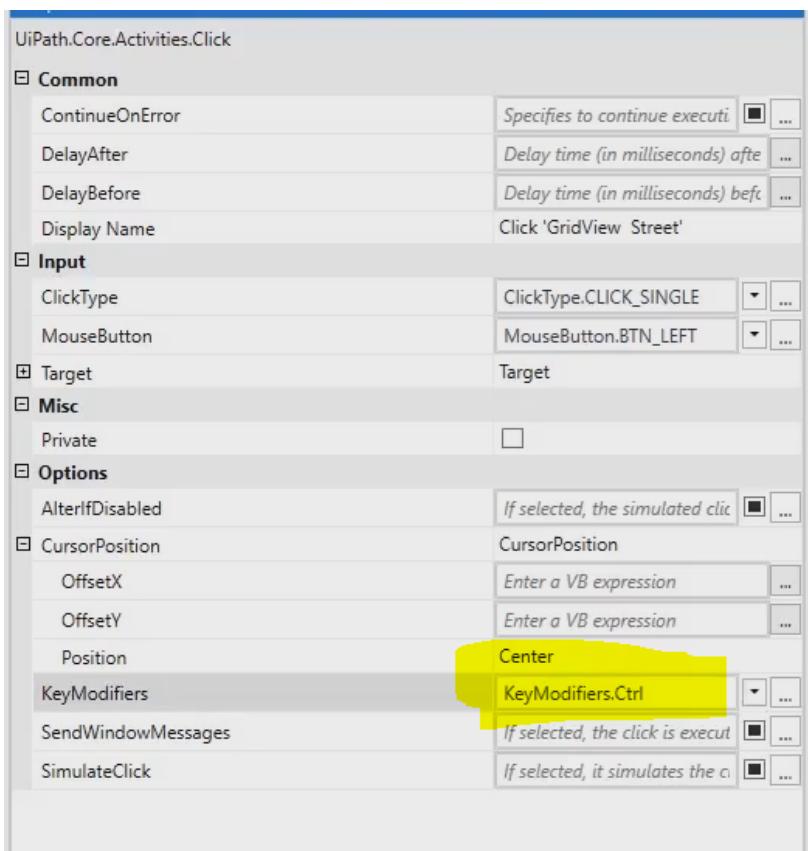


Common Table Operations

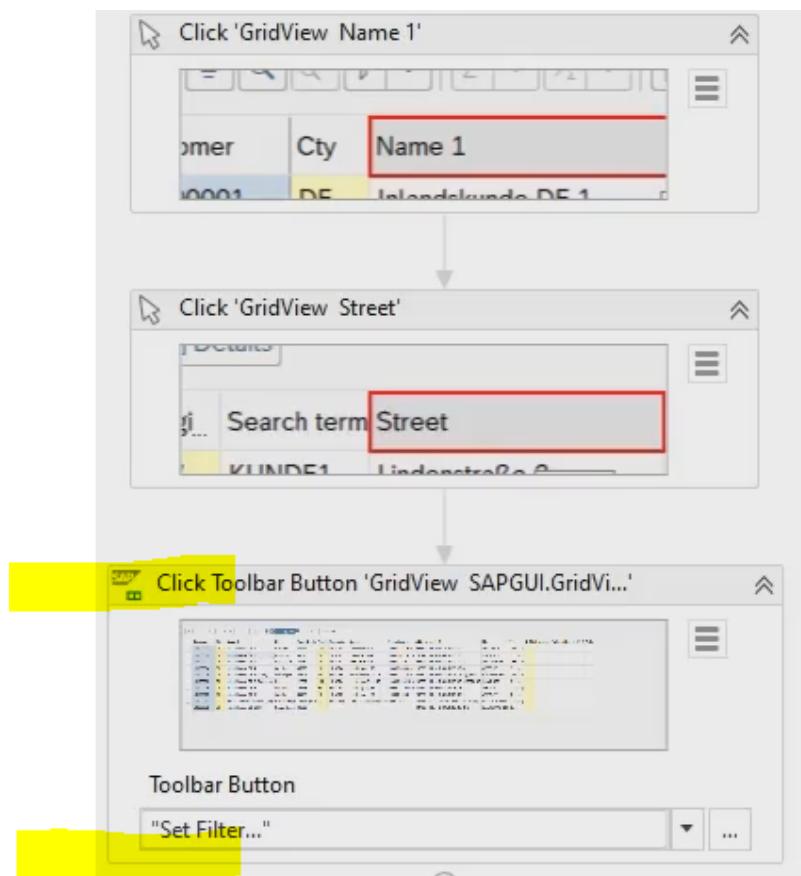
When we select any column using click activity, column name and other attributes will be a part of the selector. When we select the whole row, row number & row elements will be a part of the selector

While selecting the columns in parallel, I am making use of “Key modifiers” in the property section and selecting “Control” in the 2nd Click activity so that the 2nd column will be selected in parallel with the 1st one.

	Customer	Cty	Name 1	City	Postal Code	Regi...	Search term	Street	Telephone 1	Address
	10100001	DE	Inlandeskunde DE 1	Schöntal	74214	BW	KUNDE1	Lindenstraße 2	09990 4513-0	23742
	10100002	DE	Inlandeskunde DE 2	Chemnitz	09119	SN	KUNDE2	Platnerstraße 45	09990 41252-0	23744
	10100003	DE	Inlandeskunde DE 3	Budenheim	55257	RP	KUNDE3	Römerstraße 1230	09990 365-0	23768
	10100004	DE	Inlandeskunde DE 4	Wetzlar	35578	HE	KUNDE4	Lahnbergweg 23	09990 3568-0	23770
	10100005	DE	Inlandeskunde DE 5 (CMS)	Steinhagen	33803	NW	KUNDE5	Heinrichstraße 1020	09990 4513-0	23780
	10100006	DE	Inlandeskunde DE 6 (Retouren)	Hamburg	22767	HH	KUNDE6	Holstenstraße 1142	09990 25420-0	23772
	10100008	DE	Inlandeskunde DE 8	Wetzlar	35578	HE	KUNDE8	Lahnbergweg 83	09990 28920-0	23774
	10100009	DE	Inlandeskunde DE 9	Wetzlar	35578	HE	KUNDE9	Lahnbergweg 93	09990 4558-0	23776
	10100050	US	Ausländischer Kunde 50 (US)	Baton Rouge	70817-3609	LA	KUNDE50	15400 Confederate Ave	999 236 5237	23786
	10100273	DE	Inlandeskunde DE CPD	Saarbrücken	66119					23782

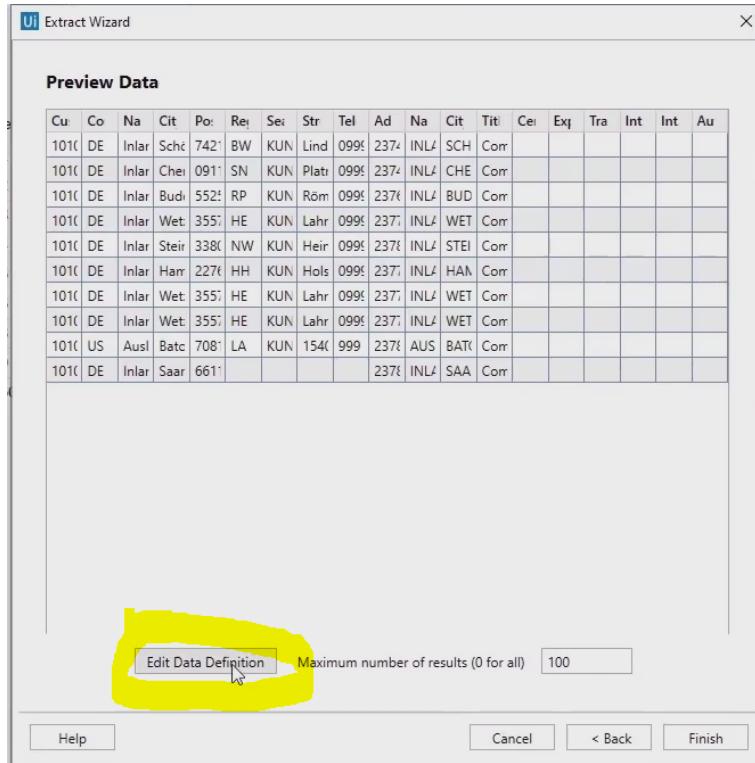


As shown in the above snaps, for clicking on Filter button in snap step 3, we will use “Click Toolbar button” as below

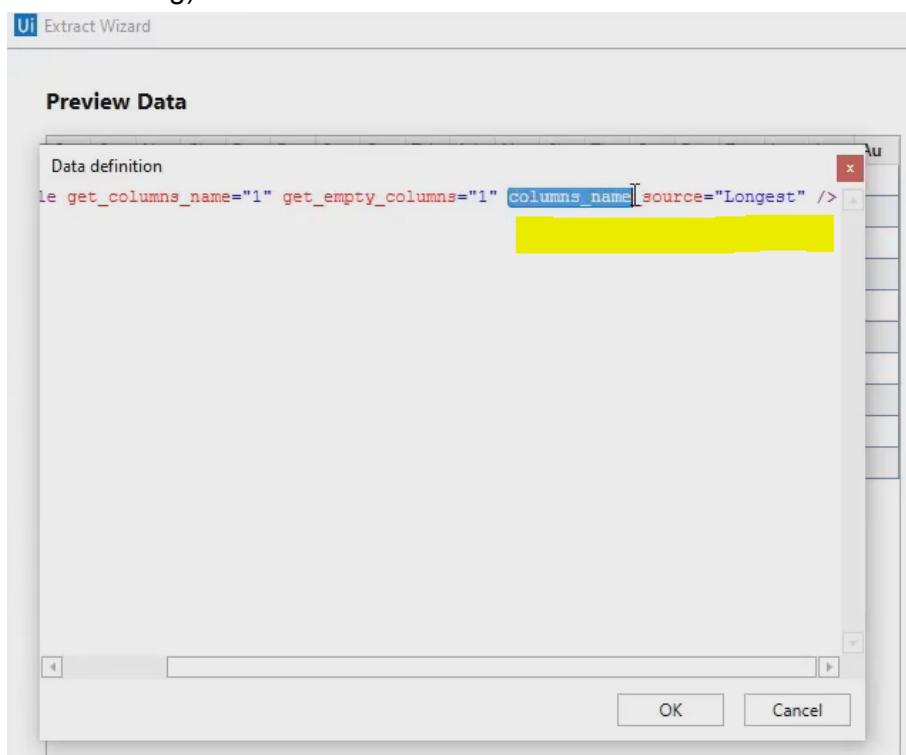


Data Scraping in SAP Tables

For extracting SAP table data into excel, we can use ‘Data scraping’ functionality. Once we click on the table, whole data will be extracted. We can click on “Edit Data Definition” like below before saving the table data.



Here we can indicate the exact column source of our column from this specific expression like below. Ours is set to display the longest version of the column name (which is also the default setting) even when table column is hidden a bit



Now lets click on Finish

UI Extract Wizard

Preview Data

Cu	Co	Na	Cit	Po	Re	Se	Str	Tel	Ad	Na	Cit	Tit	Cer	Exp	Tra	Int	Int	Au
1010	DE	Inlar	Schö	742	BW	KUN	Lind	099	237	INLA	SCH	Cor						
1010	DE	Inlar	Chei	0911	SN	KUN	Platr	099	237	INLA	CHE	Cor						
1010	DE	Inlar	Bud	552	RP	KUN	Röm	099	237	INLA	BUD	Cor						
1010	DE	Inlar	Wet	355	HE	KUN	Lahr	099	237	INLA	WET	Cor						
1010	DE	Inlar	Steir	3380	NW	KUN	Heir	099	237	INLA	STEI	Cor						
1010	DE	Inlar	Harr	227	HH	KUN	Hols	099	237	INLA	HAN	Cor						
1010	DE	Inlar	Wet	355	HE	KUN	Lahr	099	237	INLA	WET	Cor						
1010	DE	Inlar	Wet	355	HE	KUN	Lahr	099	237	INLA	WET	Cor						
1010	US	Ausl	Batc	708	LA	KUN	154	999	237	AUS	BATC	Cor						
1010	DE	Inlar	Saar	6611					237	INLA	SAA	Cor						

Edit Data Definition Maximum number of results (0 for all) 100

Help Cancel < Back Finish

Our data is now stored in data table

Data Extraction

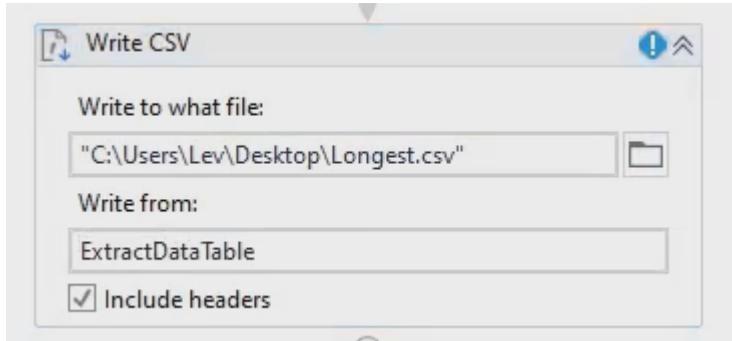
Data Scraping

Attach Window 'KNA1Displ Sap_front'

Do

Extract Structured Data 'GridView SAPGUI.GridVi...'

Let's write the extracted data in csv using 'write csv' activity like below



Below are the column name options

Data Scraping

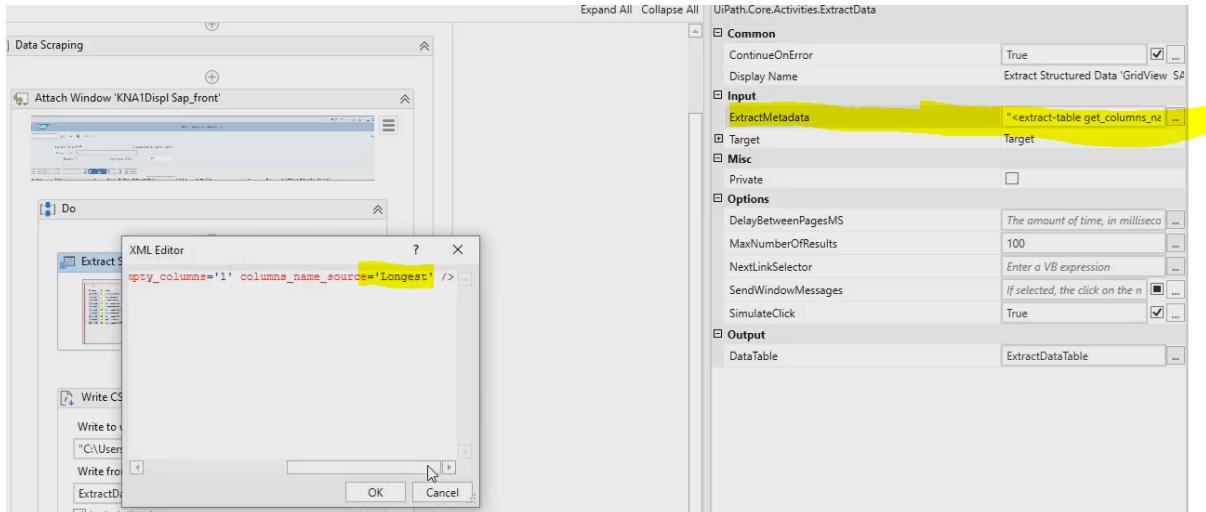
Data scraping enables you to extract structured data from SAP Tables, as exemplified below.

The **Extract Structure Data** activity provides additional possibilities while extracting the data from the SAP GridView table. You can use the following values for the `columns_name_source` parameter to define the way you want to get the table information:

- **Longest** - The full name of column is extracted and used.
- **Shortest** - The short name of column is extracted and used.
- **Displayed** - The visible/displayed name in SAP WinGUI is used.
- **Technical** - The technical header name of the column is extracted and used.

If you place 'Longest' with 'Shortest' in the 'Edit Data Definition' you will see the change.

Do this in the properties of the data extraction activity like below



Now the shortest column names will be displayed like below

Customer	Cty	Name	City	Post.Code	Rg	SearchTer	Street	Telephone	Address	Name1	City1	Title	OrBk	Expr.stat.	Train stat	ILN 1	ILN 2	AuGr
10100001	DE	Inlandsku	Schöntal	74214	BW	KUNDE1	Lindenstr	09990 451	23742	INLANDSK	SCHÖNTA	Company						
10100002	DE	Inlandsku	Chemnitz	9119	SN	KUNDE2	Platnerstr	09990 412	23744	INLANDSK	CHEMNITZ	Company						
10100003	DE	Inlandsku	Budenhei	55257	RP	KUNDE3	Römerstr	09990 365	23768	INLANDSK	BUDENHEI	Company						
10100004	DE	Inlandsku	Wetzlar	35578	HE	KUNDE4	Lahnberg	09990 356	23770	INLANDSK	WETZLAR	Company						
10100005	DE	Inlandsku	Steinhage	33803	NW	KUNDE5	Heinrichstr	09990 451	23780	INLANDSK	STEINHAG	Company						
10100006	DE	Inlandsku	Hamburg	22767	HH	KUNDE6	Holstenstr	09990 254	23772	INLANDSK	HAMBURG	Company						
10100008	DE	Inlandsku	Wetzlar	35578	HE	KUNDE8	Lahnberg	09990 289	23774	INLANDSK	WETZLAR	Company						
10100009	DE	Inlandsku	Wetzlar	35578	HE	KUNDE9	Lahnberg	09990 455	23776	INLANDSK	WETZLAR	Company						
10100050	US	Ausländis	Baton Rou	70817-360	LA	KUNDE50	15400 Con	999 236 52	23786	AUSLÄNDIS	BATON RC	Company						
10100273	DE	Inlandsku	Saarbrück	66119					23782	INLANDSK	SAARBRÜ	Company						

If we change the Longest to Displayed, then the displayed column data will be extracted in the CSV like below

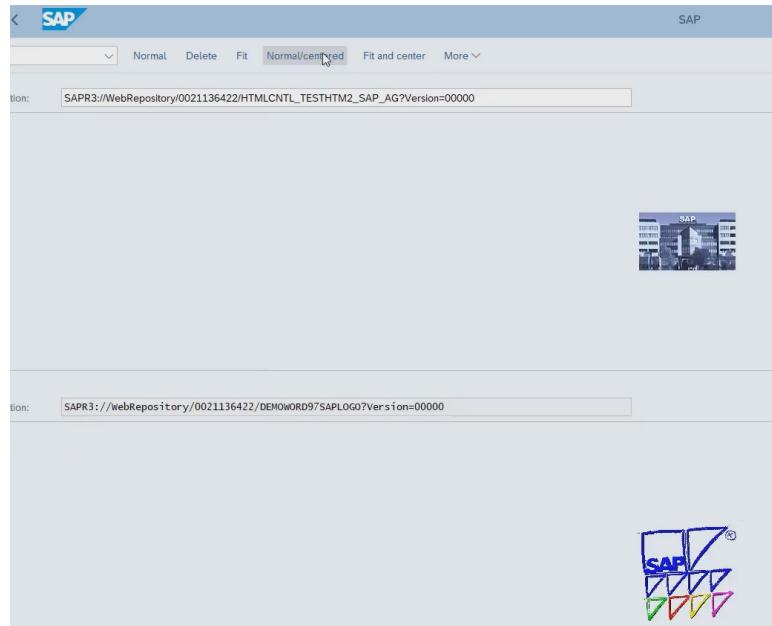
Customer	Cty	Name	City	Postal	Code	Rg	Search	ter Street	Telephone	Address	Name	1	City1	Title	OrBk	Expr.stat.	Train stat	ILN 1	ILN 2	AuGr
10100001 DE	Inlandsru Schöntal			74214	BW		KUNDE1	Lindenstr	09990 451		23742	INLANDSK SCHONTA Company								
10100002 DE	Inlandsru Chemnitz			9119	SN		KUNDE2	Platnerstr	09990 412		23744	INLANDSK CHEMNTZ Company								
10100003 DE	Inlandsru Budenhei			55257	RP		KUNDE3	Römerstr	09990 365		23768	INLANDSK BUDENHEI Company								
10100004 DE	Inlandsru Wetzlar			35578	HE		KUNDE4	Lahnberg	09990 356		23770	INLANDSK WETZLAR Company								
10100005 DE	Inlandsru Steinlage			33803	NW		KUNDE5	Heinrichstr	09990 451		23780	INLANDSK STEINHAG Company								
10100006 DE	Inlandsru Hamburg			22767	HH		KUNDE6	Holstenstr	09990 254		23772	INLANDSK HAMBURG Company								
10100008 DE	Inlandsru Wetzlar			35578	HE		KUNDE8	Lahnberg	09990 289		23774	INLANDSK WETZLAR Company								
10100009 DE	Inlandsru Wetzlar			35578	HE		KUNDE9	Lahnberg	09990 455		23776	INLANDSK WETZLAR Company								
10100050 US	Ausländs Baton Rou	70817-360	LA				KUNDE50	15400 Con	999 236 52		23786	AUSLÄNDI BATON RC Company								
10100273 DE	Inlandsru Saarbrück			66119							23782	INLANDSK SAARBRÜ Company								

Picture on screen and SAP calendar

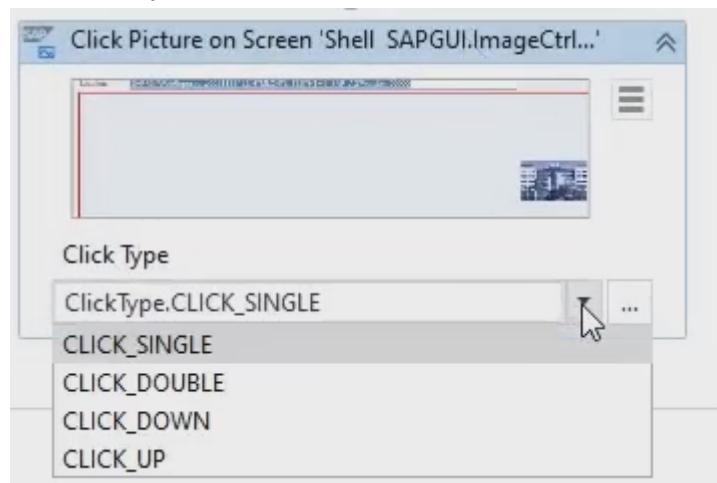
Open a demo transaction dwdm which contains multiple display elements

Here we click on 'Screen display'

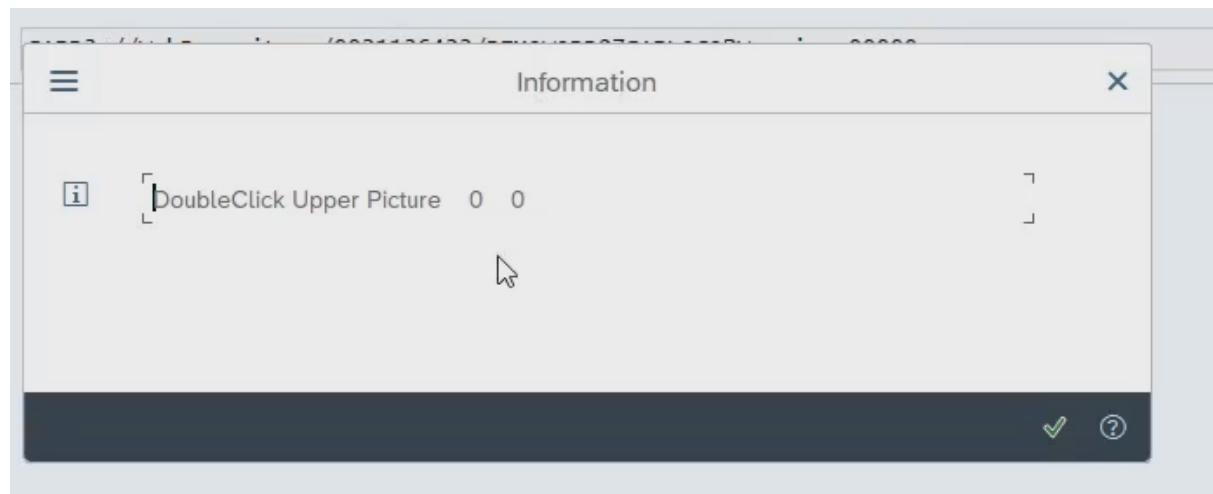
These pictures can be displayed like below based on the options



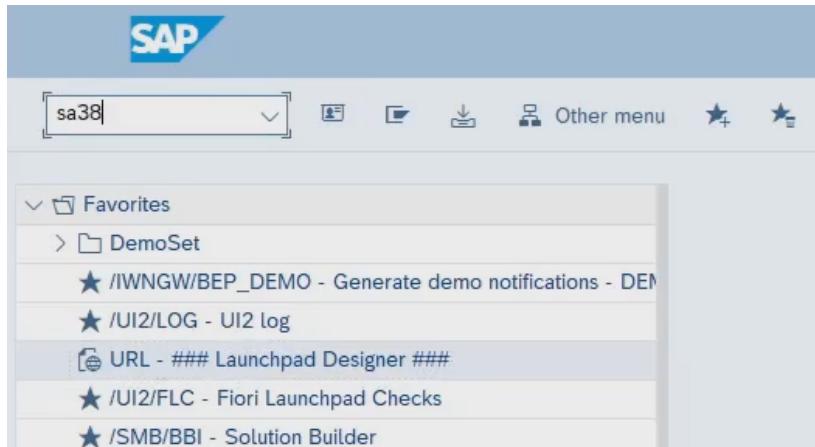
To click on this picture, we use the activity “**Click Picture on Screen**” & choose the type of interaction you want to do with that picture.



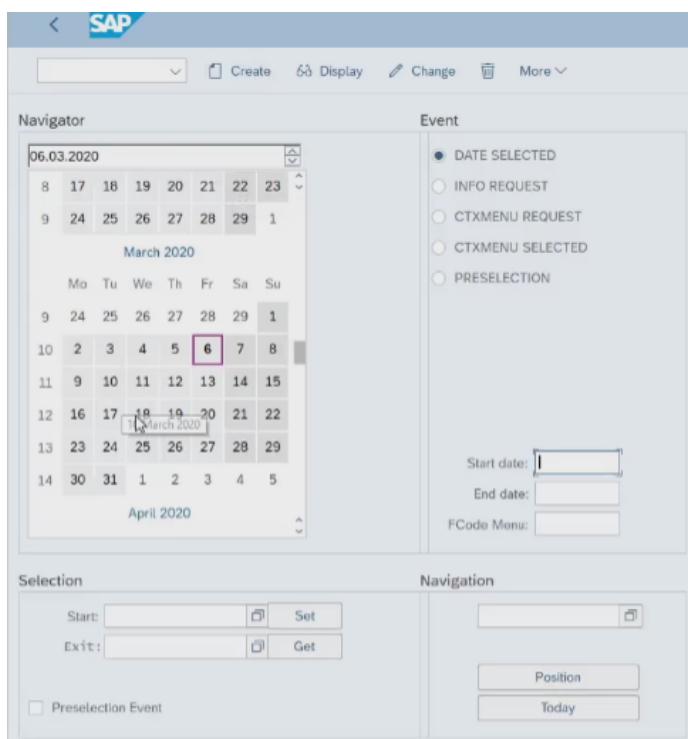
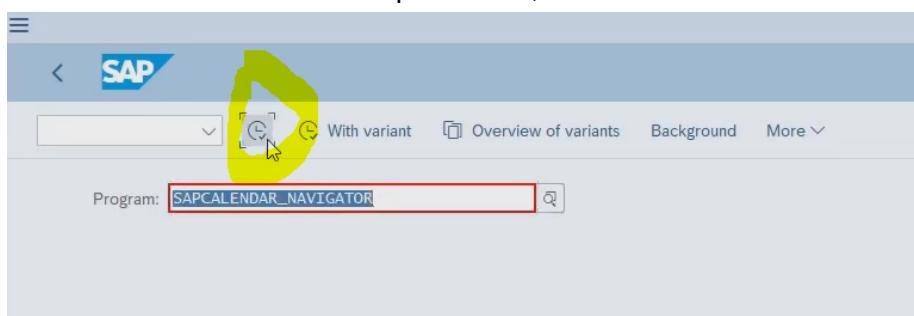
Let's choose double click and run the workflow, then we get a popup which shows what was done on the picture like below



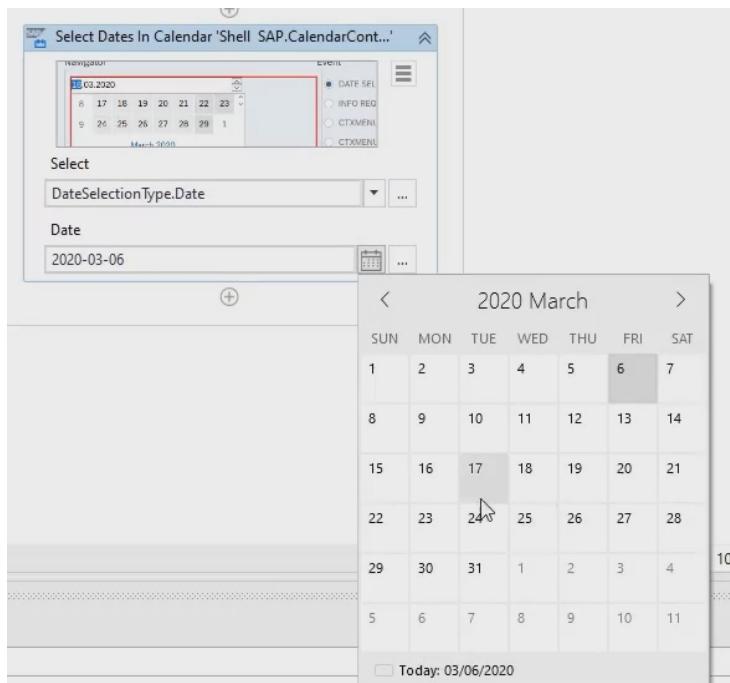
To select dates in calendar, select a demo transaction “sa38”



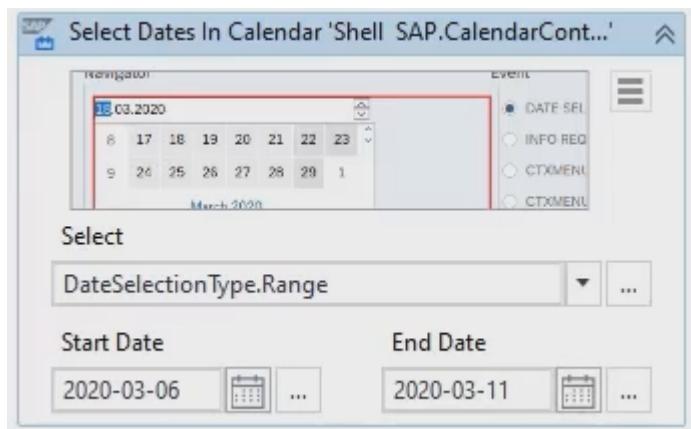
Here below we can see the sap calendar, click on that



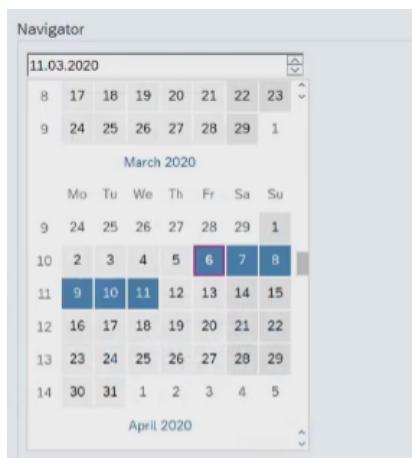
Due to the deep integration of sap with uipath studio, all of these actions can be easily automated using “**Select dates in calendar**” activity. We can indicate the calendar and indicate the date.



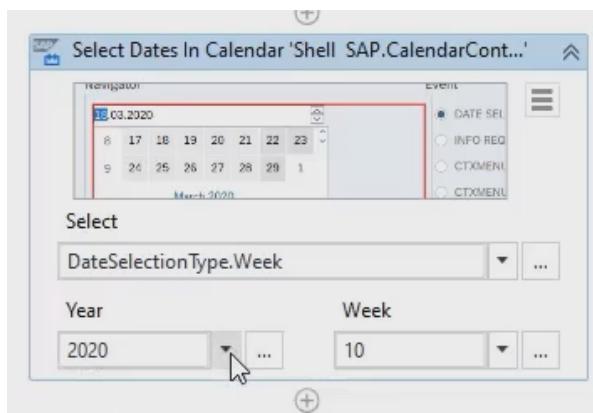
To select the dates in range, simply select the activity open type from date to range like below and mention start and end date



When the bot is run, below will be the selected sap application

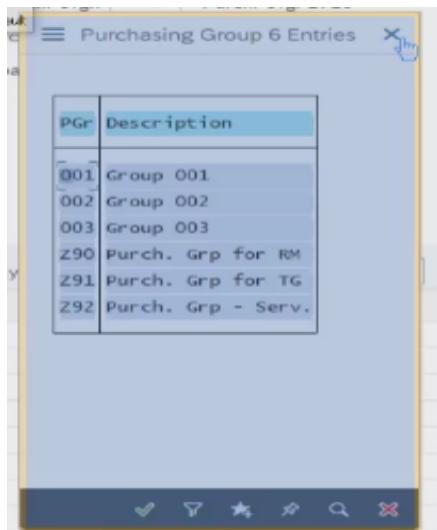


If you want to select the week, we can also do that like below

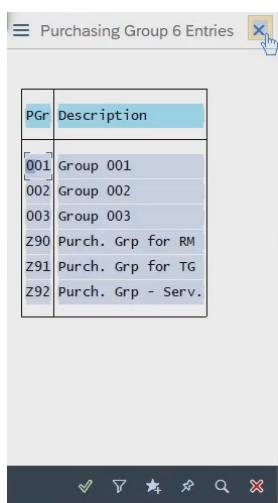


SAP Context Menu and Pop-Ups

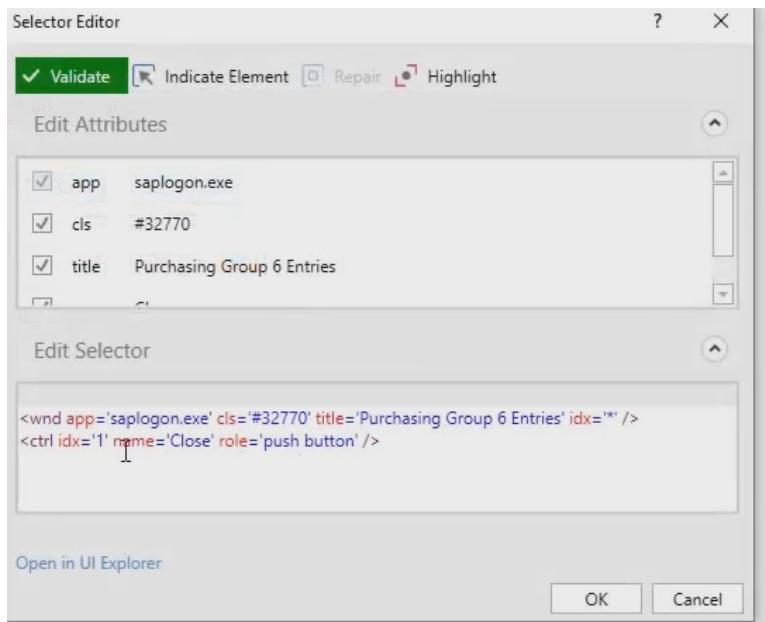
When we want to open any field and select data out of its menu, like below, we are unable to close this context popup



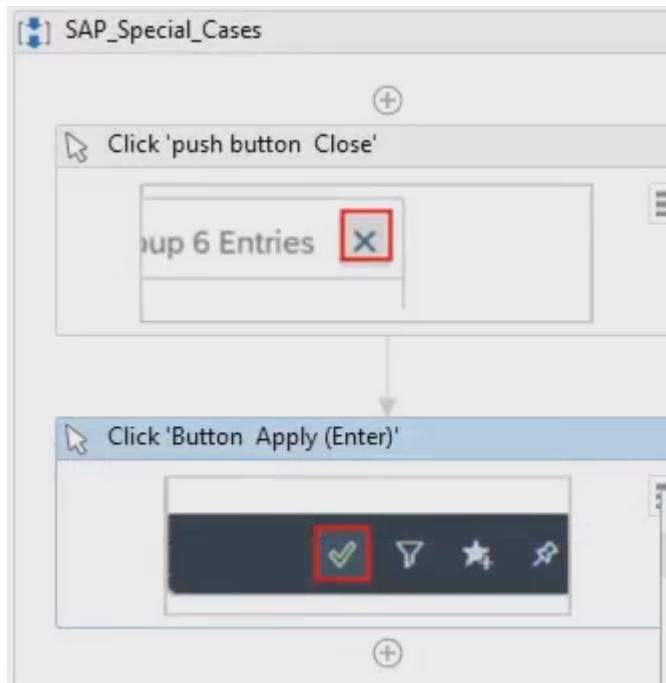
When we are unable to click on popup buttons, we can do it by switching the identification frame but andk. It is easily done by pressing F4 button when indicating the element.



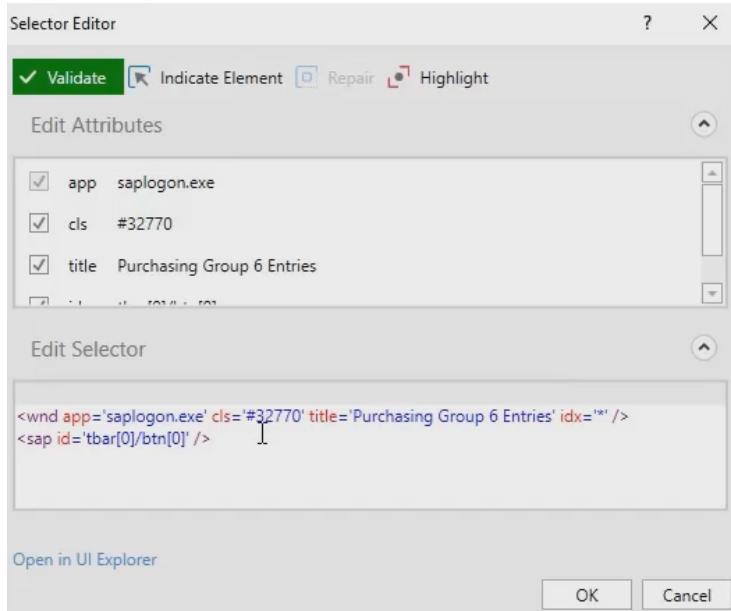
After the close button is indicated we can verify that it is not a native SAP selector but an AA automation framework selector by clicking “edit selector”.



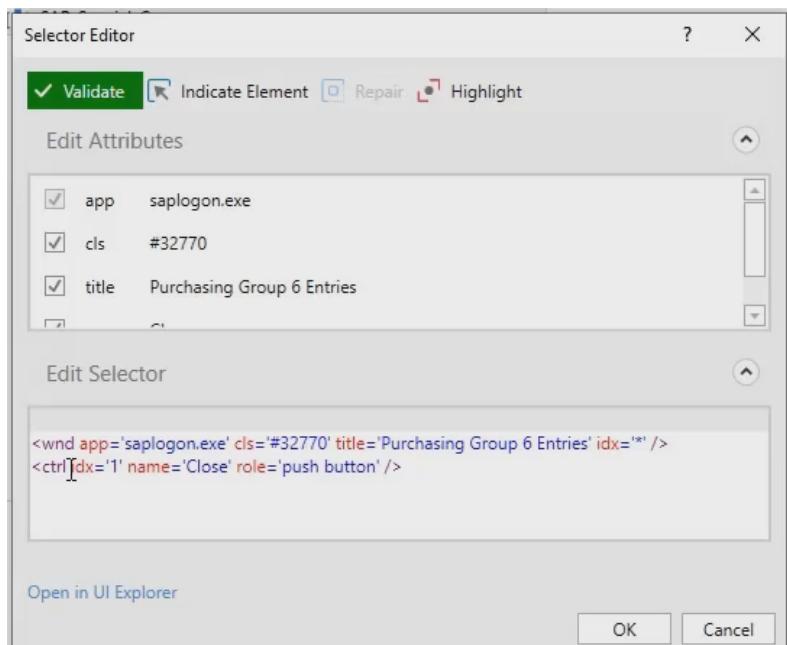
Lets see what a native SAP button looks like when using the click activity. We will change the framework to default which is SAP scripting interface by pressing F4 & click on the button so we can edit its selector.



To better differentiate between the 2 frameworks lets view the selectors in more detail.

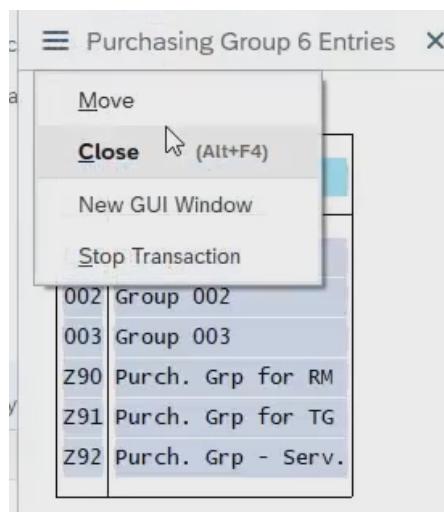


This is an SAP ID in a scripting interface shown above
Below is an AA identification framework. Shown above.

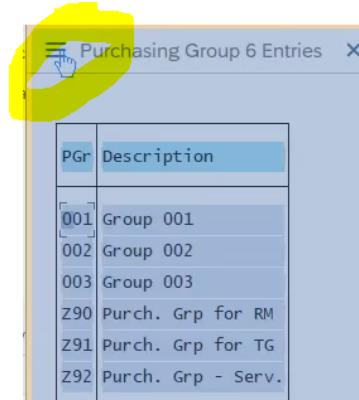


When you need to work with these type of popup and interact with their buttons, you will need to change the identification framework in order to match the selected type button.

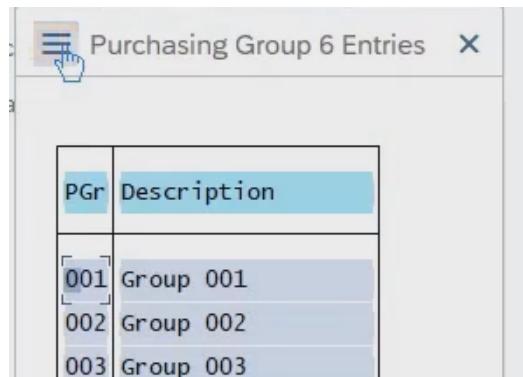
Lets see how we can interact with the menu button. To close the below popup from menu button, using studio activity click we will not be able to identify the menu buttons that are inside it.



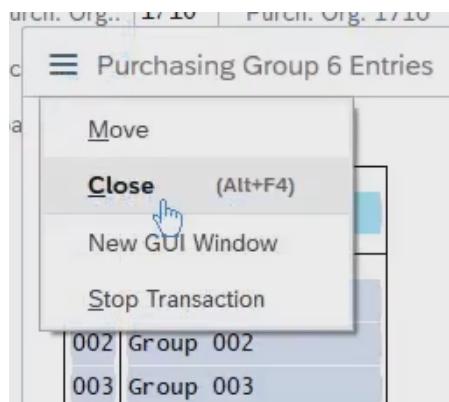
The element of interest does not get recognised as below.



So again we will need to click F4 and change the identification framework to AA and then menu button will be selectable & stable like below

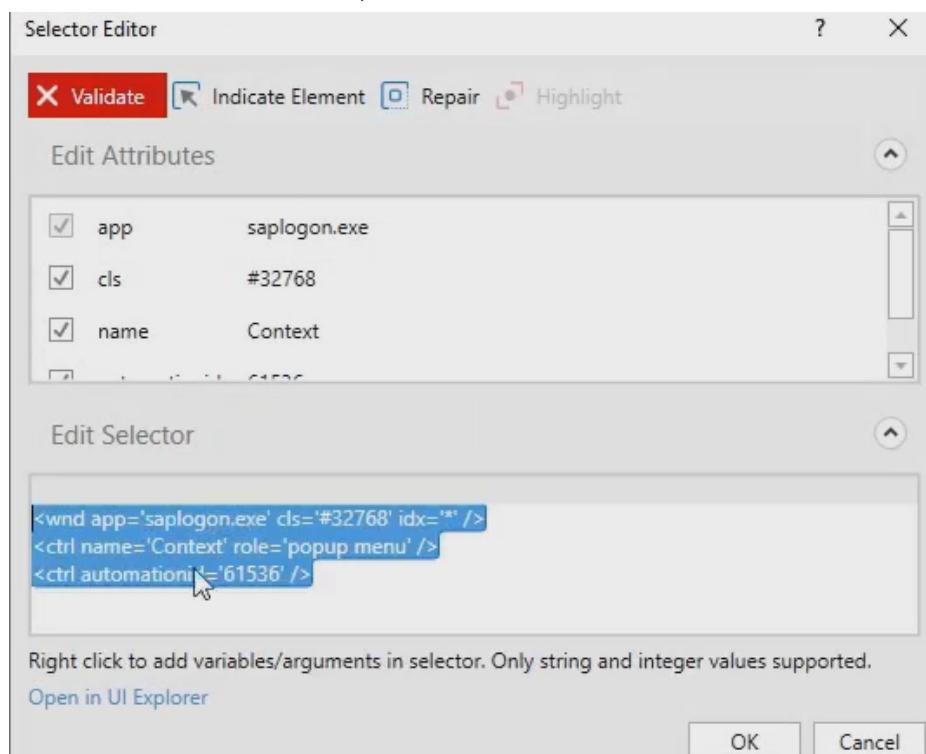


Now the menu button will be clicked along with which we have to indicate the close button from dropdown menu



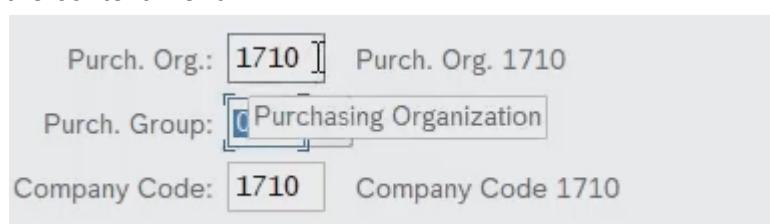
We can use F2 for a delay and then make the close button visible to get indicated.

When we check its selector, these have correct automation ID.

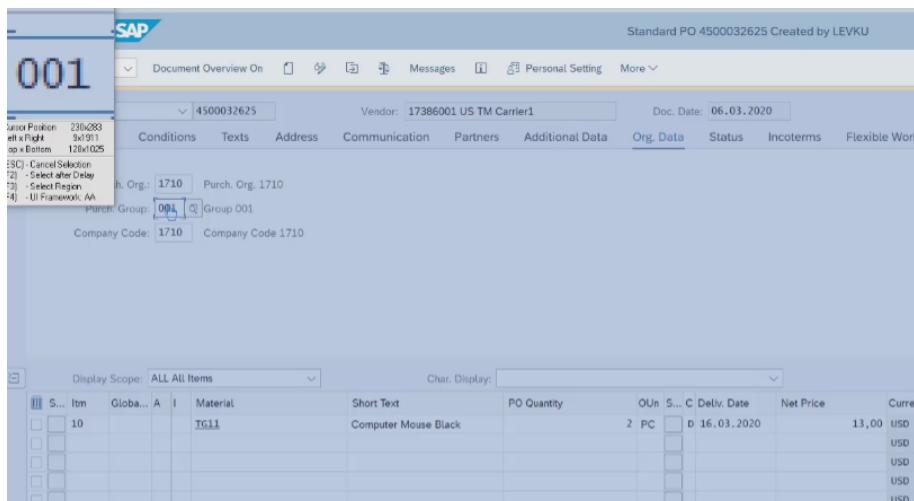


This is the way we have to interact with popups while working on SAP automation.

Let's look on another special case where we have to right click the 'purchase group' and call the context menu



Now to right click on this, indication has taken the default framework,



lets press F4 & click the field you are interested in & click the region. From Click activity property section, adjust the type to right click

Properties

JiPath.Core.Activities.Click

Common

- ContinueOnError
- DelayAfter
- DelayBefore
- Display Name: Click 'CTextField Purchasing'

Input

- ClickType: ClickType.CLICK_SINGLE
- MouseButton: MouseButton.BTN_LEFT

Target

Misc

- Private

Options

Use another click activity and indicate the dropdown items, use F2 to delay the selection.