Winshuttle Excel Add-in Metabot for Automation Anywhere

Table of Contents

Automation Anywhere Winshuttle Excel Add-in Metabot

Version

1.0 June 2020 - Initial release of Automation Anywhere Winshuttle Metabot

Terms of usage

This Metabot, and associated Taskbot examples, are provided solely as a template examples demonstrating how to integrate Winshuttle runner activities with Automation Anywhere. We recommend that users test this Metabot in a non-production environment in a non-production SAP system. The user assumes all risks related to or resulting from automating Winshuttle activities in their SAP systems.

The user of this Metabot, and associated Taskbot examples, must also act in compliance with the Winshuttle end-user license agreement, ("EULA"), and acknowledges that use of the Metabot is not included in Winshuttle's standard support and maintenance plan. Winshuttle expressly disclaims all liability with respect to the Metabot and your use thereof.

Environment

This was built and tested in an environment including the following:

- Automation Anywhere 11.3
- Winshuttle Studio 12.x Connect license
- Excel 2016, as part of Office 365
- Windows 10

Overview

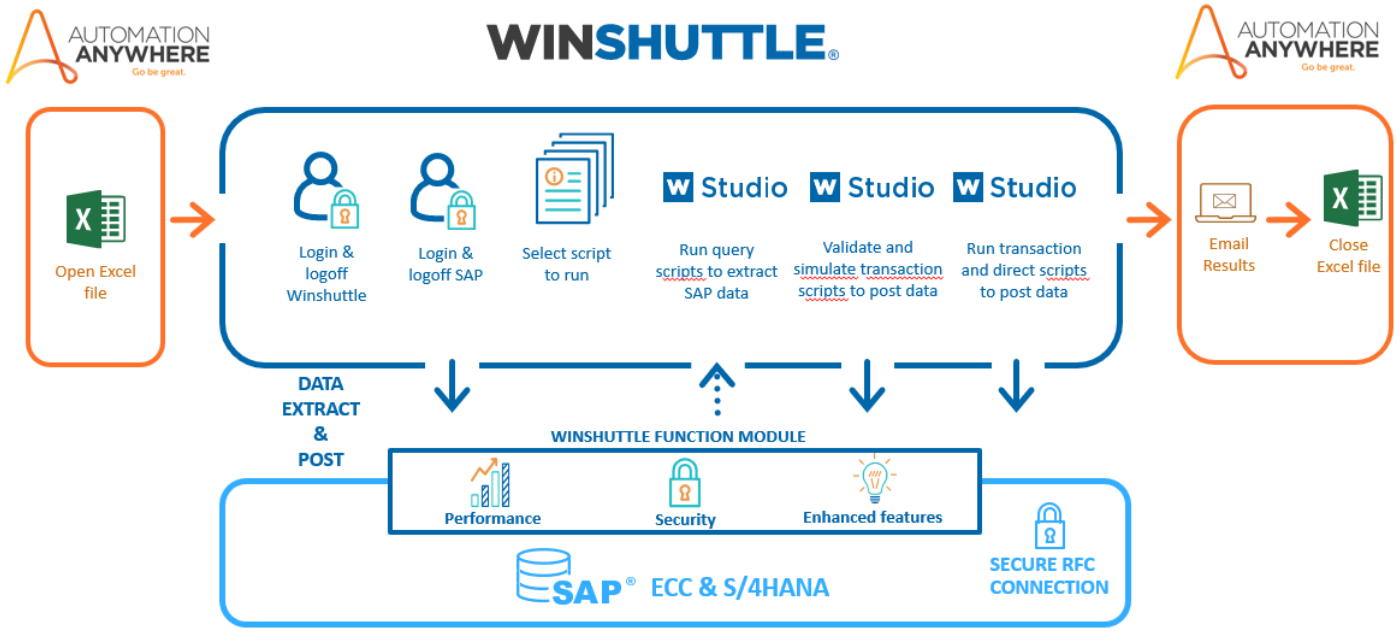
Winshuttle empowers business users to rapidly move data in and out of SAP using standard business applications. With Winshuttle products, companies improve productivity, reduce repetitive manual labor and harness the power of their data to fuel their business. Winshuttle provides integration to SAP without requiring code or screen captures.

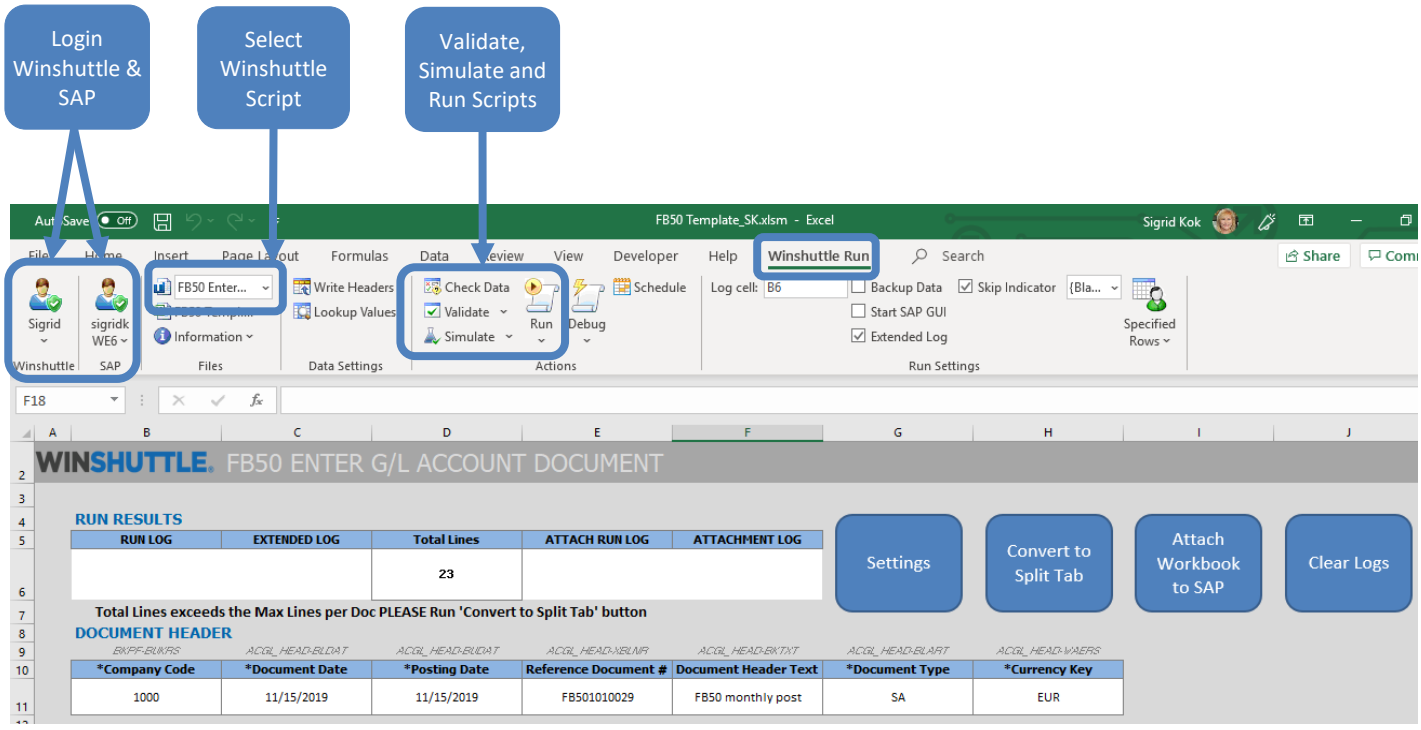
The Automation Anywhere Winshuttle Metabot is meant to provide a template demonstrating how to automate the task of a Winshuttle Runner. This version only works with the Winshuttle Connect Excel Add-in to automate tasks such as opening the Excel file, logging in to Winshuttle and SAP, and taking Script Actions, such as: Validate, Simulate and Run.

Automation Anywhere has additional capabilities to integrate with other systems to complete tasks prior to and after Winshuttle runs, and automate how files are submitted, captured and run.

The combination of Winshuttle integration to SAP and Automation Anywhere RPA capabilities is powerful. It brings together the best of both worlds, accelerating Automation Anywhere customers' integration to SAP via Winshuttle.

Below is a high-level diagram of how the Winshuttle Metabot can be used:





The Winshuttle Excel add-in Metabot works best when you follow best practices:
1. Create your Winshuttle Script(s) and test them
2. Lock each script to the appropriate Excel sheet
3. Lock the start row for each Script
4. Publish your Script(s) into the Excel file

Contacts

Contact your Winshuttle sales representative with questions related to Winshuttle software and EULAs. If you don't have a Winshuttle sales representative, please contact Winshuttle here: https://www.winshuttle.com/contact-us/

For questions about the Winshuttle and Automation Anywhere partnership, please contact Richard Rogers at the Winshuttle corporate office: +1 (800) 711-9798 or +1 (425) 368-2708

For questions about Automation Anywhere, please contact your Automation Anywhere representative or visit the Automation Anywhere website: www.automationanywhere.com

Setup

1. Install Automation Anywhere 11.3 (or compatible release)
2. Install Winshuttle version 12.x Studio. See system requirements here: https://winshuttle-help.s3.amazonaws.com/studio/en/connect-sap/help/12/system-requirements.htm
3. Ensure you have the appropriate Winshuttle Function module installed on the non-production SAP system you are testing against. See the Winshuttle documentation here: https://winshuttle-help.s3.amazonaws.com/Winshuttle-function-module/EN/12/wfm-compatibility.htm
4. Ensure you have a compatible version of the SAP GUI installed. See the Winshuttle documentation here: https://mysupport.winshuttle.com/hc/en-us/articles/360023688031-Which-SAP-GUI-versions-are-supported-by-Winshuttle-
5. Ensure you are using a supported version of SAP (listed in the pre-requisites – see #2 above).
6. Open Excel and ensure Winshuttle Excel Add-in is available, via the Winshuttle Run tab
7. Create Winshuttle scripts using Winshuttle Studio against a non-production SAP system, typically Transaction or Direct scripts. Test and publish the scripts to embed them in the Excel file.
8. Place the Excel file with the published script(s) (including data), in a centralized directory accessible by Automation Anywhere.
9. Ensure you fill in the parameters that will be passed into the Metabot.

Input Parameters

Parameter	Opt/Req	Description
inExcelFileName	Required	Excel file name, example: myfile.xlsx or myfile.xlsm
inRequestorEmail	Required	Email address of the requestor, used for notifications
inSAPSavedCredentials	Required	Winshuttle saved SAP Credentials. Example: PRD-mysapusername-100 where PRD is your SAP system ID, mysapusername is your SAP user name and 100 is the SAP client. See how to save below in the LoginSAPSavedCredentials logic.
inScriptName	Required	Winshuttle script name, as published into the Excel file. This Metabot assumes your Winshuttle scripts are published into the Excel file and locked to the sheet they will run from. During the publish step, the script is given a name. The script name is visible from the Winshuttle Excel add-in after logging onto Winshuttle. The name must be an exact match, with case and spacing important to getting a match when selecting the script. Note that if there is only one script published into the Excel file, script select is not required and thus this input parameter is not needed.
inRunLogCell	Required	The Excel cell that contains the Winshuttle script SAP Log message, example: A1
inValidateLogCell	Required	The Excel cell that contains the Winshuttle script SAP Log message, example: A1
inSimulateLogCell	Required	The Excel cell that contains the Winshuttle script SAP Log message, example: A1
inExcelMacro	Optional	Only needed if you want to run an Excel Macro and not do it in the Taskbot.
inExcelSession	Required	Excel session name, which is passed in from the Taskbot. Note that the Metabot cannot initiate opening an Excel file and pass the session name to the Taskbot. Per Automation Anywhere Support, best practice is to open Excel file session from the Taskbot and pass it to Metabot logic.

See the input parameters used in the Metabot Logic section below.

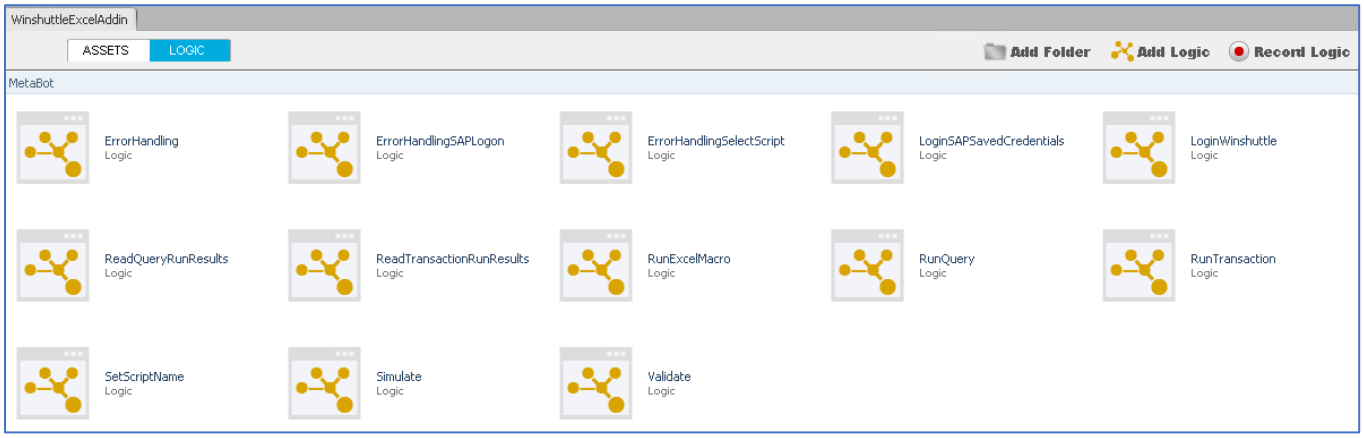
Output Parameters

Parameter	Description
outReturnCode	Return message created by logic used to share success and failures with the Taskbot.
outExtracted	Number of records extracted from a Query Script execution.
outErrors	Number of Errors from a Transaction/Direct Script execution.
outSAPRunResult	
outSuccesses	Number of Successful records posted from a Transaction/Direct Script execution.

Parameter	Description
outScriptSelected	The script selected in the SelectScriptName logic. This returns the Winshuttle Script that is currently selected and allows the Taskbot to know if the input ScriptName was published in the Excel file.
outSimulateResult	The SAP Log message from a Simulate Script execution.
outValidateResult	The SAP Log message from a Validate Script execution.

See the output parameters used in the Metabot Logic section below.

Metabot Logic

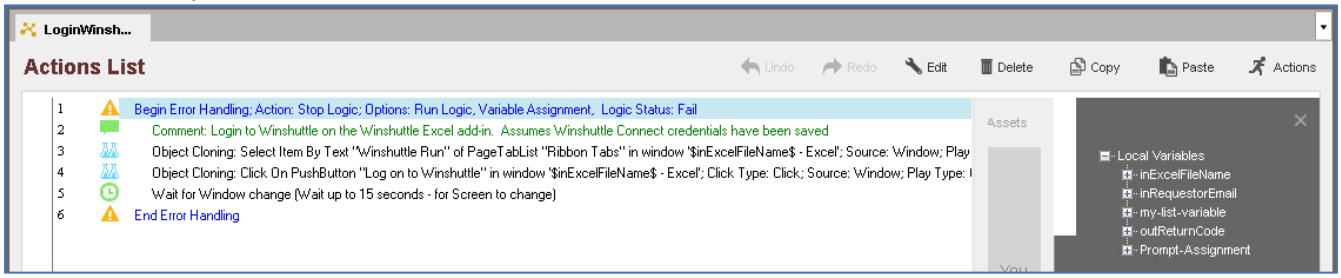


Note: The excel file is opened in the Taskbot so that it can name the Excel session and pass it to the Metabot, so that is outside of the scope of the Metabot. Checking return codes from the Metabot logic is also done in the Taskbot, as is passing input parameters. Please see the examples below.

LoginWinshuttle

Login to Winshuttle. This Logic assumes the Winshuttle Connect credentials have been saved.

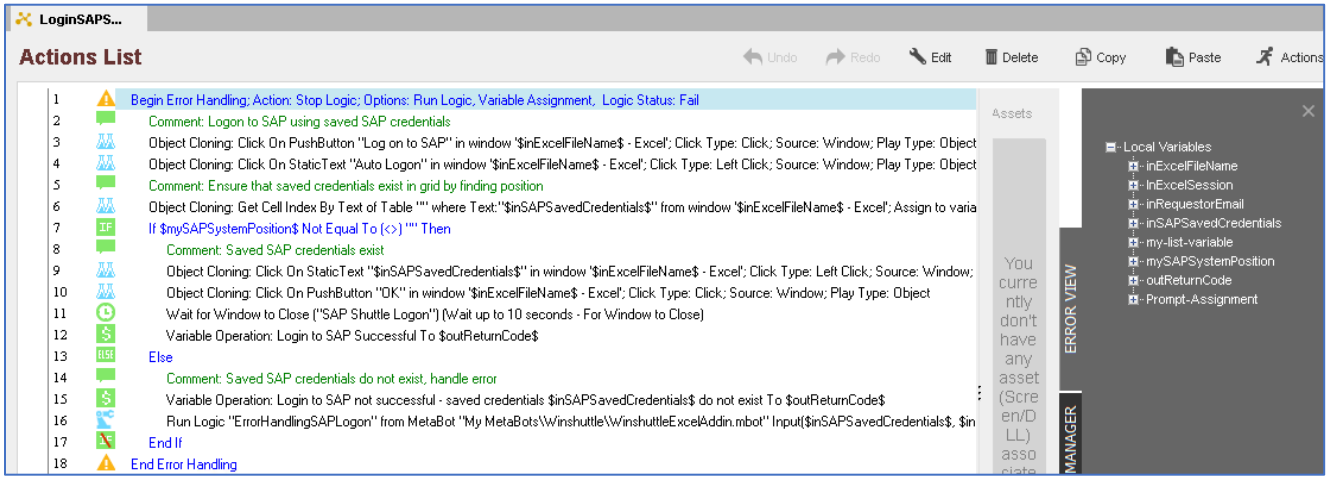
- Navigate to the Winshuttle Run tab in Excel
- Click Logon to Winshuttle and wait for it to complete
- Parameters
  - Input: inExcelFileName, InRequestorEmail
  - Output: outReturnCode



LoginSAPSavedCredentials

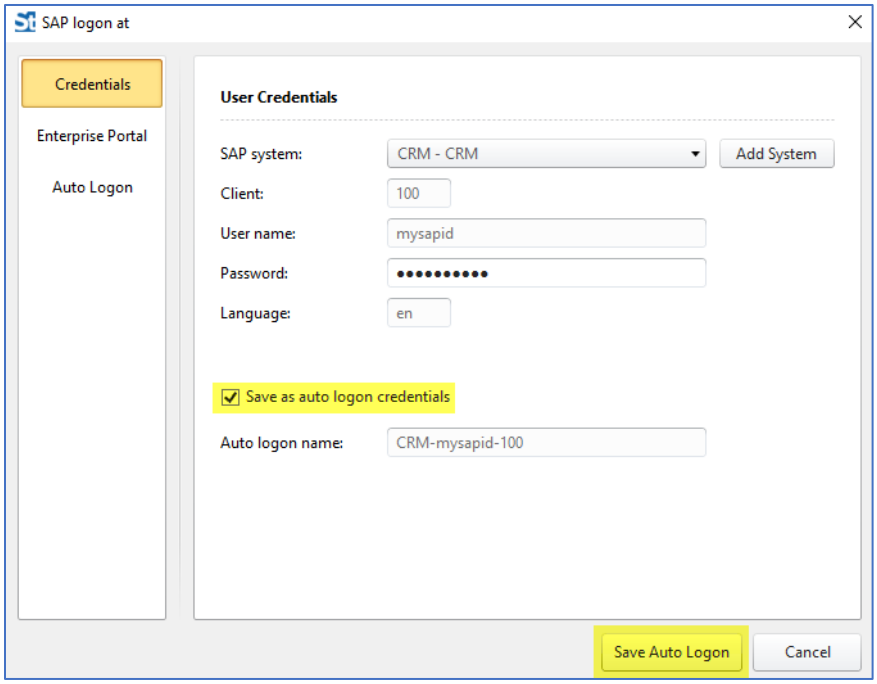
Login to SAP via Winshuttle saved credentials. This Logic assumes the Winshuttle SAP credentials have been saved, example: PRD-myusername-100, where PRD is the SAP system ID, myusername is your SAP user name and 100 is the client

- Click button to Logon to SAP
- Click Auto Logon
- Checks to see if the saved SAP credentials exist
    - If they do not exist, sets an error return code
    - If they do, selects the saved SAP credentials from list
- Parameters
    - Input: inExcelFileName, InRequestorEmail, inSavedSAPCredentials
    - Output: outReturnCode



Instructions to create saved SAP credentials in Winshuttle Studio:

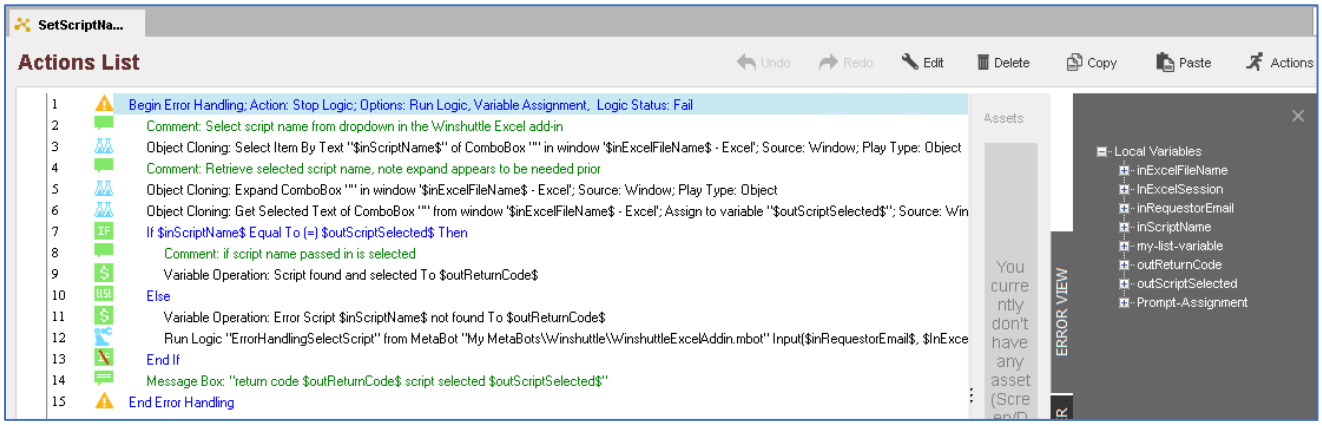
1. Open Winshuttle Studio
2. Click Account on left-hand side
3. Click Authorization Check
4. Click Credentials and enter SAP system, client, user name and password
5. Click the checkbox to “Save as auto logon credentials”
6. Click “Save Auto Logon”, noting the Auto logon name



SetScriptName

Select Script Name from Excel Add-in dropdown that you want to run.

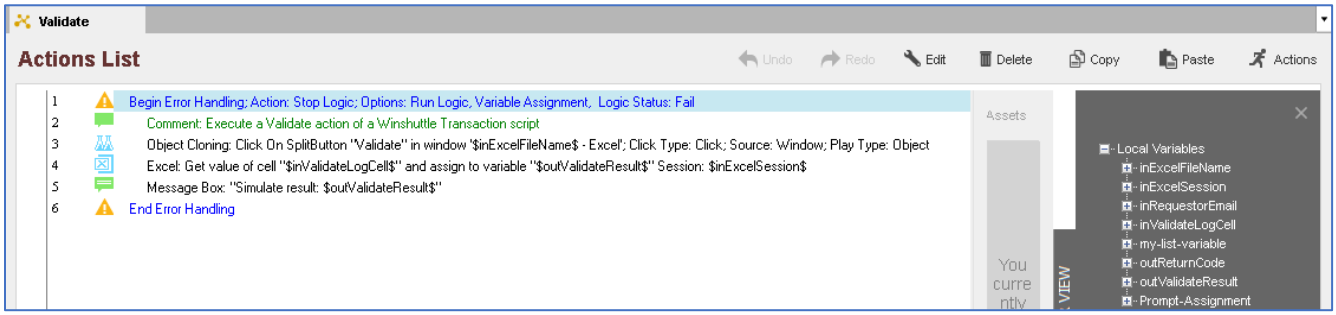
- Click on the script dropdown list to select the script name in the input parameter
- Reads the script name selected for error checking purposes
    - If the script selected matches the script input parameter, it sets a success return code
    - If the script selected does not match the script name input parameter, it sets an error return code and executes the ErrorHandlingSelectScript logic
- Parameters
    - Input: inExcelFileName, InRequestorEmail, inScriptName
    - Output: outReturnCode, outScriptSelected



Validate

Run a Winshuttle Transaction script in Validate mode, checking the data against SAP but not posting it.

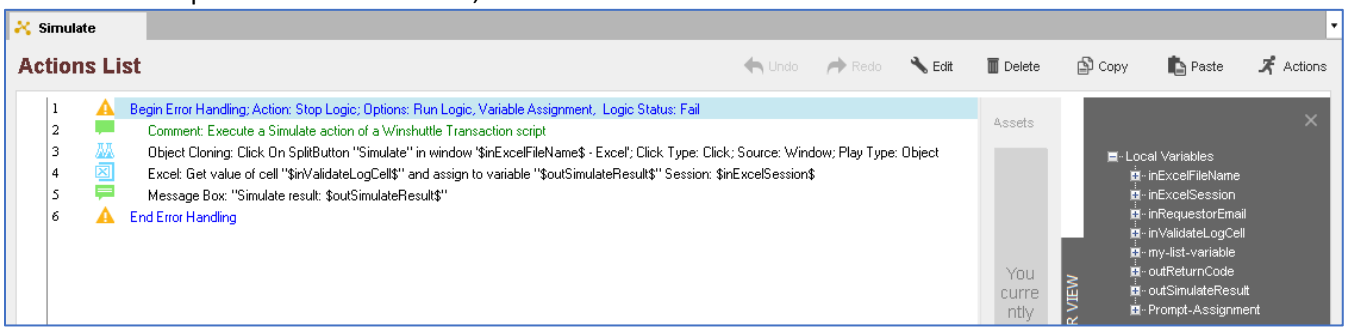
- Click on Validate button in the Excel add-in ribbon.
- Reads the results from the SAP Log Message Excel cell to return to the Taskbot.
- Parameters
    - Input: inExcelFileName, inExcelSession, inRequestorEmail, inValidateLogCell
    - Output: outReturnCode, outValidateResult



Simulate

Run a Winshuttle Transaction script in Simulate mode, checking the data against SAP but not posting it.

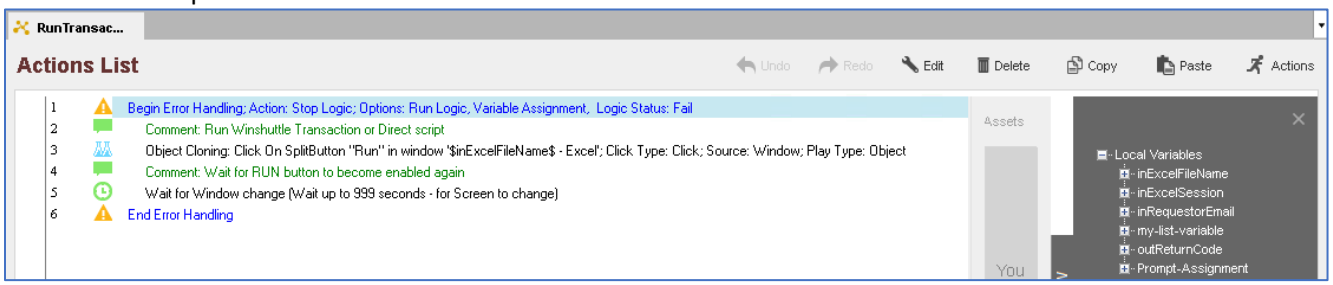
- Click on Simulate button in the Excel add-in ribbon.
- Reads the results from the SAP Log Message Excel cell to return to the Taskbot.
- Parameters
    - Input: inExcelFileName, inExcelSession, inRequestorEmail, inValidateLogCell
    - Output: outReturnCode, outSimulateResult



RunTransaction

Run a Winshuttle Transaction or Direct script to post data to SAP, and sometimes read data from SAP.

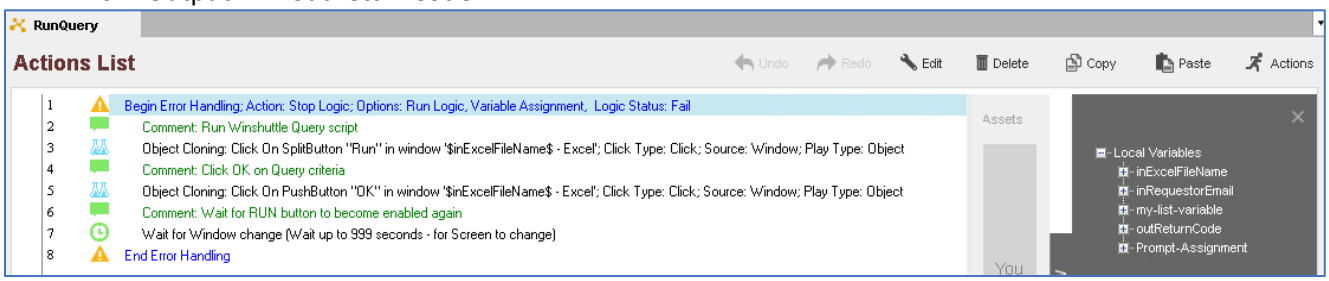
- Click on Run button in the Excel add-in ribbon.
- Parameters
    - Input: inExcelFileName, inExcelSession, inRequestorEmail
    - Output: outReturnCode



RunQuery

Run a Winshuttle Query script to extract data from SAP.

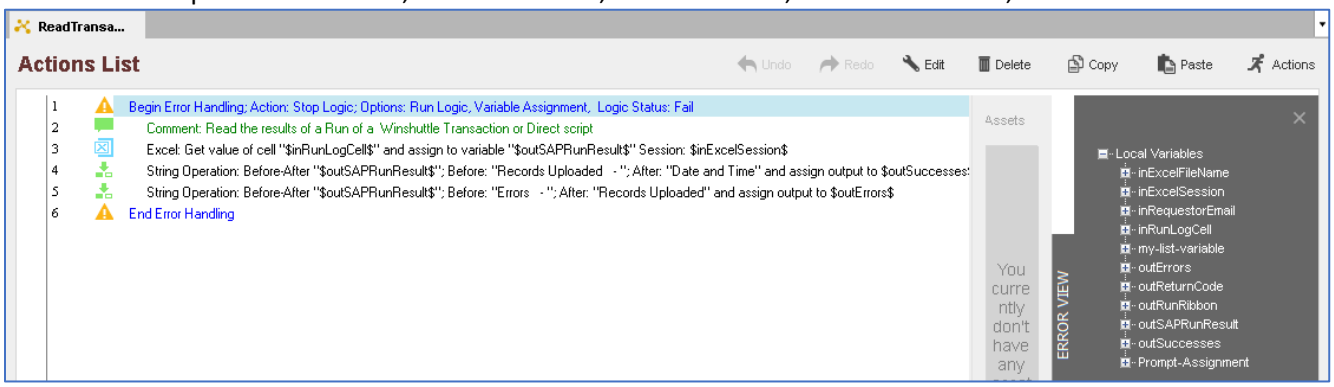
- Click on Run button in the Excel add-in ribbon.
- Click on the OK button on the Query criteria window. We assume the Query criteria are data driven. If you have to pass in specific Query criteria, you will need to handle it in this step.
- Parameters
    - Input: inExcelFileName, inRequestorEmail
    - Output: outReturnCode



ReadTransactionResults

Read results from a Winshuttle Transaction or Direct script run.

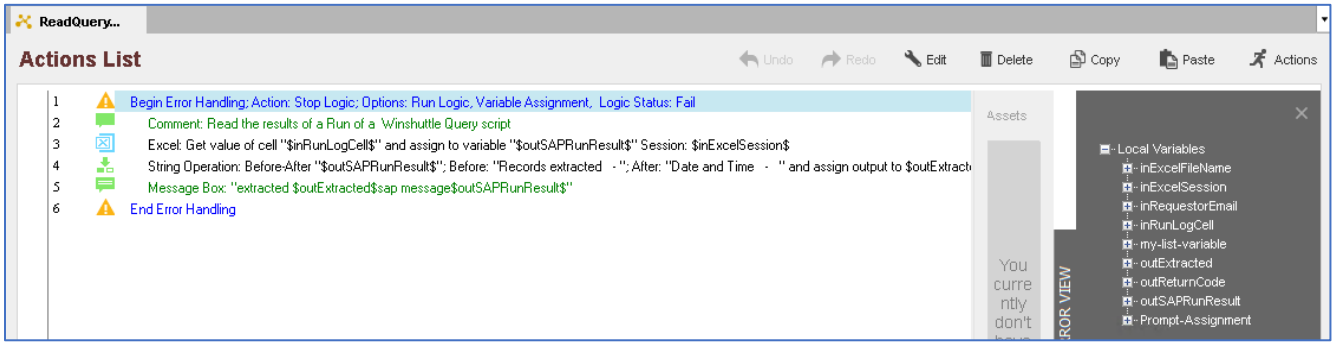
- Reads results from Excel
- Parses out of the SAP Log Message the
  - Number posted to SAP
  - Number of errors
  - The String Operations assume English is the language being used by the Metabot. Please update as needed to handle other languages
- Parameters
  - Input: inExcelFileName, inExcelSession, inRequestorEmail, inRunLogCell
  - Output: outErrors, outReturnCode, outRunRibbon, outSAPRunResult, outSuccesses



ReadQueryResults

Read results from a Winshuttle Transaction or Direct script run.

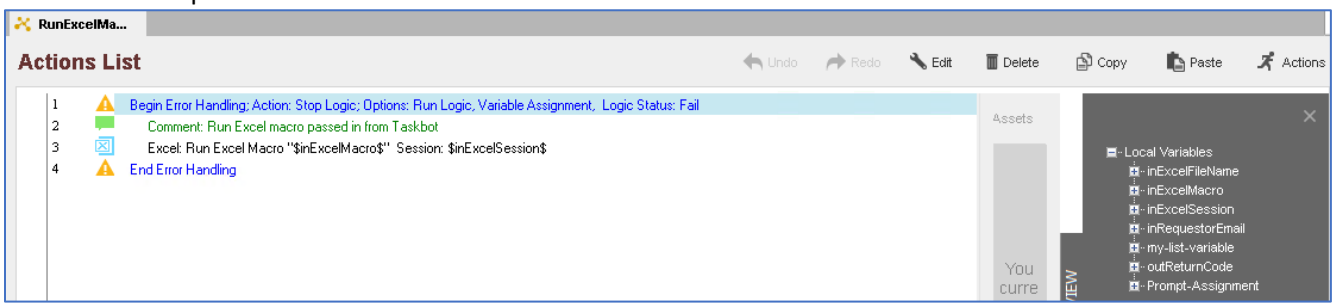
- Reads results from Excel
- Parses out of the SAP Log Message the number of extracted messages. Note that the String Operations assume English is the language being used by the Metabot. Please update as needed to handle other languages.
- Parameters
  - Input: inExcelFileName, inExcelSession, inRequestorEmail, inRunLogCell
  - Output: outExtracted, outReturnCode, outSAPRunResult



RunExcelMacro

Run Excel Macro

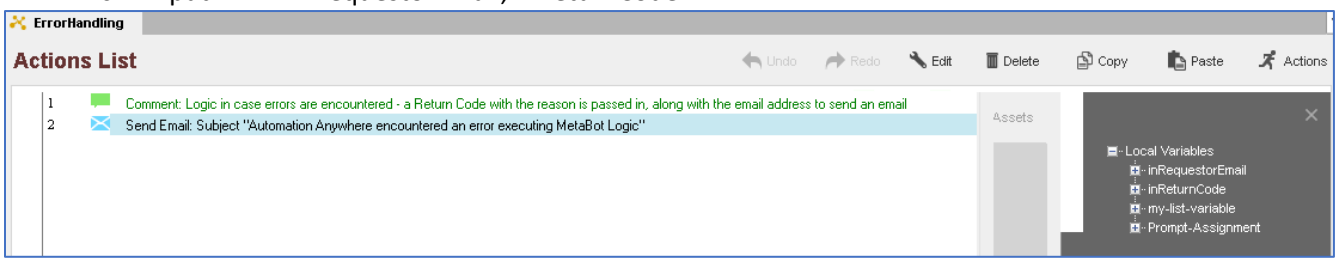
- Runs Excel Macro. Note that this can easily be done via the Taskbot, but it was added to the Metabot Logic set for consistency.
- Parameters
  - Input: inExcelFileName, inExcelMacro, inExcelSession, inRequestorEmail
  - Output: outReturnCode



ErrorHandling

Logic to run for general errors. This logic is setup to send an email, but it can be replaced with other actions.

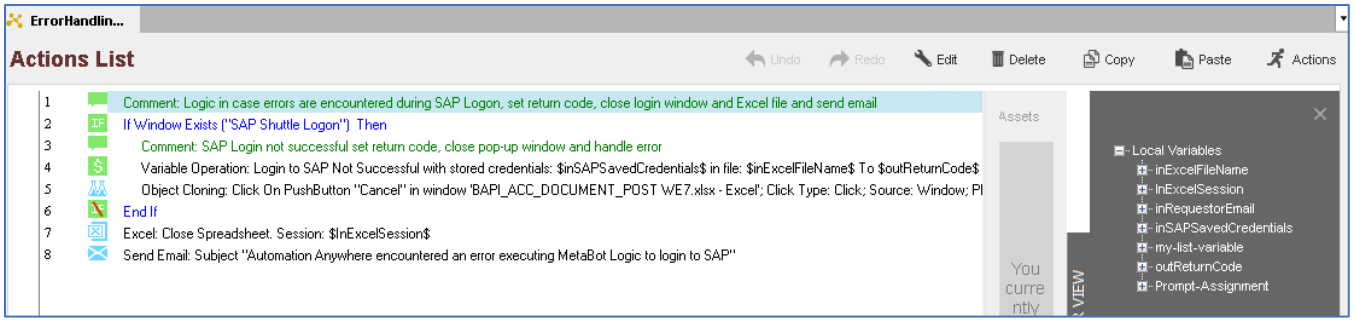
- Parameters
  - Input: inRequestorEmail, inReturnCode



ErrorHandlingSAPLogon

Logic to run for SAP Logon errors, typically encountered when invalid saved Winshuttle SAP credentials are passed as input.

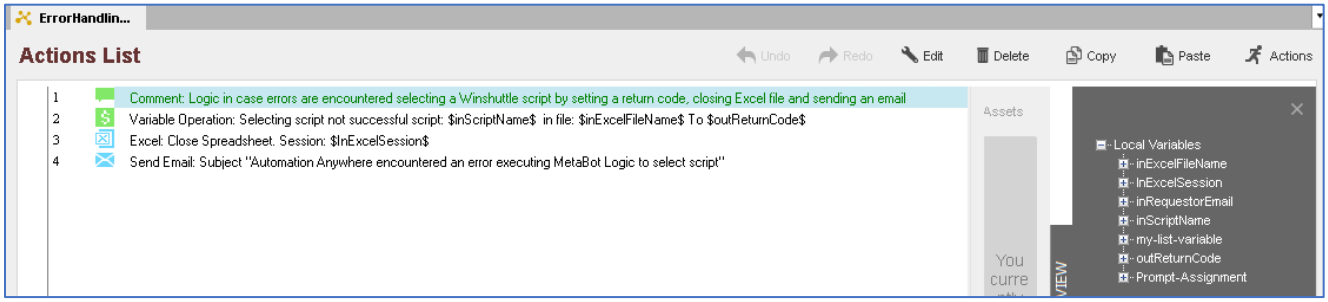
- Sets a return code with the error specifics
- If the Logon window is still open, it clicks Cancel on the login window
- Closes the Excel file to clean up the process
- Sends an error email
- Parameters
  - Input: inExcelFileName, inExcelSession, inRequestorEmail, inSAPSavedCredentials
  - Output: outReturnCode



ErrorHandlingSelectScriptLogic

Logic to run for Script selection errors, typically encountered when invalid script name is passed as input.

- Sets a return code with the error specifics
- Closes the Excel file to clean up the process
- Sends an error email
- Parameters
    - Input: inExcelFileName, inExcelSession, inRequestorEmail, inScriptName
    - Output: outReturnCode



Taskbot Examples

WinshuttleExcelAddinMetabtTransactionExample

The Transaction/Direct Task bot will leverage the WinshuttleExcelAddinMetabot to:

1. Open an Excel file
2. Clear the SAP Log message Excel cell
3. Run logic to
    a. Login to Winshuttle and SAP
    b. Select a Script (only needed if more than one script is published into the Excel file)
    c. Run the Transaction/Direct script and retrieve the results
    d. Send an email and pop-up up a message box for demonstration purposes

```
1  Begin Error Handling; Action: Stop Task; Options: Send Email,  Task Status: Fail
2      Comment: Run Transaction/Direct Script via Winshuttle Excel Add-in
3      If File Exists ("$myExcelFileNameWithPath$")  Then
4          Comment: Open Excel File
5          Excel: Open Spreadsheet "$myExcelFileNameWithPath$". ActiveSheet: "Default". Session: $myExcelSession$
6          Comment: Clear Log cell before run
7          Excel: Set value of Cell "$myLogCell$" with " ". Session: $myExcelSession$
8          Comment: Login Winshuttle
9          Run Logic "LoginWinshuttle" from MetaBot "My MetaBots\Winshuttle\WinshuttleExcelAddin.mbot" Input($myExcelFileName$, $myRequestorEmail$) Output($myReturnMessage$)
10         Comment: Login SAP with saved credentials
11         Run Logic "LoginSAPSavedCredentials" from MetaBot "My MetaBots\Winshuttle\WinshuttleExcelAddin.mbot" Input($myRequestorEmail$, $myExcelFileName$, $mySAPSavedSystem$, $myExcelSes
12         If $myReturnMessage$ Equal To (=) "Login to SAP Successful" Then
13             Comment: Choose Winshuttle script
14             Run Logic "SetScriptName" from MetaBot "My MetaBots\Winshuttle\WinshuttleExcelAddin.mbot" Input($myRequestorEmail$, $myExcelFileName$, $myScriptName$, $myExcelSession$) Output($r
15             If $myScriptName$ Equal To (=) $myScriptSelected$ Then
16                 Comment: if script selected sucessfully
17                 Comment: Run Transaction script
18                 Run Logic "RunTransaction" from MetaBot "My MetaBots\Winshuttle\WinshuttleExcelAddin.mbot" Input($myExcelFileName$, $maxruntime$, $myExcelSession$, $myRequestorEmail$, $myScrip
19                 Comment: Read results from Log Message Cell and parse
20                 Run Logic "ReadTransactionRunResults" from MetaBot "My MetaBots\Winshuttle\WinshuttleExcelAddin.mbot" Input($myExcelSession$, $myRequestorEmail$, $myLogCell$, $myExcelFileNam
21                 Comment: Close Excel File
22                 Excel: Close Spreadsheet. Session: $myExcelSession$
23                 Comment: Email Results
24                 Send Email: Subject "Automation Anywhere executed the WinshuttleExcelAddin Metabot - Script $myScriptName$ in Excel file: $myExcelFileName$" with Attachment(s).
25                 Comment: Display results for demo purposes
26                 Message Box: "SUCCESS - Winshuttle Transaction/Direct Script has completed successfully:Success $myRunSuccess$Error $myRunError$SAP Message: $mySAPRunResult$Return message
27             Else
28                 Comment: ERROR Winshuttle script not selected sucessfully
29                 Message Box: "ERROR- Selecting script $myScriptName$ in file $myExcelFileName$Return message (if any): $myReturnMessage$"
30             End If
31         Else
32             Comment: ERROR Could not login to SAP
33             Message Box: "ERROR - logging into SAP with saved credentials: $mySAPSavedSystem$Return message (if any): $myReturnMessage$"
34         End If
35     Else
36         Message Box: "ERROR - Excel file $myExcelFileNameWithPath$ does not exist"
37     End If
38 End Error Handling
```


WinshuttleExcelAddinMetabtQueryExample Taskbot

The Query Task bot will leverage the WinshuttleExcelAddinMetabot to:

4. Open an Excel file
5. Clear the SAP Log message Excel cell
6. Run logic to
    a. Login to Winshuttle and SAP
    b. Select a Script (only needed if more than one script is published into the Excel file)
    c. Run the Query script and retrieve the results
    d. Send an email and pop-up up a message box for demonstration purposes

```
1  Begin Error Handling; Action: Continue; Options: Send Email,  Task Status: Fail
2      Comment: Run Query Script via Winshuttle Excel Add-in
3      If File Exists ("$myExcelFileNameWithPath$")  Then
4          Comment: Please enter the conditional commands here.
5          Comment: Open Excel File
6          Excel: Open Spreadsheet "$myExcelFileNameWithPath$". ActiveSheet: "Default". Session: $myExcelSession$
7          Comment: Clear Log cell before run
8          Excel: Set value of Cell "$myLogCell$" with " ". Session: $myExcelSession$
9          Comment: Login Winshuttle
10         Run Logic "LoginWinshuttle" from MetaBot "My MetaBots\Winshuttle\WinshuttleExcelAddin.mbot" Input($myExcelFileName$, $myRequestorEmail$) Output($myReturnMessage$)
11         Comment: Login SAP with saved credentials
12         Run Logic "LoginSAPSavedCredentials" from MetaBot "My MetaBots\Winshuttle\WinshuttleExcelAddin.mbot" Input($myExcelSession$, $myExcelFileName$, $myRequestorEmail$, $mySAPSystem$) I
13         Comment: Choose Winshuttle script
14         If $myReturnMessage$ Equal To (=) "Login to SAP Successful" Then
15             Comment: Please enter the conditional commands here.
16             Run Logic "SetScriptName" from MetaBot "My MetaBots\Winshuttle\WinshuttleExcelAddin.mbot" Input($myScriptName$, $myExcelSession$, $myExcelFileName$, $myRequestorEmail$) Output($n
17             If $myScriptSelected$ Equal To (=) $myScriptName$ Then
18                 Comment: Please enter the conditional commands here.
19                 Comment: Run Transaction script
20                 Run Logic "RunQuery" from MetaBot "My MetaBots\Winshuttle\WinshuttleExcelAddin.mbot" Input($myRequestorEmail$, $myExcelFileName$) Output($myReturnMessage$)
21                 Comment: Read results from Log Message Cell and parse
22                 Run Logic "ReadQueryRunResults" from MetaBot "My MetaBots\Winshuttle\WinshuttleExcelAddin.mbot" Input($myExcelSession$, $myExcelFileName$, $myRequestorEmail$, $myLogCell$) Ou
23                 Comment: Close Excel File
24                 Excel: Close Spreadsheet. Session: $myExcelSession$
25                 Comment: Email Results
26                 Send Email: Subject "Automation Anywhere executed the WinshuttleExcelAddin Metabot - Script $myScriptName$ in Excel file: $myExcelFileName$" with Attachment(s).
27                 Comment: Display results for demo purposes
28                 Message Box: "SUCCESS - Winshuttle Query Script has completed successfully:Number Extracted: $myRunExtracted$SAP Message: $mySAPRunResult$Return message (if any): $myReturnMe
29             Else
30                 Comment: ERROR Winshuttle script not selected sucessfully
31                 Message Box: "ERROR - Selecting script $myScriptName$ in file $myExcelFileName$Return message (if any): $myReturnMessage$"
32             End If
33         Else
34             Comment: ERROR Could not login to SAP
35             Message Box: "ERROR - logging into SAP with saved credentials: $mySAPSavedSystem$Return message (if any): $myReturnMessage$"
36         End If
37     Else
38         Comment: ERROR File does not exist
39         Message Box: "ERROR - Excel file $myExcelFileNameWithPath$ does not exist"
40     End If
41 End Error Handling
```


About Winshuttle

Winshuttle software empowers business teams to make an impact through solutions that make it quick and easy to exchange data with SAP using Excel, streamline SAP business processes using forms and workflows, and improve data quality using data stewardship capabilities.

Its business led, IT-enabled solutions enable users to automate processes and solve problems without compromising security or governance. Business teams can author solutions across lines of business and the SAP landscape, speeding product launches and financial accounting processes, streamlining customer and vendor onboarding, improving plant maintenance efficiency, tackling data migration projects, and much more.

Winshuttle offers two flexible data management platforms. Winshuttle Studio enables business teams to author Excel-based solutions that eliminate manual data entry in SAP and accelerate master and transactional data tasks. Winshuttle Foundation is an enterprise workflow solution with deep SAP integration capabilities built in, enabling business teams to streamline SAP processes and extract more value from your ERP system.

Both platforms are certified for use with SAP ECC and SAP S/4HANA and enable business teams to enhance and protect one of your most valuable strategic assets—your data.

Learn more about Winshuttle's SAP data management solutions by visiting www.winshuttle.com.

