

WINSHUTTLE.

Automating an end to end
process using a Winshuttle

Excel Workflow

Excel Workflows enable businesses to automate end-to-end process data, including conditional routing and automatically running SAP transactions – keeping track of history along the way.

Table of Contents

System Configuration Prerequisites.....	2
Overview	3
Use Case	3
Prerequisites	3
High level steps	3
Excel Data Template	4
Excel Advanced Properties.....	4
Data File Library	5
Manually create a Data File Library	5
Designing an Excel Workflow Solution.....	8
Where to see the Data File Library	10
The Workflow.....	11
Workflow Node Reference.....	11
Overview of an example journal entry workflow	11
Workflow Screen Shots/Details	13
BAPI_ACC_DOCUMENT_POST	13
Post Loop Subprocess	18
Resolve Errors Loop Subprocess	22
Data File Upload Loop Subprocess.....	25
Validate and Deploy a Composer Solution.....	25
Deploying a solution	27

System Configuration Prerequisites

- Winshuttle Workflow 11.1 (or higher)
- Winshuttle SAP Integration Server 11.1 (or higher)
- Winshuttle Foundation 11.1 (or higher)
- Winshuttle LMS 11.1 (or higher)
- Winshuttle Studio 11 (or higher)
- Winshuttle Composer 11.1 or higher (Excel workflow was not supported with Composer 11.0.3)

In addition:

- SVService must be running for SVService job type '**linkeddatafileformprocess**'. If you have multiple webapps, SVService for each Webapp is required.
- The **Process Controlled Form List feature** must be activated for the Foundation site (see below).

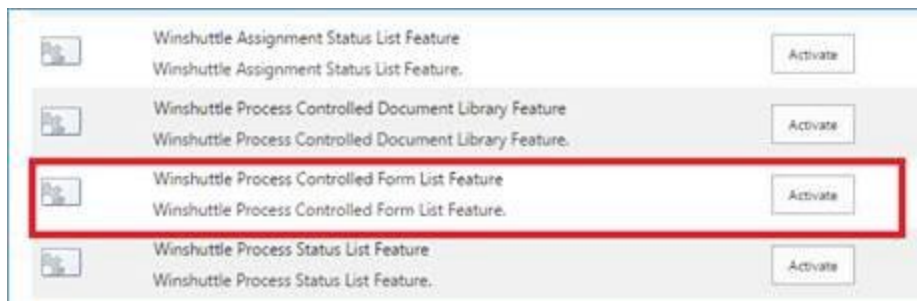
Activating the Process Controlled Form List Feature

This feature is activated automatically as soon as the Workflow Central administration site is associated in Winshuttle Foundation Site.

Activating this feature creates a standard document library with name **Forms**.

Note: To see assigned tasks for an Excel Workflow in the Studio Excel Add-in, the Workflow Admin site URL must be added to Winshuttle Foundation, under Workflow Configuration.

1. Open your Foundation site.
2. Open **Site Settings**.
3. Click **Manage Site Features**.
4. Next to Winshuttle Process Controlled Form List Feature, click **Activate**.



Overview

With Winshuttle Foundation an Excel data file can be routed using a workflow.

Use Case

Journal Entry that uses the BAPI_ACC_DOCUMENT_POST SAP BAPI to upload journal entry transaction data into SAP and two transaction scripts that use FB03 to upload the data template and supporting document to SAP.

Prerequisites

These instructions assume you already know how to create Winshuttle Studio scripts. See the links below for more information:

[Create a Script for Transaction](#)

[Create a Script to Query SAP](#)

High-level steps

1. [Excel data template](#) is created
2. Winshuttle Studio scripts (Transaction and/or Query) relating to the template are created and mapped to Excel data template (see [Prerequisites](#) for more information)
3. [Data File Library](#) is created
4. [Design an Excel Workflow](#)
5. Winshuttle Studio script is submitted to Foundation
6. Script approval task is automatically generated by Foundation
7. Custom workflow created above is selected during approval task completion (this is what ties everything together)

NOTE: Steps 1 – 4 must be completed before step 5.

Excel Data Template

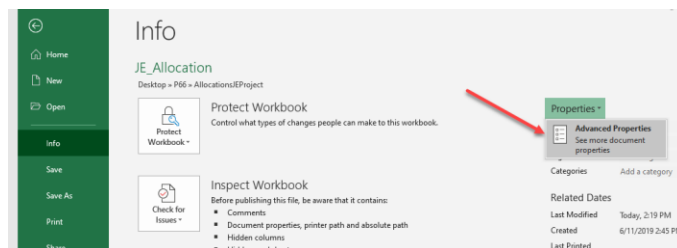
Excel Data Templates are created to provide runners a pre-formatted data template. About the template:

- Formatted data columns and/or cells set up and mapped to the Winshuttle Transaction/Query scripts used with the template
- May contain instruction/read me tab, lookup data, etc.
- May contain [Excel Advanced Properties](#) to be synced with the workflow.
- The Winshuttle runner will download the Excel Data Template from the when it is time to launch a new workflow.
- The Excel workflow will route the template.

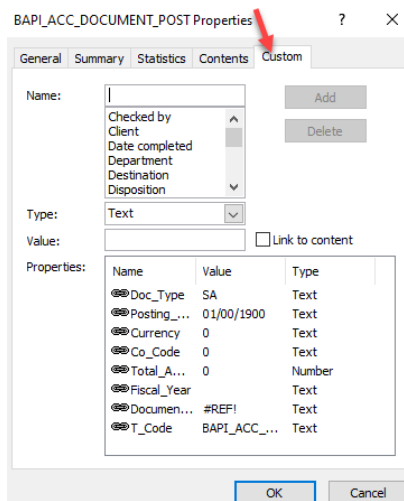
Excel Advanced Properties

Excel Advanced Properties can be synced to the workflow.

- Consideration - Only promote key properties containing data required for things like
 - determining workflow routing
 - reporting/auditing
- Named Ranges identified in Excel (list can be seen in the Formulas ribbon, Name Manager) can be promoted to the Winshuttle workflow.
- To make named ranges available for the workflow, you must link advanced properties to the named range. Go to File, Info, Properties, Advanced Properties, Custom.



- **IMPORTANT NOTE:** Excel Property names and types must match the Excel workflow field names exactly (names are case sensitive), so plan accordingly to avoid data type mismatches with the workflow and Winshuttle scripts. Excel property name examples:



Data File Library

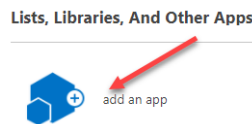
Excel Workflow (Foundation) uses a SharePoint Data File Library to store completed Excel Data Files. Winshuttle automatically includes a list of fields by default. However, fields from the Excel Data Template's Advanced Properties can be also be added to the Library and synced.

You have the option to create a Data File Library before creating your workflow, or the workflow will automatically create a new Data File Library for you with the default fields. In this example, we are going to create the Data File Library before we create the workflow.

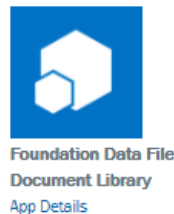
Manually create a Data File Library

Steps:

1. Open Foundation
2. Go to Site Contents
3. Add an app



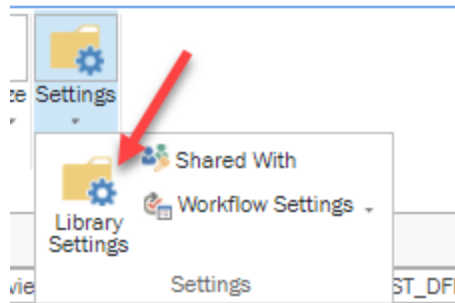
4. Foundation Data File Document Library



5. Enter information about the Data File Library and then click Create

Name and Description Type a new name as you want it to appear in headings and links throughout the site. Type descriptive text that will help site visitors use this document library.	Name: BAPL_ACC_DOCUMENT_POST_DFL
Document Version History Specify whether a version is created each time you edit a file in this document library. Learn about versions.	Description: Data File Library for BAPL_ACC_DOCUMENT_POST Excel workflow
Document Template Select a document template to determine the default for all new files created in this document library.	Create a version each time you edit a file in this document library? <input checked="" type="radio"/> Yes <input type="radio"/> No
	Document Template: Microsoft Excel spreadsheet
	<input type="button" value="Create"/> <input type="button" value="Cancel"/>

6. If there are Excel Advanced Properties you would like to sync to fields, add the fields to the new Data File Library. Steps:
 - a. Open Settings, Library Settings



b. Note: The default fields created in the new library are displayed under Columns:

Columns

A column stores information about each document in the document library. Because this document library allows document library:

Column (click to edit)

[Approval date](#)

[Approval error](#)

[AutoRun](#)

[Comments](#)

[Created](#)

[Data file workflow history.](#)

[Data post status](#)

[Data review process](#)

[Description](#)

[Due Date](#)

[Error rows](#)

[File As Attachment](#)

[FileId](#)

[FileInfo](#)

[Modified](#)

[Reminder](#)

[Reviewer](#)

[Run Date Time](#)

[Run Reason](#)

[Runner](#)

[SAP Client](#)

[SAP System ID](#)

[Schedule Date Time](#)

[Sheet Name](#)

[Shuttle File Path](#)

[Shuttle File Version](#)

[SolutionProps](#)

[Status](#)

[Supporting Documents](#)

[Supporting Documents List](#)

[System Post](#)

[T-Code](#)

[Title](#)

[Total rows](#)

[Created By](#)

[Modified By](#)

Type

Date and Time

Single line of text

Single line of text

Multiple lines of text

Date and Time

Multiple lines of text

Single line of text

Choice

Single line of text

Date and Time

Single line of text

Yes/No

Guid

Multiple lines of text

Date and Time

Choice

Person or Group

Date and Time

Single line of text

Person or Group

Single line of text

Single line of text

Single line of text

Single line of text

Single line of text

Single line of text

Multiple lines of text

Choice

Hyperlink or Picture

Multiple lines of text

Single line of text

Single line of text

Single line of text

Single line of text

Person or Group

Person or Group

c. Click Create column to create columns for the Excel data to be synced.

Important: Column name and data type must exactly match the Excel Advanced Property (case sensitive) for the data to sync correctly.

1. Tips:

- a. Have your **Excel Advanced Properties** open as a reference as you are creating columns.
- b. Workflow Status
 - i. A field called **Status** will be automatically added by default. The workflow will automatically update the Status field during the workflow.
 - ii. If you would like to have custom status messages to communicate to users, consider adding a field to update the workflow status along the way – for example, Workflow_Status.



d. Example:

Name and Type
Type a name for this column.

Column name:

The type of information in this column is:
 Single line of text
 Multiple lines of text
 Choice (menu to choose from)
 Number (L, S, D, 100)
 Currency (\$, ¥, €)
 Date and Time

Additional Column Settings
Specify detailed options for the type of information you selected.

Description:

Enforce unique values:
 Yes No

Maximum number of characters:

Default value:
 Text Calculated Value

Column Validation

Delete OK Cancel

Designing an Excel Workflow Solution

1. In Composer, open the main menu and then click **New**.
2. Click **Solution Wizard**.

The screenshot shows the 'New Solution' dialog box. On the left, a 'Steps' sidebar lists: 1. Solution Start (selected), 2. Swim Lanes, 3. Data Connections, 4. Field Packs, 5. Scripts / Web Services, and 6. Finish. The main form has the following fields: 'NAME *:' with a text input box; 'TYPE OF SOLUTION *:' with radio buttons for 'FORM WORKFLOW' and 'EXCEL WORKFLOW' (selected); 'FOUNDATION SITE *:' with a dropdown menu; and 'DATA FILE LIBRARY:' with a dropdown menu showing '- Select -'. Callout boxes provide labels: 'Solution name' for the name field, 'Type of Solution (Form or Excel-based)' for the radio buttons, 'Foundation site URL' for the dropdown, and 'Select an existing SharePoint ldata file library, or create a new one.' for the data file library dropdown. At the bottom are buttons for 'Previous', 'Next', 'Finish', and 'Cancel'.

3. In the **New Solution Wizard** dialog box, under **Name**, type a name for the solution. (*Required)
4. Select **Excel Workflow**: Create an Excel-based Workflow in Winshuttle Composer. Excel-based workflows and their associated tasks are handled entirely through Microsoft Excel.
5. In the **Foundation Site** field, select a Foundation site from the menu, or type the URL to your Foundation site. This will be used for the SharePoint Site element property.
6. In the **Data File Library** field, select the SharePoint data file library created earlier from the drop down. (Or, you can choose **Create a New Library** to create a new one.) **Note:**
 - Data File Libraries of the type 'Foundation Data File Document Library' are populated in this property.
 - You can select the 'Data File Sync Columns' if you want to route the workflow based on a cell value in the data file. (For more information, see [Using Data File Sync columns](#), below.)
7. Click **Next**.

Note: Swim Lanes, Data Connections, Field Packs, and Scripts/Web Services can all be created later if you do not wish to specify them now.

8. **Swim Lanes(optional).**

Under Swim Lanes, click **Add**. Select the Type, Role, and then type a name for the Swim Lane. See [Working with Swim Lanes](#) for more information. Click **Next**.

9. **Data Connections (optional).**

Click **Add** to Add Data Connections. See [Adding Data Connections](#) for instructions on how to add different types of data connections. Click **Next**.

10. **Scripts/Web Services (optional).**

Click **Add** to add Transaction or Query scripts. See [Importing Transaction/Query scripts](#) for more information.

11. Click **Finish**.

12. **Publish the Workflow.**

13. If you need to publish the same Excel Workflow on multiple Data File libraries, rename the Excel Workflow solution, change the Data File library, and then publish the workflow. (See [Republishing Excel Workflow solutions](#), below.)

Now whenever a Data File is added in the library specified in the **Data File Library** property, the Excel Workflow will start automatically.

See [Design Considerations](#) (below) for special information on how certain nodes, properties, and plug-ins behave in Excel Workflows.

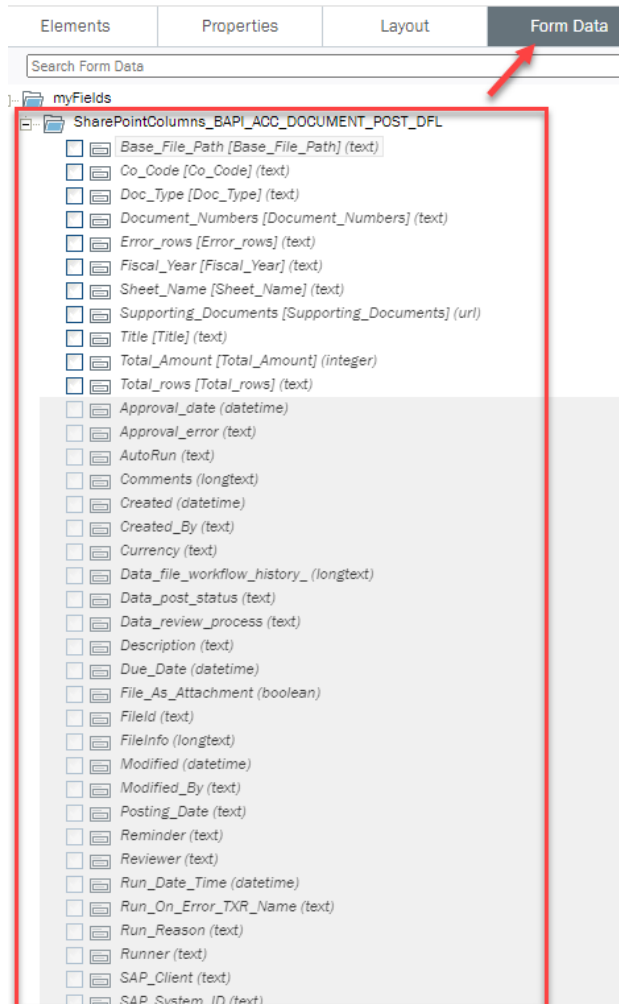
Important notes

- For an Excel Workflow to start automatically, SVService should be running for the SVService job type '**linkeddatafileformprocess**'.
- An Excel Workflow is triggered only if the data type for the submitted data file is **datafiledoc**.

Tasks related to an Excel Workflow can be completed from a supported Web browser, the Winshuttle Studio Microsoft Excel add-in, or using email.

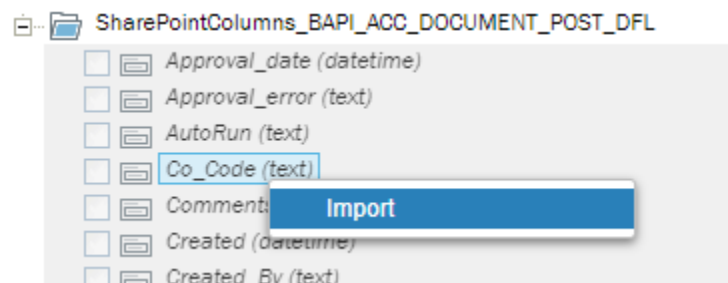
Where to see the Data File Library

The Data File Library for the solution can be seen in the Form Data tab. Example:



IMPORTANT NOTES: Additional steps will be required to enable fields to be used with additional features. Examples:

1. To enable the workflow to write to a field, right click on the field and then click Import.



Other options once a field has been Imported

To [Promote a field to the form SharePoint](#) list, create a field [available for Reporting](#).

The Workflow

[Workflow Node Reference](#) – Refer to [Workflow nodes \(A-Z\)](#) to see details about individual nodes. Workflow nodes are created to define the steps and route of the workflow. In the instructions below, we will provide tips worth noting for the workflow.

Overview of an example journal entry workflow

Once the Originator completes the data and creates a supporting document zip file, if applicable, he/she uploads or submits the completed journal entry Excel data file and supporting document to Foundation from the Winshuttle Foundation ribbon. A workflow will then be launched.

An example of a workflow is below. It is important to note, all workflow requirements are different and there may be multiple ways to accomplish the same task. This example does not require any approvals. You are encouraged to use this example to come up with the workflow that works with your scenario.

Overview

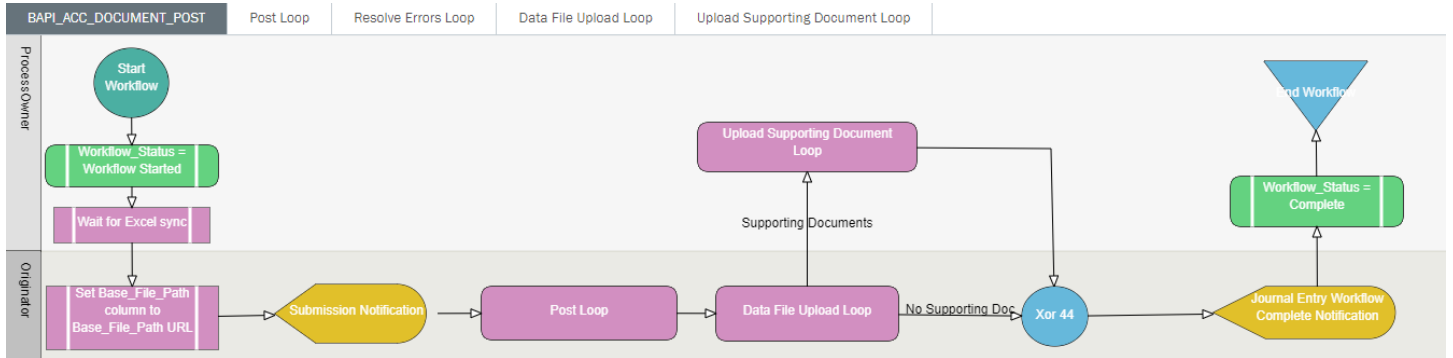
- [BAPI_ACC_DOCUMENT_POST](#) process starts
 - Syncs data from Excel to the workflow
 - NOTE: A wait node is the first step in the workflow to pause the workflow while the data is being synced.
 - Adds a URL link to the Excel Data Field
 - Sends a notification to the Originator letting him/her know the workflow has started.
 - [Post Loop subprocess](#) starts
 - Executes a journal entry upload Transaction script that will upload journal entries that don't contain errors to SAP
 - If there are any errors, the [Resolve Errors Loop subprocess](#) starts
 - the workflow will assign a task to the originator to fix the errors
 - once errors are resolved the originator can resubmit to Foundation
 - upload script is executed only on error lines and process repeats until there are no errors This feature, post only error rows is not there in 11.x version. It was introduced in 12.1 version. In earlier versions, user had to rerun all the records again. In 12.1, we included RunOnlyErrors, RunOnlyUnprocessed, StopRunOnError options. These options were not available in earlier versions of Foundation.
 - if errors have been resolved, the Resolve Errors Loop subprocess ends
 - The workflow returns to the Post Loop and ends the Post Loop
 - The workflow returns to the BAPI_ACC_DOCUMENT_POST process
 - The [Data File Upload Loop subprocess](#) starts
 - attaches the completed data template to the document numbers in SAP
 - if there is an error, a notification is sent to the Originator

- if there is no error or if the notification is complete, the Data File Upload Loop ends
 - The workflow returns to the BAPI_ACC_DOCUMENT_POST process
 - If there is a Supporting Document, the [Upload Supporting Document Loop subprocess](#) starts
 - attaches the supporting document to the document numbers in SAP
 - if there is an error, a notification is sent to the Originator
 - if there is no error or the notification has been sent, the Upload Supporting Document Loop ends
 - If there is not a Supporting Document or the Supporting Document Loop has ended
 - a notification is sent to the Originator letting him/her know the workflow has ended.
 - attaches supporting documents to the document numbers in SAP
 - notifies the originator the submission is complete
- NOTES:
 - The Workflow_Status field is updated at key steps along the way to give users quick visibility of the current status of the workflow.
 - In this example, we are using a Value Setter node to set the Workflow_Status field. However, there are many other Nodes/options for setting field values...look for Set Values in the properties of Nodes.

Set Values (0 Items) ☰

Workflow Screen Shots/Details

BAPI_ACC_DOCUMENT_POST - The main process that controls the entire workflow



Node Tips:



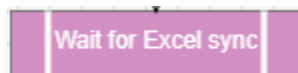
Value Pair Collection Editor

Add

Form Field : Workflow_Status "Workflow Started" [icon] [trash]

OK Cancel

- Use double quotes around text



Action	
Continue Without Waiting	False
Plug-In Name	Wait
Plug-In Parameters	(Collection)

- **Continue Without Waiting** set to False means the workflow will wait until the current step (running in the background) is complete before continuing.

Plugin Arguments Editor

Validate

Parameter	Value
serviceidcolumn	
until *	0.00.00:05

OK Cancel

- Pause the workflow for 5 seconds to allow time for the **Excel Advanced Properties** to be synced to the workflow.

Set Base_File_Path column to Base_File_Path URL

- Set Title column in the SharePoint form list to the Base_File_Path URL - puts a link to the Excel data file in the form list.

Action	
Plug-In Name	Set Form Data
Plug-In Parameters	(Collection)

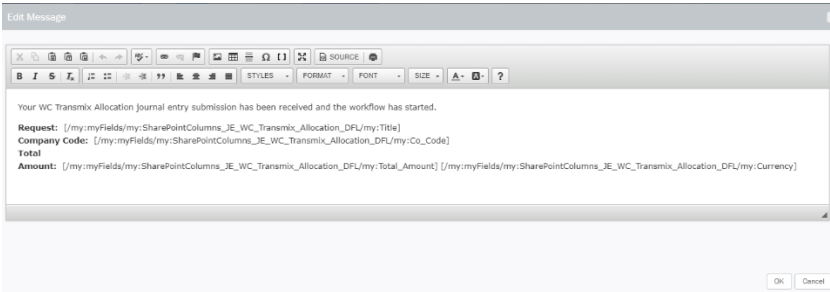
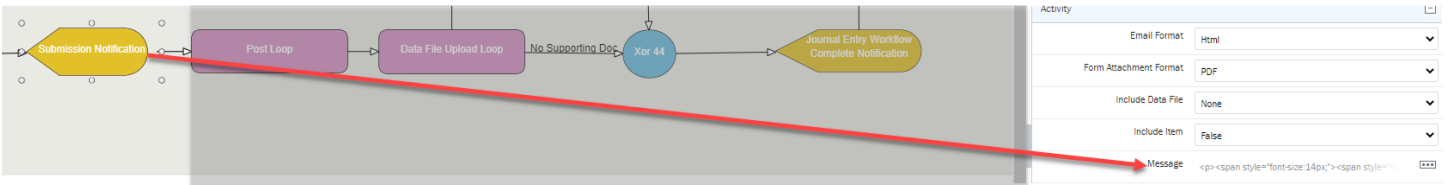
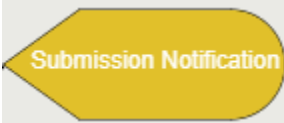
Plugin Arguments Editor

Validate Add

Parameter	Value
xpath	Base_File_Path
value	http://.../_WC_Transmix_Allocation/Title.aspx

Enter the form list URL here

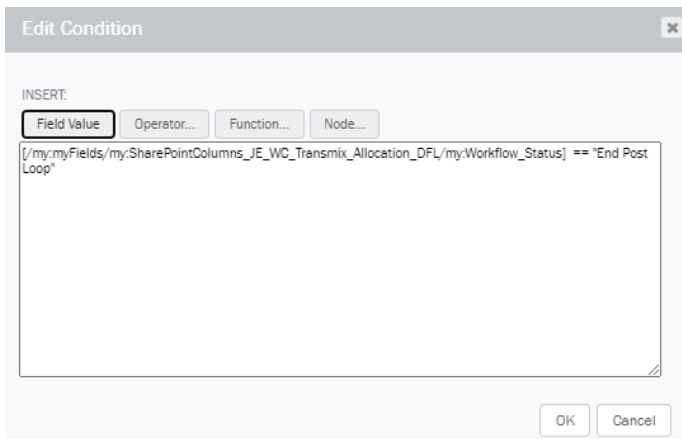
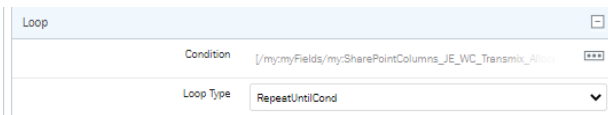
OK Cancel

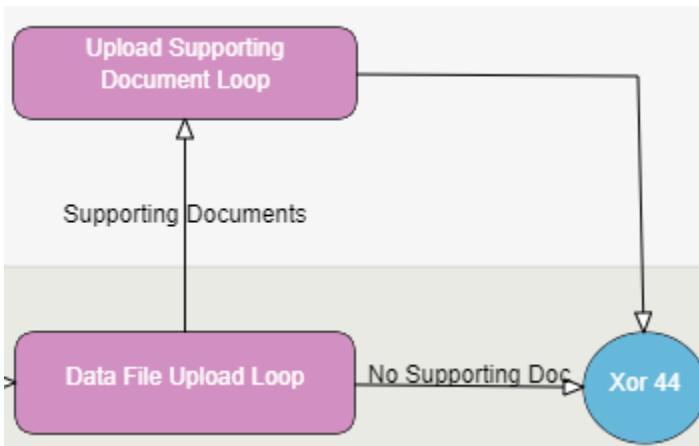


NOTE: To enter dynamic fields in the email, use the [] button:



- Any time you start a loop (subprocess), you must set when to exit the loop using a condition.
- The Post Loop is complete when the Workflow_Status field = "End Post Loop"





- Add Conditions to transitions when needed.
 - If Supporting Document exists
 - Use a Label on the transition to make your workflow easier to read

Transition	
Condition	[/my:myFields/my:SharePointColumns_BAPI_ACC_DOCUMENT_POST_DFL/my:Supporting_Documents] != ""
Label	Supporting Documents
Name	

- Condition

Edit Condition

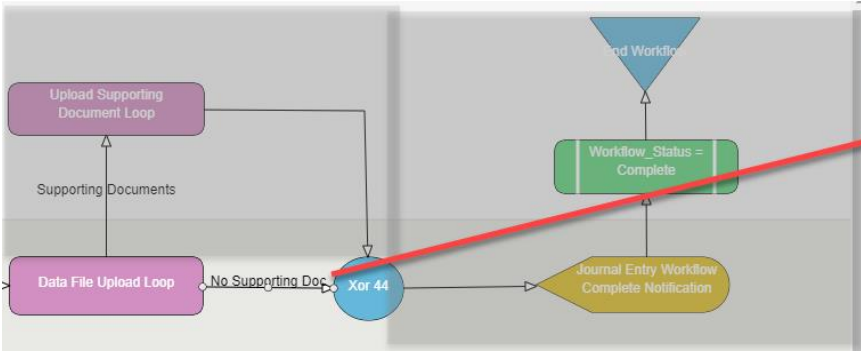
INSERT:

Field Value Operator... Function... Node...

[/my:myFields/my:SharePointColumns_BAPI_ACC_DOCUMENT_POST_DFL/my:Supporting_Documents] != ""

OK Cancel

- Otherwise (no Supporting Document) condition

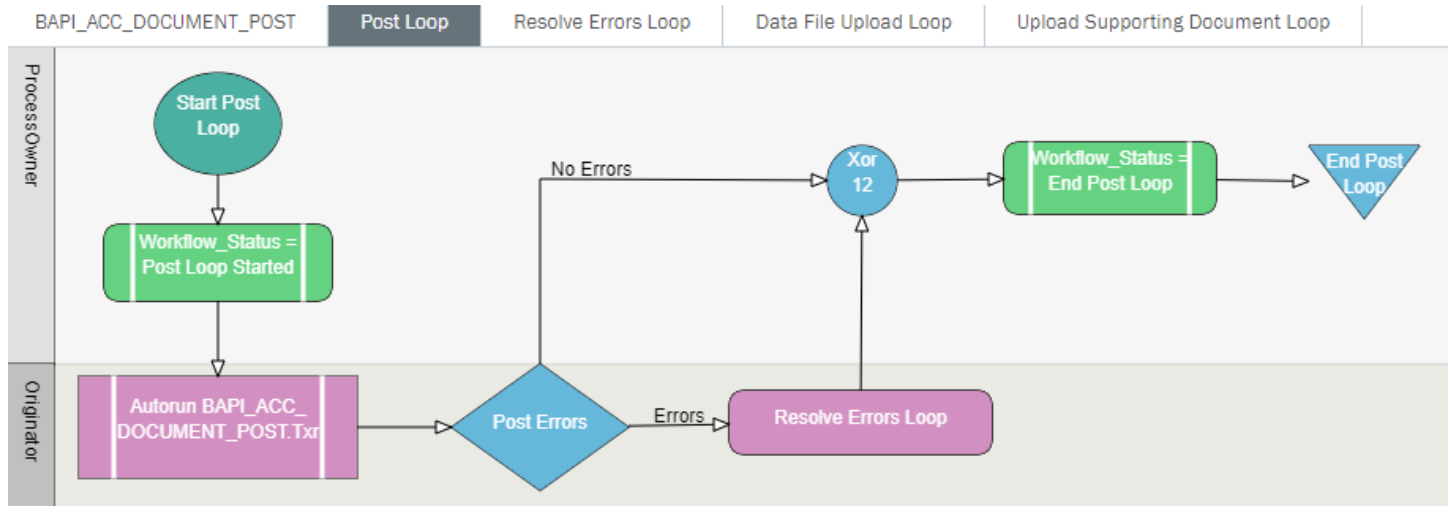


Search Properties

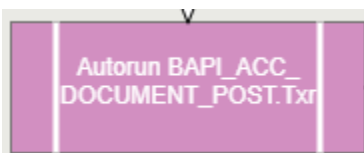
Transition	
Condition	Otherwise
Label	No Supporting Doc
Name	



Post Loop Subprocess - Uploads the journal entry data into SAP



Node Tips:



- The Winshuttle Autorun Plug-in will automatically run the Winshuttle script identified in the p-lug-in parameters.
- Set **Continue Without Waiting** to False to prevent the workflow from moving forward before the transaction script is complete.

Action	
Continue Without Waiting	False
Plug-In Name	Winshuttle Autorun
Plug-In Parameters	(Collection)

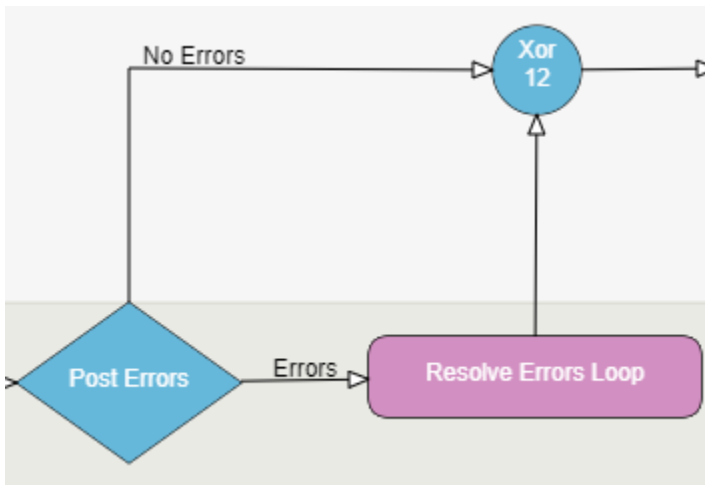
- If there are multiple scripts with the template, enter the name of the Winshuttle script you want the workflow to run in the **script name parameter**. (The **Runner node parameter** will be described in the next update.)

Plugin Arguments Editor

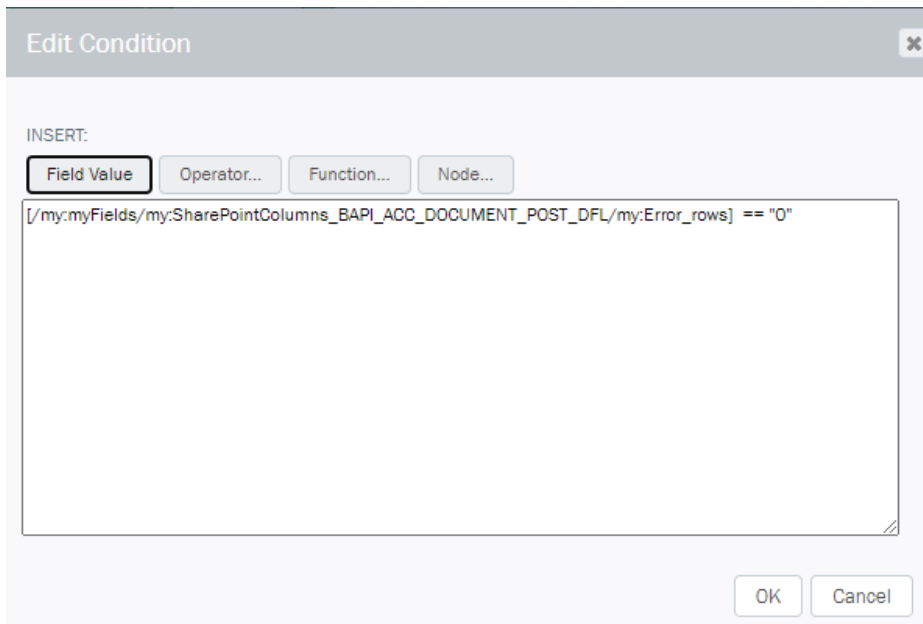
Validate

Parameter	Value
validate	<input type="radio"/> Yes <input checked="" type="radio"/> No
RunOnlyErrors	<input type="radio"/> Yes <input checked="" type="radio"/> No
RunOnlyUnprocessed	<input type="radio"/> Yes <input checked="" type="radio"/> No
StopRunOnError	<input type="radio"/> Yes <input checked="" type="radio"/> No
retries	<input type="text"/> <input type="button" value="..."/> <input type="button" value="🗑"/>
retryminutes	<input type="text"/> <input type="button" value="..."/> <input type="button" value="🗑"/>
runnernode	<input type="text"/> <input type="button" value="..."/> <input type="button" value="🗑"/>
scriptname	BAPI_ACC_DOCUMENT_POST.Txr <input type="button" value="..."/> <input type="button" value="🗑"/>

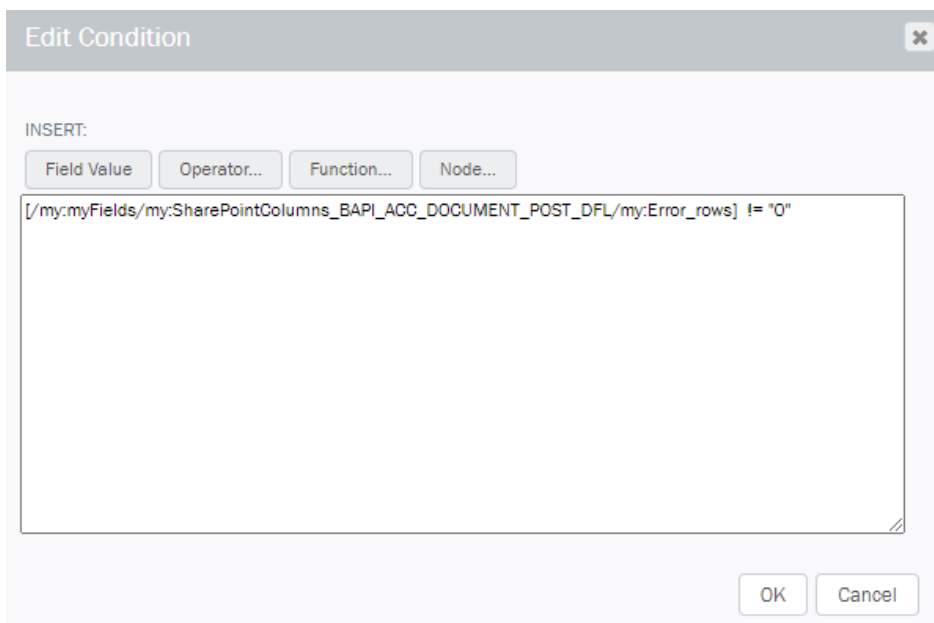
OK Cancel



- Use a decision node when there are conditions that will determine the next step in the workflow.
- No Errors condition – The script reported no errors in the Error_rows field upon completion:



- Errors condition – The script reported errors in the Error_rows field upon completion:



- Otherwise could also be used instead of one of these conditions.

-
- End Loop Condition - Always make sure you set the end loop condition to let the workflow know when the Loop is complete.



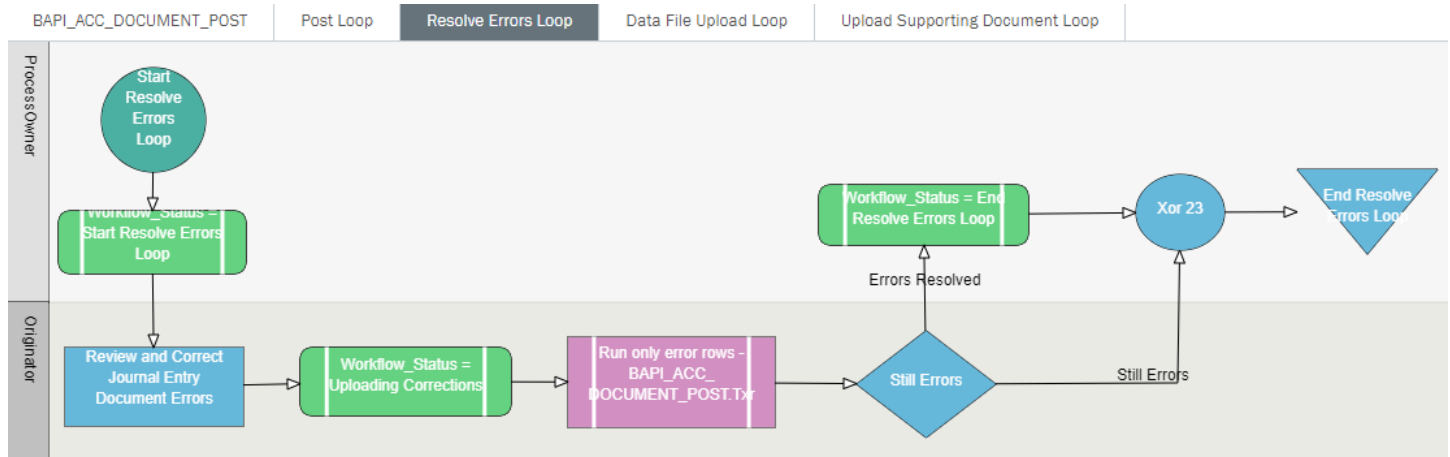
Value Pair Collection Editor ✕

Add

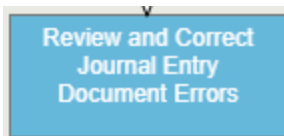
Form Field : Workflow_Status	"End Post Loop"	✍	🗑
------------------------------	-----------------	---	---

OK Cancel

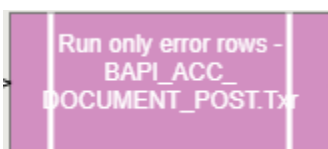
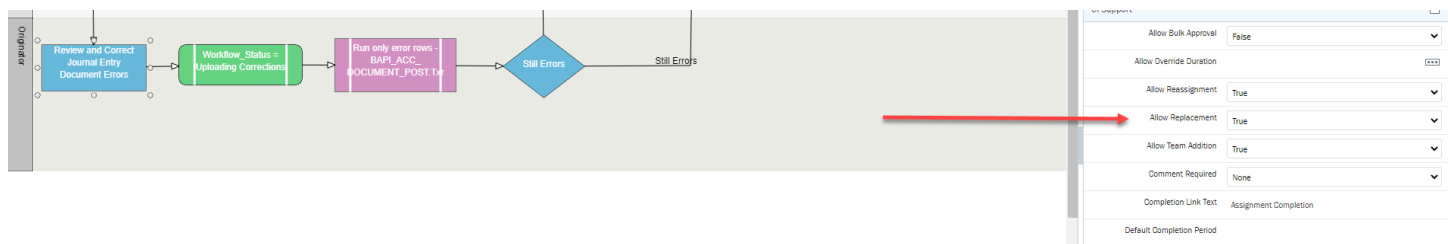
Resolve Errors Loop Subprocess – Assigns an activity task to the originator to resolve the errors



Node Tips:



- When assigning a task that requires the Originator to modify the data in the Excel data file, you must set Allow Replacement = True to change the Excel file Read Only property to false.
 Note: For this task we can set Require Review property to True. This will force user to open the file at least once before completing the task. This will also restrict user to complete the task from browser as user will have to open the data file at least once.
 Additional Information: When user will open the file, it will be checked out to that user. In case of team assignment, other users will not be able to open the file until this user saves and closes the file.





- If a Winshuttle script has already processed successful rows and you want it to only run on error rows, set the **RunOnlyErrors** parameter to Yes. This feature is available only in Studio 12.1 and later versions. In Studio 12.1, we added the RunOnlyErrors, RunOnlyUnprocessed, and StopRunOnError options.

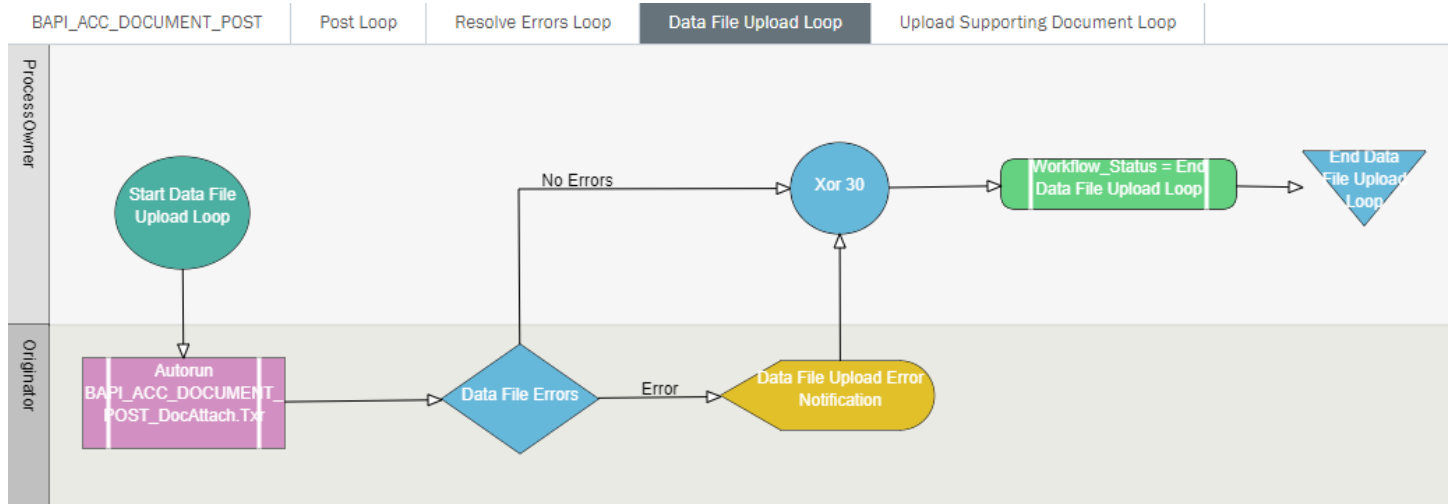
Plugin Arguments Editor ✕

Validate

Parameter	Value
? validate	<input type="radio"/> Yes <input checked="" type="radio"/> No 🗑️
? RunOnlyErrors	<input checked="" type="radio"/> Yes <input type="radio"/> No 🗑️
? RunOnlyUnprocessed	<input type="radio"/> Yes <input checked="" type="radio"/> No 🗑️
? StopRunOnError	<input type="radio"/> Yes <input checked="" type="radio"/> No 🗑️
? retries	<input style="width: 100%;" type="text"/> ⋮ 🗑️
? retryminutes	<input style="width: 100%;" type="text"/> ⋮ 🗑️
? runnernode	<input style="width: 100%;" type="text"/> ⋮ 🗑️
? scriptname	<input style="width: 100%;" type="text" value="BAPI_ACC_DOCUMENT_POST.Txr"/> ⋮ 🗑️

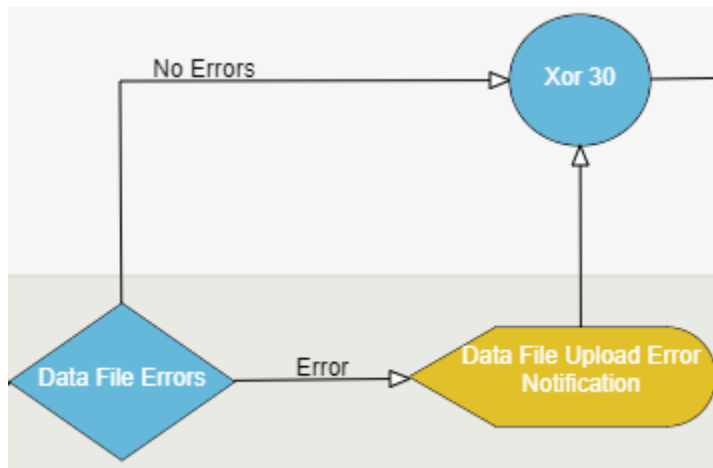
OK
Cancel

Data File Upload Loop Subprocess – Uploads the Excel data file into SAP



Node Tips:

- If there was an error uploading an attachment, you can notify the Originator to manually upload the attachment.



Validate and Deploy a Composer Solution

Validating a solution checks the solution for any potential issues that would prevent the workflow or the form from working properly.

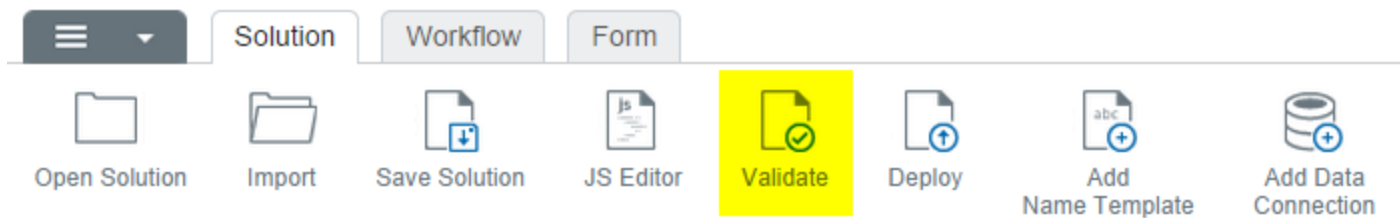
For example, it will verify Workflow validity by making sure there is a Start Node and an End Node, and that all **Nodes** have transitions.

Notes:

Validation is performed automatically when you deploy a solution.

Validation errors can be returned on fields that are collapsed; to avoid the error, hide the field.

Solution names are case sensitive. If you have used the same name for two solutions, varying only the initial letter's case (for example, "Test" and "test"), you will receive the following error message when you deploy the solution: "An unexpected error occurred on the server. Have your administrator check the server logs."



To validate a solution:

1. Open the solution.
2. On the Composer ribbon, on the Solution tab, click **Validate**.
3. Correct errors as needed.
4. Click **OK**.

Solution Validation Checks

Workflow validation checks

- Checks for a Start node (main graph and each loop graph)
- Checks that graphs have only one Start node
- Checks for at least one End node (main graph and each loop graph)
- Checks for missing transitions.
- Checks that Loops have until or while conditions
- Validates Swim Lanes (checks that Group/User specified)

- Validates variables used in Workflow expressions
- Checks for ambiguous **Otherwise** transitions. (e.g. multiple Otherwise transitions coming from the same node)
- Checks for missing plug-ins
- Checks for missing Swim Lanes (e.g. assignable tasks outside of a Swim Lane)
- Checks for valid connection Strings, SharePoint sites, SharePoint lists, users and groups, and plug-in arguments.

Form validation checks

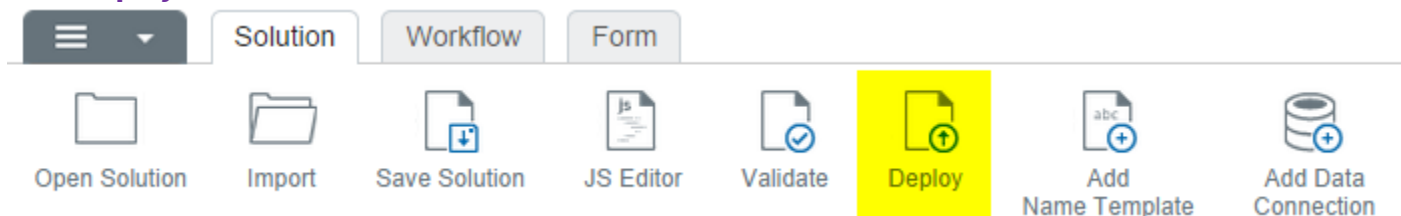
- None: No validation checks are performed on the form

Deployment Profile/Variants validation checks

- Checks for SharePoint site
- Checks for the Winshuttle Central/User Governance site (if there are scripts)
- Checks for SAP Integration Server (if there are scripts)

Deploying a solution

1. Click the **Solution** tab.
2. Click **Deploy**.



- Check **Force update of name templates** to re-publish the **name template** and reset initial values.
- Check **Publish Form in Background** if the solution is large. The form will be published in a different thread asynchronously, and you will receive an email notification after the form is published successfully.
- Check **Republish scripts** to re-create the Web services on the CENTRAL server for any mapped Transaction/Query script.

Deploy Solution ✕

FORCE UPDATE OF NAME TEMPLATES

PUBLISH FORM IN BACKGROUND

REPUBLISH SCRIPTS (IF THIS IS UNCHECKED, MAKE SURE ALL THE WEB SERVICES ARE ALREADY PUBLISHED ON WINSHUTTLE SERVER)

DEPLOYMENT PROFILE

Default ▼

SHAREPOINT SITE: HTTP://CHA-EN-VSTPT4/

OK Cancel

Deploy Solution ✕

FORCE UPDATE OF NAME TEMPLATES

REPUBLISH SCRIPTS (IF THIS IS UNCHECKED, MAKE SURE ALL THE WEB SERVICES ARE ALREADY PUBLISHED ON WINSHUTTLE SERVER)

DEPLOYMENT PROFILE

Default ▼

SHAREPOINT SITE: HTTP://SP2013/SITES/BF

OK Cancel

3. Select the **Deployment Profile** from the drop down.
4. Click **OK**.

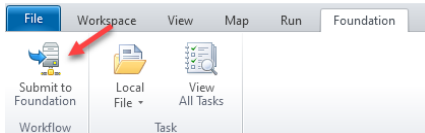
Where it all Comes Together

Once you have completed all of the pieces that make up an Excel Workflow (Excel Data Template, Winshuttle Scripts, and Workflow), you need to link all of the pieces together.

Steps:

Submit Excel Data Template and Studio Scripts to Foundation

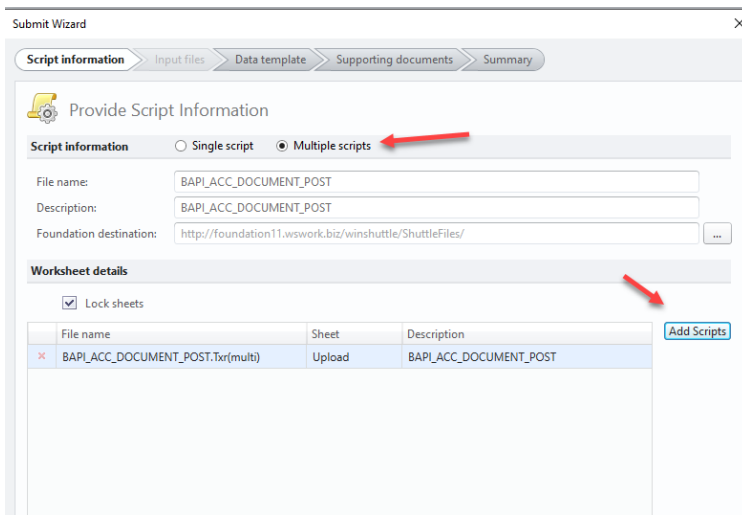
1. In Studio, open a script that will be used with the workflow.
2. Click Submit to Foundation on the Foundation tab.



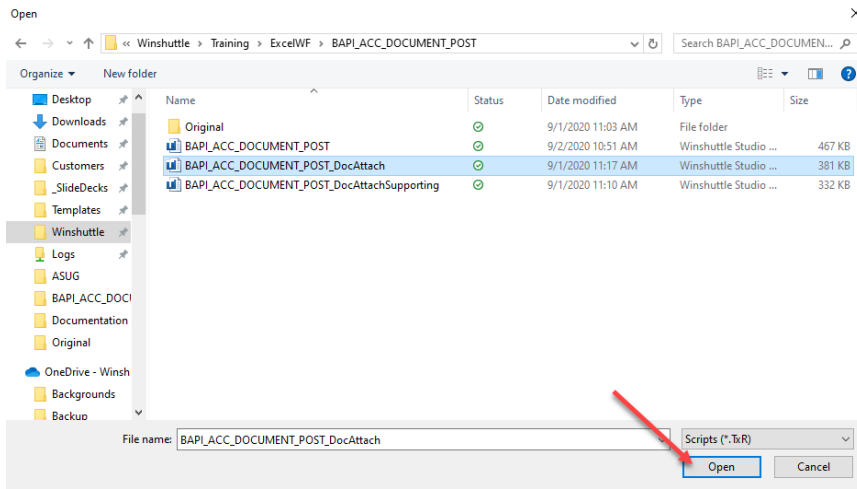
3. In the Submit Wizard

a. Script Information

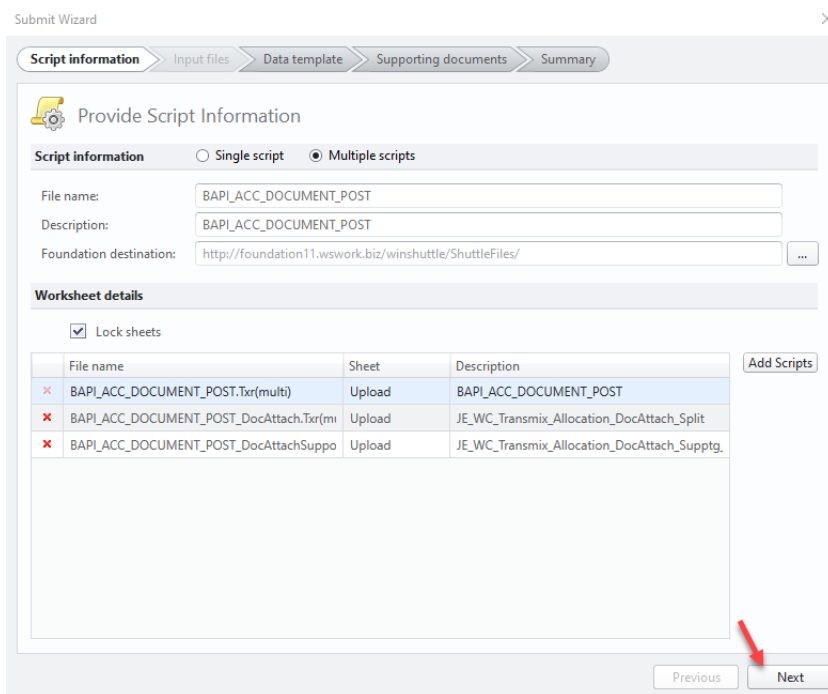
- i. Click Single script if the workflow will only use one script or click Multiple scripts if the workflow will require more than one.
- ii. If there are Multiple scripts, click Add Scripts



- iii. Select an additional script and then click Open, repeat as necessary



iv. Once all additional scripts have been added, click **Next**.



b. Data Template

The data file mapped in the original script will be listed by default in the Data template screen. Only change the data file if required. Then click **Next**.

Submit Wizard

Script information > Input files > **Data template** > Supporting documents > Summary

Provide the data file that will be submitted with this script

File name

Data File*: BAPI_ACC_DOCUMENT_POST [X] ...

Description: []

Foundation destination: http://foundation11.wswork.biz/winshuttle/Data Templates/ []

Previous Next

c. Supporting documents

If you have any Supporting documents, for example work instructions or training guides, you can add them in the **Supporting documents** screen.

i. When finished, click **Next**.

Submit Wizard

Script information > Input files > Data template > **Supporting documents** > Summary

Provide Supporting Document Information

Supporting documents attached:

File name	Type	Foundation destination
-----------	------	------------------------

Add Existing

Add New

Previous Next

d. Summary – Review what you just submitted.

i. Click Submit

The screenshot shows the 'Submit Wizard' dialog box with the 'Summary' step selected. The wizard has four steps: 'Script information', 'Input files', 'Data template', and 'Supporting documents', with 'Summary' being the final step. The main content area is titled 'Summary: Submit to Foundation' and contains four sections:

- Script info:** File name: BAPI_ACC_DOCUMENT_POST.Txx(multi) Total Scripts attached : 3; Foundation destination: http://foundation11.wswork.biz/winshuttle/ShuttleFiles/
- Input files info:** Input Files: Not applicable
- Data template:** File path: http://foundation11.wswork.biz/winshuttle/Data Templates/BAPI_ACC_DOCUMENT_POST.xlsm
- Supporting documents:** Supporting documents: Not applicable

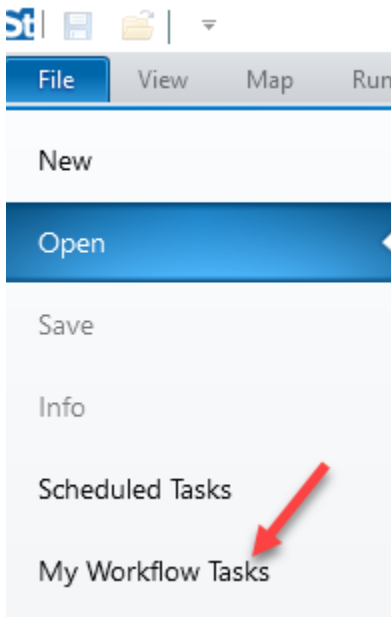
At the bottom of the dialog, there are two buttons: 'Previous' and 'Submit'.

ii. A message will display letting you know your files have been submitted. Click **OK**.

The screenshot shows a confirmation message dialog box titled 'Submit to Foundation'. It contains an information icon (i) and the text: 'Your script and data file have been sent for processing.' At the bottom of the dialog, there is an 'OK' button.

4. Approve the scripts and assign the workflow

a. In Studio, open **My Workflow Tasks**.

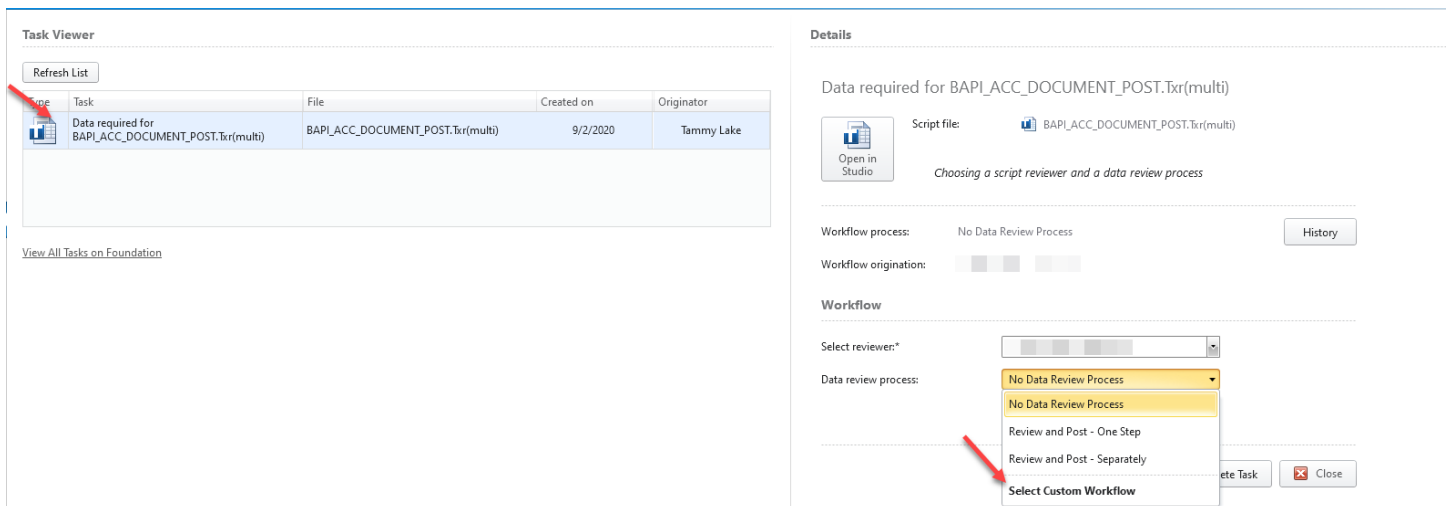


b. Left screen - Click the Task for the scripts you just submitted in the Task Viewer

c. Right screen –

i. Select a Reviewer from the Select reviewer drop down list

ii. **Important:** Select **Custom Workflow** from the Data review process drop down. This is how you link everything together.



d. Select the workflow you want to link the files to and click **OK**.



Custom Workflow List

bapi

Workflow name	Data file library
BAPI_ACC_DOCUMENT_POST_NoApproval	http://foundation11.wswork.biz/winshuttle/Lists/BAPI_ACC_DOCUMENT_POST
WC_Transmix_Allocation_BAPI	http://foundation11.wswork.biz/winshuttle/WC_Transmix_Allocation_BAPI_DFL
BAPI_ACC_DOCUMENT_POST	http://foundation11.wswork.biz/winshuttle/BAPI_ACC_DOCUMENT_POST_DFL

OK Cancel

- e. Choose the SAP system you want the workflow to write to and click **Complete Task**. Ensure that the Runner user (depends on the swim lane in which the AutoRun plug-in is placed and also on the Runner Node property of the AutoRun plug-in), in whose context the SAP posting will be done, has SAP credentials saved. Otherwise, the AutoRun plug-in will fail. In addition, ensure that they have a valid license and policy.

Details

Data required for BAPI_ACC_DOCUMENT_POST.Txr(multi)

Script file: BAPI_ACC_DOCUMENT_POST.Txr(multi)

Open in Studio

Choosing a script reviewer and a data review process

Workflow process: BAPI_ACC_DOCUMENT_POST History

Workflow origination: Tammy Lake - 9/2/2020

Workflow

Select reviewer*:

Data review process: BAPI_ACC_DOCUMENT_POST


SAP server:

Complete Task Close

- f. The reviewer you selected will be assigned the approval task. Open the task in Studio and then click Approve to complete the task.

Task Viewer



Refresh List

Type	Task	File	Created on	Originator
	Review required for BAPI_ACC_DOCUMENT_POST.Tx(multi)	BAPI_ACC_DOCUMENT_POST.Tx(multi)	9/2/2020	Timmy Lake

[View All Tasks on Foundation](#)

Details

Review required for BAPI_ACC_DOCUMENT_POST.Tx(multi)

 Script file:  BAPI_ACC_DOCUMENT_POST.Tx(multi)




[Open in Studio](#)

This is a simple script review task with a single approver

Workflow process: BAPI_ACC_DOCUMENT_POST [History](#)

Workflow origination: Timmy Lake - 9/2/2020

Workflow

Actions:   

Comments:

[Complete Task](#) [Close](#)

g. You have now completed your Excel workflow!