UiPath Reboot Work.™ | Complete Exercise Walkthrough

Generate Yearly Report

Contents

1. Overview

For this exercise, we will use the producer-consumer model. This means we will build two projects:

- A Dispatcher – using the REF without Queue Items design.
- A Performer – using the REF with Queue Items design.

We encourage you to use the two checklists and to identify reusable components.

2. The Dispatcher process

2.1. Prerequisites

The automation will need to identify if a page number is available on the Work Items page by using the transaction number. If a section is available, it will scrape it, filter it for WI4 type work items, and add a queue item for each resulting Datatable row. The TransactionItem will be of type Int32.

2.2. Configuration in UiPath Studio

2.2.1. Create a new project

1. Create new project using the Robotic Enterprise Framework template.
2. Set a proper name for the project.
3. Provide a proper description.

2.2.2. Whiteboard your workflows

Module Name	Description	Pre-condition	Post-action	Arguments
System1_Login	Login into Acme with the desired account	N/A	Browser opened to the required URL Logged in with the credentials provided	in_System1URL - String in_System1Credential – String
System1_Close	Close browser	Browser open to ACME page	Browser is closed	N/A
System1_NavigateTo_WI	Navigate to the WI URL	Be logged into the ACME website	Navigated successfully to the desired URL	in_URL - String
System1_ScrapeDataTable	Extract data from the work items datatable	Data exists and be on the work items page	All data table data is extracted	out_DataTable - Datatable

Module Name	Description	Pre-condition	Post-action	Arguments
System1_FilterWIDatatable	Filters the input Datatable for rows with the desired Work Item	Datatable exists	The filtered datatable is passed to the PopulateQueue workflow.	in_Type – String in_Datatable – DataTable out_FilteredDatatable – DataTable in_Status - String

2.2.3. Develop your workflows

1. Create a new folder in the Project panel. Name it System1.

2.2.3.1. System1_Login.xaml

1. Inside the System1 folder, create a new Sequence type workflow called System1Login.
2. Provide a description to the workflow.
3. Open the Arguments panel and create two in arguments of type string:
    a. In_System1URL
    b. In_System1Credential
4. Add a Log message activity to mark the start of the workflow execution.
5. Add a Get Credential activity. Configure the following properties:
    a. Asset Name – Provide the in_System1Credential argument.
    b. Password – Create a new variable called Password of type SecureString.
    c. Username – Create a new variable called Username of type String.
6. Add an Open Browser activity. Configure the following properties:
    a. BrowserType – Set it to the desired browser.
    b. URL – Provide the in_System1URL argument.
7. Inside the Do Container of the Open Browser activity add:
    a. A Maximize Window activity.
    b. A Type Into activity where you indicate the email field in the browser and provide the Username variable for the Text property.
    c. A Type Secure Text activity where you indicate the password field in the browser and provide the Password variable for the SecureText property.
    d. A Click activity where you indicate the Login button in the browser as target.
8. Log into ACME in your browser.
9. After the Open Browser activity, add an Element Exists activity.
    a. Indicate the Dashboard header in the browser as a target.
    b. Create a new Boolean variable for the Exists property called LogInSuccessful.
10. Add an If activity.
    a. Set the condition to Not LogInSuccessful.
11. Go to ACME, return to the login page, try to log in with the wrong credentials. Leave the error pop-up window open.
12. Return to Studio. Inside the Then block of the If activity:
    a. Add a Click activity and indicate the OK button of the error pop-up window.
    b. Add a Throw activity with the Exception property set to: new Exception("Incorrect Credentials supplied to System1")

   c. Note that later during development, we will invoke a workflow here to send an email to the process owner.
13. After the If activity, add a Log Message to mark the end of the workflow.

2.2.3.2. System1_Close.xaml

1. Make sure you are logged into ACME System1 in your browser.
2. In Studio, add a Log Message activity to mark the start of the workflow.
3. Add an Attach Browser activity and indicate the browser logged into ACME.
4. Inside the Do container of the Attach Browser activity:
   a. Add a Click activity and indicate the Log Out button.
   b. Add a Close tab activity.
5. Add a Log Message activity to mark the end of the workflow.

2.2.3.3. System1_ NavigateTo_WI.xaml

1. Open the Arguments tab and create a new in argument of String type called in_URL.
2. Add a Log Message activity to mark the start of the workflow.
3. Make sure the ACME System 1 Dashboard page is open in your browser.
4. In Studio, add an Attach Browser activity and indicate the ACME System 1 page.
5. Inside the Do Container of the Attach Browser activity, add a Navigate To activity and provide the in_URL argument for the URL property.
6. Add a Log Message activity to mark the end of the workflow.

2.2.3.4. System1_ ScrapeDataTable.xaml

1. Open the Arguments panel and create a new out argument of DataTable type called out_DataTable.
2. In your browser, navigate to the Work Items page on ACME System1.
3. In Studio, use the Data Scraping wizard to extract the work items data. Configure its properties as follows:
   a. For the Output DataTable property, enter out_DataTable.
   b. Set MaxNumberOfResults to 0
4. Add a Log Message to note the number of rows extracted.
   a. For the Message property enter: "Found " + out_Datatable.Rows.Count.ToString + " elements in the datatable"
5. Add a Log Message activity to mark the end of the workflow.

2.2.3.5. System1_ FilterWIDatatable.xaml

1. Open the Arguments panel and create four new arguments:
    a. in_Type of type String
    b. in_Datatable of type DataTable
    c. out_FilteredDatatable of type DataTable
    d. in_Status of type String
2. Add a Log message activity to mark the start of the workflow execution.
3. Add a Filter Data Table activity. Configure it as follows:
    a. Open the Filter Wizard, set the Filtering Mode to Keep and set:
        i. “Type” = in_Type
        ii. “Status” = in_Status
    b. Close the Filter Wizard.
    c. Set the Input DataTable property to in_Datatable.
    d. Set the Output DataTable property to out_FilteredDatatable.
4. Add a Log Message activity to mark the end of the workflow.

2.2.4. Edit the Configuration file

1. Provide the following values in the Settings sheet:

Name	Value
OrchestratorQueueName	Acme4Queue
logF_BusinessProcessName	ACME4_Dispatcher
System1_URL	https://acme-test.uipath.com
System1_Credential	System1_Credential
System1_WorkItemsURL	https://acme-test.uipath.com/work-items

2. In the Constants sheet, set the MaxRetryNumber to 2.
3. In Orchestrator, create a Credential type Asset for ACME System 1 with the name System1_Credential.

2.3. Change the TransactionItem data type

2.3.1. In Main.xaml

1. Locate the TransactionItem variable and change its type to Int32.

2.3.2. In the Get Transaction Data State

1. Configure the End Process (Stop process requested) to:
    a. To: TransactionItem
    b. Value: 0
2. Select the Invoke GetTransactionData activity. Open the Arguments list and change the data type for the out_TransactionItem argument from QueueItem to Int32.
3. Expand the Exception section in the Try GetTransactionData section and locate the
4. End Process (Could not get new transaction) activity. Configure it to:
    a. To: TransactionItem
    b. Value: 0

2.3.2. In the Process State

1. Locate the Invoke Process workflow activity and open the Arguments list; change the data type for the in_TransactionItem argument from QueueItem to Int32.
2. Locate the Invoke SetTransactionStatus workflow activity and open the Arguments list; locate the entry for the in_TransactionItem argument and change the value (not the data type!) from TransactionItem to Nothing.

2.4. Applications Used: open/close/kill

2.4.1. Edit the InitiAllApplications.xaml workflow

1. Add a Log Message activity to mark the start of the workflow.
2. Invoke the System1\System1_Login.xaml workflow.
    a. Click Import Arguments and make the following changes:

Name	Direction	Type	Value
in_System1URL	In	String	in_Config("System1_URL").ToString
in_System1Credential	In	String	in_Config("System1_Credential").ToString

3. Invoke the System1\ System1_NavigateTo_WI.xaml workflow.
    a. Click Import Arguments and make the following changes:

Name	Direction	Type	Value
in_URL	In	String	in_Config("System1_WorkItemsURL").ToString

4. Add a Log Message activity to mark the end of the workflow.

2.4.2. Edit the Framework/CloseAllApplication.xaml workflow

1. Add a Log Message activity to mark the start of the workflow.
2. Invoke the System1\System1_Close.xaml workflow.
3. Add a Log Message activity to mark the end of the workflow.

2.4.3. Edit the Framework/KillAllProcesses.xaml workflow

1. Add a Log Message activity to mark the start of the workflow.
2. Add a Kill Process activity.
    a. For the ProcessName property, enter the process name for your browser (for example, “msedge” for Edge)
3. Add a Log Message activity to mark the end of the workflow.

2.5. Business Process: Transaction Data and Process

2.5.1. Edit GetTransactionData.xaml workflow

1. Set the type of the out_TransactionItem argument to Int32.
2. Delete the Retry Get transaction item activity.
3. At the top of the Get Transaction Data sequence, add an Element Exists activity.
    a. Indicate the second Page Button on the Work Items page.
    b. Create a dynamic selector with the xml code:
        <html title='ACME System 1 - Work Items' />
        <webctrl aaname='{{in_TransactionNumber}}' tag='A' parentclass='page-numbers' />
    c. Set the Output Exists property to a new variable of Boolean type called PageExists.
4. Add an If activity and set the Condition to Page Exists.
    a. In the Then block, add an Assign Activity:
        i. To: out_TransactionItem.
        ii. Value: in_TransactionNumber.
    b. In the Else branch, add another Assign activity:
        i. To: out_TransactionItem.
        ii. Value: 0.
5. Set the Condition of the If a new transaction item is retrieved, get additional information about it activity to out_TransactionItem > 0.

2.5.2. Edit Process.xaml workflow

1. Change the data type for the in_TransactionItem argument from QueueItem to Int32.
2. Create three new variables:

   a. CurrentPageAlreadyOpen of type Boolean
   b. WorkItems of type DataTable
   c. FilteredWorkItems of type DataTable
3. Add an Element Exists activity:
   a. Indicate the first page button in the Work Items page.
   b. Set the Selector to:
      <html title='ACME System 1 - Work Items' />
      <webctrl aaname='{{in_TransactionItem}}' tag='A' class='page-numbers current' />
   c. Set the Output Exists property to CurrentPageAlreadyOpen.
4. Add an If Activity.
   a. Set the Condition to CurrentPageAlreadyOpen.
   b. Inside the Then branch:
      i. Add a Log Message activity with the Message: "Already on the correct page for transaction " + in_TransactionItem.ToString
   c. Inside the Else branch:
      i. Add a Click activity and set the Selector to:
      <html title='ACME System 1 - Work Items' />
      <webctrl aaname='{{in_TransactionItem}}' tag='A' parentclass='page-numbers' />
5. Invoke the System1\ System1_ ScrapeDataTable.xaml
   a. Click Import Arguments and make the following changes:

Name	Direction	Type	Value
out_DataTable	Out	DataTable	WorkItems

6. Invoke System1_FilterWIDatatable.xaml
   a. Click Import Arguments and make the following changes:

Name	Direction	Type	Value
in_Type	In	String	“WI4”
In_Datatable	In	DataTable	WorkItems
out_FilteredDataTable	Out	DataTable	FilteredWorkItems
In_Status	In	String	“Open”

7. Add an If activity and set the condition to FilteredWorkItems.Rows.Count > 0
   a. Inside the Then branch of the If activity, add a Bulk Add Queue Items activity. Configure its properties to:
      i. DataTable: FilteredWorkItems
      ii. QueueName: in_Config("OrchestratorQueueName").ToString

3. The Performer process

3.1. Prerequisites

Make sure the queue indicated in the previous section is available in Orchestrator. Set the Retry value to 2.

3.2. Configuration in UiPath Studio

3.2.1. Create a new project

1. Create new project using the Robotic Enterprise Framework template.
2. Set a proper name for the project.
3. Provide a proper description.

3.2.2. Whiteboard your workflows

Module Name	Description	Pre-condition	Post-action	Arguments
System1_Login	same as in Dispatcher			
System1_Close	same as in Dispatcher			
System1_NavigateTo	same as System1_NavigateTo_WI in Dispatcher			
System1_GetClientDetails	Get the client information: tax id	Logged in and on the work item WIID page	Got all the client information : taxID	out_TaxID - String
System1_DownloadMonthlyReports	Enter tax ID, enter year and for each month download report if exists (treat exceptions)	Logged in and on the download monthly report page, report directory exists	Downloaded all available reports for a given TaxID and a given year into a given location	in_TaxID - String in_Year - String in_ReportDirPath - String
System1_MergeMonthlyReports	Merge the downloaded csv from a given TaxID and a year, located into a certain folder, into one excel file - yearly report	Monthly reports exist into a certain folder	Constructed the merged excel file and save it with name format "Yearly-Report-year-taxid.xlsx"	in_TaxID - String in_Year - String in_ReportDirPath - String out_YearlyReportPath - String

Module Name	Description	Pre-condition	Post-action	Arguments
System1_UploadYearlyReport	Upload a given yearly report to the upload yearle report page for a given tax id and a given year	Yearly report excel file exists into the specified folder and we are on the upload yearly report page Post Report uploaded successfully and retrieved confirmation id	Report uploaded successfully and retrieved confirmation id	in_TaxID - String in_Year - String in_YearlyReportPath - String out_Confimation - String
System1_UpdateWorkItems	Add hash into comment section and change status to "Completed"	Be on the update work item page	Hash added to the comment section and status changed to "Completed", popup closed	in_Comment - String in_Status - String

3.2.3. Develop your workflows

1. Create a new folder in the Project panel. Name it System1.

3.2.3.1. System1_Login.xaml

Note: We recommend saving the System1_Login.xaml workflow from the Dispatcher project as a component and reusing it here. Below are the steps to develop the workflow.

1. Inside the System1 folder, create a new Sequence type workflow called System1Login.
2. Provide a description to the workflow.
3. Open the Arguments panel and create two in arguments of type string:
    a. in_System1URL
    b. in_System1Credential
4. Add a Log message activity to mark the start of the workflow execution.
5. Add a Get Credential activity. Configure the following properties:
    a. Asset Name – Provide the in_System1Credential argument.
    b. Password – Create a new variable called Password of type SecureString.
    c. Username – Create a new variable called Username of type String.

6. Add an Open Browser activity. Configure the following properties:
    a. BrowserType – Set it to the desired browser.
    b. URL – Provide the in_System1URL argument.
7. Inside the Do Container of the Open Browser activity add:
    a. A Type Into activity where you indicate the email field in the browser and provide the Username variable for the Text property.
    b. A Type Secure Text activity where you indicate the password field in the browser and provide the Password variable for the SecureText property.
    c. A Click activity where you indicate the Login button in the browser as target.
8. Log into ACME in your browser.
9. After the Open Browser activity, add a Pick activity.
10. Inside the first branch of the Pick activity, add a PickBranch activity.
    a. Inside the Trigger block of the PickBranch activity add an Element Exists activity and indicate the Dashboard header in the browser as a target.
    b. Inside the Action block of the PickBranch activity add a Log Message activity to note the successful login.
11. Log out of ACME in your browser and attempt to log in using the wrong credentials. Leave the resulting message box displayed on screen.
12. Inside the second branch of the Pick activity, add a PickBranch activity.
    a. Inside the Trigger block of the PickBranch activity, add an Element Exists activity and indicate the OK button of the error message box in your browser.
    b. Inside the Action block of the PickBranch activity:
        i. Add a Log Message activity to note the unsuccessful login attempt.
        ii. Add a Click activity and indicate the OK button for the error message.
        iii. Add a Throw activity and inside the Exception property field enter: New BusinessRuleException("Could not login successfully")
13. After the Pick activity, add a Log Message to mark the end of the workflow.

3.2.3.2. System1_Close.xaml

Note: We recommend saving the System1_Close.xaml workflow from the Dispatcher project as a component and reusing it here. Below are the steps to develop the workflow.

1. Make sure you are logged into ACME System1 in your browser.
2. In Studio, add a Log Message activity to mark the start of the workflow.
3. Add an Attach Browser activity and indicate the browser logged into ACME.
4. Inside the Do container of the Attach Browser activity:
    a. Add a Click activity and indicate the Log Out button
    b. Add a Close tab activity
5. Add a Log Message activity to mark the end of the workflow.

3.2.3.3. System1_ NavigateTo.xaml

Note: We recommend saving the System1_NavigateTo.xaml workflow from the Dispatcher project as a component and reusing it here. Below are the steps to develop the workflow.

1. Open the Arguments tab and create a new in argument of String type called in_URL.
2. Add a Log Message activity to mark the start of the workflow.
3. Make sure the ACME System 1 Dashboard page is open in your browser.
4. In Studio, add an Attach Browser activity and indicate the ACME System 1 page.
5. Inside the Do Container of the Attach Browser activity, add a Navigate To activity and provide the in_URL argument for the URL property.
6. Add a Log Message activity to mark the end of the workflow.

3.2.3.4. System1_DownloadMonthlyReports.xaml

1. Open the Arguments panel and create three in arguments of type string:
    a. in_TaxID
    b. in_Year
    c. in_ReportDirPath
2. Add a Log message activity to mark the start of the workflow execution.
3. In your browser, navigate to the Reports – Download Monthly Report page in ACME System 1.
4. Include all following activities inside the Do container of the Attach Browser activity.
5. Add a Type Into activity.
    a. Indicate the Vendor TaxId field.
    b. Provide the in_TaxID for the Text property.
6. Add a Click activity.
    a. Indicate a the first year in the Year drop down.
    b. Edit the Selector to provide the following string: <webctrl aaname='{{in_Year}}' parentid='searchForm' tag='SPAN' />
7. Add an Assign activity.
    a. In the To field, create a new variable of type String[] called Months.
    b. In the Value field, add the expression: DateTimeFormatInfo.CurrentInfo.MonthNames
8. Add a For Each activity.
    a. Name the iterator “month”.
    b. Provide Months in the Values property.
    c. Change the TypeArgument to String.
9. Inside the Body of the For Each activity add an If activity.
    a. Name it If not an empty month
    b. Provide the condition: not String.IsNullOrEmpty(month)
10. Inside the Then Block of the If activity add a Click Activity.
    a. Indicate January in the Month drop down menu.
    b. For the selector, provide: "<webctrl aaname='" + month +"' parentid='searchForm' tag='A' />"
11. Add a Click activity and indicate the Download Report button.
12. In your browser, trigger a Report Download in ACME.
13. In Studio, depending on your browser and settings, handle the potential cases encountered during download using If and Element Exists activities:
    a. Incomplete information provided
    b. The Report does not exist
    c. Report Found

14. Implement the logic to store the downloaded report to in_ReportDirPath.

3.2.3.5. System1_MergeMonthlyReports.xaml

1. Open the Arguments panel and create four arguments of type string:
    a. in_TaxID
    b. in_Year
    c. in_RportDir
    d. out_YearlyReportPath
2. Add a Log message activity to mark the start of the workflow execution.
3. Add an Assign activity:
    a. To: Create a new variable of type String[] called ReportsByTaxID and add it to the field
    b. Value: Directory.GetFiles(in_ReportDir, "Report-"+in_TaxID+"*.csv")
4. Add a For Each activity:
    a. Set the iterator name to “file”
    b. Set the Values property to RportsByTaxID
    c. Set the TypeArgument to String
5. Inside the body of the For Each activity:
    a. Add a Read CSV activity
        i. Set the FilePath to file
        ii. Set the Output to a new variable of DataTable type called DTMonthlyReport
        iii. Make sure the Has headers option is checked
    b. Add a Merge Data Table activity
        i. Set the Destination to a new variable of DataTable type called DTYearlyReports
        ii. Set the Source to DTMonthlyReport
    c. Add a Delete activity and provide file for the Path property
6. After the For Each activity, add an Assign activity:
    a. To: out_YearlyReportPath
    b. Value: Path.Combine(in_ReportDir, "Yearly-Report-"+ in_Year + "-" + in_TaxID + ".xlsx")
7. Add a Write Range activity
    a. For the DataTable property enter DTYearlyReports
    b. For the WorkbookPath enter out_YearlyReportPath
    c. Make sure the AddHeaders option is enabled
8. Add a Log Message activity to mark the end of the workflow.

3.2.3.6. System1_UploadYearlyReport.xaml

1. Open the Arguments panel and create four arguments of type string:
    a. in_TaxID
    b. in_Year
    c. in_YearlyReportPath
    d. out_Confirmation

2. Add a Log message activity to mark the start of the workflow execution.
3. In your browser, navigate to the Reports – Upload Yearly Report page in ACME
4. In Studio, Add an Attach Browser activity and indicate the Reports – Upload Yearly Report page in ACME.
5. Until further notice, all the following activities should be added inside the Do Container of the Attach Browser activity.
6. Add a Type Into activity and indicate the Vendor TaxID field.
    a. Provide the in_TaxID argument for the Text property
7. Add a Click activity and indicate the first year in the Year drop down list.
    a. For the Selector, provide: <webctrl aaname='{{in_Year}}' parentid='searchForm' tag='SPAN' />
8. Add a Click activity and indicate the Select Report File button.
9. Click the Select Report File button in your browser window.
10. In Studio, indicate the File name input field.
    a. For the Text property enter in_YearlyReportPath + "[k(enter)]"
11. Add a Click activity and indicate the Upload Report button.
12. In your browser, make a successful upload and leave the resulting pop-up window open.
13. In Studio, add an Element Exists activity and indicate the text box for the pop-up window.
    a. For the Selector property, enter: <wnd ctrlid='65535' title='Report was uploaded - confirmation id is *' />
    b. For the Exists property, create a new Boolean variable called UploadedSuccessfully
14. Add an If activity and name it If uploaded successfully.
    a. Add the condition UploadedSuccessfully.
15. Inside the then branch of the If activity:
    a. Add a Get Text activity and indicate the Success message in the pop-up window.
        i. For the Selector property, enter: <wnd ctrlid='65535' title='Report was uploaded - confirmation id is *' />
        ii. For the Value property, create a new String variable called MessageText
    b. Add an Assign activity.
        i. To: Add out_Confirmation
        ii. Value: Add MessageText.Split(" "c).Last.ToString.Trim
    c. Add a Click activity and indicate the OK button.
    d. Add an Info level Log Message activity.
        i. For the Message property, enter "Confirmation id is : " + out_Confirmation.
16. After the For Each activity, add a Log Message activity to mark the end of the workflow.

3.2.3.7. System1_UpdateWorkItems.xaml

1. Open the Arguments panel and create two in arguments of type string:
    a. in_Comment
    b. in_Status
2. Add a Log message activity to mark the start of the workflow execution.

3. In your browser, navigate to the Work Items page, open a WI4 work item and click Update Work Item. Leave the resulting window open.
4. In studio, add an Attach Browser activity and indicate the Update Work Item window.
5. Until further notice, all following activities should be added in the Do container of the Attach Browser activity.
6. Add a Type Into activity and indicate the Add Comments input field.
    a. For the Text property, provide in_Comment
7. Add a Click activity and indicate the first status in the New Status dropdown menu.
    a. For the Selector property, enter "<webctrl aaname='{{in_Status}}' tag='SPAN' />"
8. Add a Click activity and indicate the Update Work Item button.
9. In your browser, make a successful update. Leave the resulting pop-up window open.
10. In Studio, add an Element Exists activity and indicate the OK button in the pop-up window.
    a. For the Exists property, create a new Boolean type variable called UpdatedSuccessfully.
11. Add an If activity and provide the Condition UpdatedSuccessfully.
12. Inside the Then branch of the If activity:
    a. Add an Info level Log Message with the Message: "Work item updated successfully".
    b. Add a Click activity and indicate the OK button.
13. Inside the Else branch of the If activity:
    a. Add an Error level Log Message with the Message: "Could not update work item"
    b. Add a Throw activity with the Exception: New BusinessRuleException("Could not update successfully the work item")
14. After the Attach Browser activity, add a Log Message activity to mark the end of the workflow.

3.2.4. Edit the Configuration file

1. Provide the following values in the Settings sheet:

Name	Value
OrchestratorQueueName	Acme4Queue
logF_BusinessProcessName	ACME4_Performer
ReportDirectory	C:\Users\lavinia.nastase\Desktop\Reports
System1_Credential	System1_Credential
AcmeURL	https://acme-test.uipath.com
AcmeWorkItemsURL	https://acme-test.uipath.com/work-items
AcmeDownloadMonthlyReportsURL	https://acme-test.uipath.com/reports/download
AcmeUploadYearlyReportURL	https://acme-test.uipath.com/reports/upload
AcmeUpdateURL	https://acme-test.uipath.com/work-items/update
Status	Completed

2. In the Constants sheet, make sure the MaxRetryNumber to 0.

3. In Orchestrator, make sure the Credential type Asset for ACME System 1 with the name System1_Credential is available.
4. Save and close the configuration file

3.3. Applications Used: open/close/kill

3.3.1. Edit the InitiAllApplications.xaml workflow

1. Add a Log Message activity to mark the start of the workflow.
2. Invoke the System1\System1_Login.xaml workflow.
    a. Click Import Arguments and make the following changes:

Name	Direction	Type	Value
in_System1URL	In	String	in_Config("System1_URL").ToString
in_System1Credential	In	String	in_Config("System1_Credential").ToString

3. Add a Log Message activity to mark the end of the workflow.

3.3.2. Edit the Framework/CloseAllApplication.xaml workflow

1. Add a Log Message activity to mark the start of the workflow.
2. Invoke the System1\System1_Close.xaml workflow.
3. Add a Log Message activity to mark the end of the workflow.

3.3.3. Edit the Framework/KillAllProcesses.xaml workflow

1. Add a Log Message activity to mark the start of the workflow.
2. Add a Kill Process activity.
    a. For the ProcessName property, enter the process name for your browser (for example, “msedge” for Edge)
3. Add a Log Message activity to mark the end of the workflow.

3.4. Business Process: Transaction Data and Process

3.4.1. Edit the GetTransactionData.xaml workflow

No changes are needed in this workflow.

3.4.2. Edit the Process.xaml workflow

1. Add an Assign activity:
    a. To: Create a new variable of type String called WIID
    b. Value: Add in_TransactionItem
2. Add an Info level Log Message activity with the Message: “Processing “ + WIID.
3. Invoke the System1\System1_NavigateTo.xaml workflow.
    a. Click Import Arguments and make the following changes:

Name	Direction	Type	Value
in_URL	In	String	in_Config("AcmeWorkItemsURL").ToString + "/" + WIID

4. Create a new variable of type String called TaxID
5. Invoke the System1\System1_GetClientDetails.xaml workflow.
    a. Click Import arguments and make the following changes:

Name	Direction	Type	Value
out_TaxID	Out	String	TaxID

6. Invoke the System1\System1_NavigateTo.xaml workflow.
    a. Click Import Arguments and make the following changes:

Name	Direction	Type	Value
in_URL	In	String	in_Config("AcmeDownloadMonthlyReportsURL").ToString

7. Invoke the System1\ System1_DownloadMonthlyReports.xaml workflow.
    a. Click Import Arguments and make the following changes:

Name	Direction	Type	Value
in_TaxID	In	String	TaxID
in_Year	In	String	“2021”
in_ReportDirPath	In	String	in_Config("ReportDirectory").ToString

8. Create a new variable of string type called YearlyReportPath.
9. Invoke the System1\System1_MergeMonthlyReports.xaml workflow.
    a. Click Import Arguments and make the following changes:

Name	Direction	Type	Value
in_TaxID	In	String	TaxID
in_Year	In	String	“2021”
in_ReportDir	In	String	in_Config("ReportDirectory").ToString
out_YearlyReportPath	Out	String	YearlyReportPath

10. Invoke the System1\System1_NavigateTo.xaml workflow.
    a. Click Import Arguments and make the following changes:

Name	Direction	Type	Value
in_URL	In	String	in_Config("AcmeUploadYearlyReportURL").ToString

11. Create a new variable of string type called Confirmation.
12. Invoke the System1\System1_UploadYearlyReport.xaml workflow.
    a. Click Import Arguments and make the following changes:

Name	Direction	Type	Value
in_TaxID	In	String	TaxID
in_Year	In	String	“2021”
in_YearlyReportPath	In	String	YearlyReportPath
out_YearlyReportPath	Out	String	Confirmation

13. Invoke the System1\System1_NavigateTo.xaml workflow.
    a. Click Import Arguments and make the following changes:

Name	Direction	Type	Value
in_URL	In	String	in_Config("AcmeUpdateURL").ToString + "/" + WIID

14. Invoke the System1\System1_UpdateWorkItems.xaml workflow.
    a. Click Import Arguments and make the following changes:

Name	Direction	Type	Value
in_Comment	In	String	Confirmation
in_Status	In	String	in_Config("Status").ToString

15. Invoke the System1\System1_NavigateTo.xaml workflow.
    a. Click Import Arguments and make the following changes:

Name	Direction	Type	Value
in_URL	In	String	in_Config("AcmeWorkItemsURL").ToString