Checklist REF without Queue Items

Prerequisites

Identify what type of data the automation needs to process for each transaction; decide the data type for the *TransactionItem* variable.

Configuration in UiPath Studio

Create **new project** using the ***Robotic Enterprise Framework*** template:

Set a proper name for the project – based on the name of the automated process;
Set a proper description for the project.

Edit the project's **configuration file**, found in the project folder, *Data/Config.xlsx:*

Settings sheet:

Set a proper name for the *logF_BusinessProcessName* parameter; usually, this is the same as the name of the project; it is useful to link the process with log messages generated during execution;

Define any additional values required for the process such as input file paths, application paths, parameters to control the automation flow, emails, etc.

Credential assets should also be defined in this sheet, instead of the Assets sheet;

The OrchestratorQueueName and OrchestratorQueueFolder parameters can be deleted.

Constants sheet:

Set the *MaxRetryNumber* value to a value greater than 0 if we want to use the retry mechanism built into the framework.

Optionally, you can change the values of the other items defined in this sheet according to the process being automated.

Assets sheet:

Add entries for all assets (except credential assets) defined in Orchestrator for this process.

Save and close the *Config.xlsx*.

Change the TransactionItem data type

Edit the **Main.xaml** workflow to change the data type for the *TransactionItem* variable from the default *QueueItem* to the one required for the process under automation:

Locate the *TransactionItem* variable in the *Variables* panel; change the *Variable type* to the required type;

Locate the *TransactionData* variable in the *Variables* panel; change the *Variable type* to the required type, if necessary.

In the ***Get Transaction Data*** state:

Locate the *End Process (Stop process requested)* activity and set the value to the default value for the *TransactionItem*'s data type;

Locate the *Invoke GetTransactionData* workflow activity and open the Arguments list; change the data type for the *out_TransactionItem* argument from *QueueItem* to the required type and the *io_TransactionData* from *DataTable* to the required type. Click OK to save the changes and close the dialog;

Expand the Exception section in the *Try GetTransactionData* section and locate the *End Process (Could not get new transaction)* activity; the change here is like the one performed at the beggining of this list.

In the ***Process Transaction*** state:

Locate the *Invoke Process workflow* activity and open the *Arguments* list; change the data type for the *in_TransactionItem* argument from *QueueItem* to the required type; click OK to save the changes and close the dialog;

Locate the Invoke SetTransactionStatus (Success) 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;

Locate the Invoke SetTransactionStatus (BRE) activity and do the same change as the previous one;

Locate the Invoke SetTransactionStatus (SE) activity and do the same change as the previous one.

Applications Used: open/close/kill

Edit the **Framework/InitAllApplications.xaml** workflow; it should contain the activities to start all the applications, perform the authentication and initial configuration required for all transactions in the process;

Edit the **Framework/CloseAllApplication.xaml** workflow; invoke here the activities that should logout and close the applications opened in the *InitAllApplications.xaml* workflow;

Edit the **Framework/KillAllProcesses.xaml** workflow; place here the activities to force close the applications used in the automaton.

Business Process: Transaction Data and Process

Edit *Framework/GetTransactionData.xaml* workflow:

Change the data type for the *out_TransactionItem* argument from *QueueItem* to the required type;

Change the data type for the *io_TransactionData* argument from *DataTable* to the required type, if necessary;

Delete the *Get transaction item* activity (of type *UiPath.Core.Activities.GetQueueItem*);

Store transactions data into *io_TransactionData* argument, if necessary;

Add the activities to check if a new transaction is available;

On the transaction available branch, add the activities necessary to retrieve the information to be processed in one transaction;

On the no more transactions available branch, assign *out_TransactionItem* to stop value;

Set a proper value for the *out_TransactionID* argument in the *Assign out_TransactionID* activity;

Set proper values for *out_TransactionField1* and *out_TransactionField2* arguments in the next two assign activities.

Edit *Process.xaml* workflow:

Change the data type for the *in_TransactionItem* argument from *QueueItem* to the required type;

Place all the activities and logic required to process one transaction.