What is the difference bw classic(Legacy) projects and modern projects.

Windows

* .net framework( windows legacy) will work only on old windows till windows 8 , 10
  + 1
  + 2.x
  + 3.x
  + 4.5x
* .net core windows
  + 1.0
  + 2.0
  + Windows

Uipath projects

Windows Leacy – windows till 10

Windows – create once and try to deploy on multiple platforms

- desktop

- web

- api

- image based

Cross platform – build automation that are not dependent on os apps

- api

Compatibility

If a project is built on windows legacy how can it be upgraded to windows

Dictate

When we open the previous versions of projects in uipath studio in the latest version of uipath studio it will ask you an option stating would you like to convert the previous project into modern project so if you click on continue on the options it will navigate you or it will make the changes necessary to make sure that the project is converted to a windows based project instead of a windows legacy let

ReFramework

1. Main.xaml
   1. Variables
      1. transactiondata
   2. Init
      1. First run sequence
         1. If no info in config
            1. InitallSettings.xaml

Arguments

Configfile path - in

Sheetnames – array of strings - in

Out\_Config – dictionary – out

Read config excel sheet

Read settings

Add credential asset names

for each row

And info to config dictionary

Read constants

for each row

And info to config dictionary

Read assets

For each row

And info to config dictionary

* + - 1. KillAllprocess.xaml
    1. InitAllApplications
       1. In arg – config
  1. Get transaction
     1. Check for stop signal
        1. If no stop
           1. "Framework\GetTransactionData.xaml"

Orchestrator queue name - > config

Orcehstrator folder name - > config

Out transactionitem - > type – Queue item

Type – data row

Type – string if using list of strings

* 1. Process transaction(state)
     1. In\_Transactionitem
     2. In\_config
        1. Try
           1. Process.xaml
           2. Handle success
        2. Catch
           1. Handle bre
           2. Handle se
  2. End process
     1. CloseAllapplicaitons.xaml

1. Transitions
   1. Init -> success -> get transaction data
   2. Init -> failed -> end process
   3. Get transaction -> process -> success
   4. Get transaction -> end process -> nothing
   5. Process -> get transaction -> success
   6. Process -> get transaction -> bre
   7. Process -> init -> SE
2. Variables
3. Arguments
4. Retry in reFramework

19 June 2023

1. Init
   1. First run
      1. There is a folder in which files will be added, each file will be an excel
      2. For each file
         1. Add file path to orchestrator queue
2. Get transaction
   1. Get one transaction item
3. Process
   1. Get file path from queue item
   2. Read data from file
   3. Filter data
      1. If there is no status in the status column, update it as Business exception and for each BRE send email
         1. Throw bre
      2. Else success
4. End process
   1. Send report on how many are successful and how many are failed
5. Read data, filter all true items from all files,
6. Send report with final table with true items to business users.