



SOLUTION DEFINITION DOCUMENT

<REF_CFV_Dispatcher>

INDEX

Document version history	4
Contributors	4
Source documents	4
Signed off by	4
1. Introduction	5
1.1. Required Robotic Resources	5
1.2. Involved Systems/Applications	5
1.3. Scope	5
2. General Solution Description	6
2.1. Solution Flowchart	6
2.2. Solution Description	6
3. Process Description	7
3.1. Initialization	7
3.2. Get Transaction Data	8
3.3. Process Transaction	9
3.4. End Process	11
3.5. LogIn_ACME	11
3.6. LogOut_ACME	13
3.7. Download_Excel	14
3.8. Load_Queue	16
3.9. Manage_Exceptions	17
4. Environment Variables	20
4.1. Config	20
4.2. Global Variables	20
4.3. Assets	20
4.4. Queue	20
5. Exceptions	21
5.1. Business Exceptions	21
5.2. System Exceptions	21
6. Settings and Alerts	21
6.1. Scheduling and execution settings	21
6.2. Alerts	21

7.	Data and credentials.....	22
7.1.	Data storage	22
7.2.	Data privacy	22
7.3.	Data preservation	22
7.4.	Credentials.....	22

Document version history

Date	Version	Description	Author
01.07.23	1.	Document creation	FJV

Contributors

Name	Function	Responsibility
FJV	Solution definition	Solution creation

Source documents

Date	Version	Title	Author
01.07.23	0.1	PDD Contabilidad Facturas Ventas (0.1).pdf	Andrés Díaz Romero

Signed off by

Name	Department	Responsibility
Piciorang, C.E.	Intelligent Automation	Project Manager

1.Introduction

1.1. Required Robotic Resources

Sections	Description
Machines used:	1
- Tests	127.0.0.1
- Pro	127.0.0.1
Current Sessions	1
Future Sessions	1
Frequency	Daily
Workload	Unknown
Daily System Availability	Complete

1.2. Involved Systems/Applications

Systems/Applications	Version	Description
MS Windows	10	Use the computer
Firefox	115	Open the ACME webpage
ACME webpage	Unknown	Get the invoices need them in the process
MS Excel	2019	Open/Modify Spreadsheets
Gmail	Unknown	Alert the exceptions by email

1.3. Scope

1.3.1. Functional Requirements

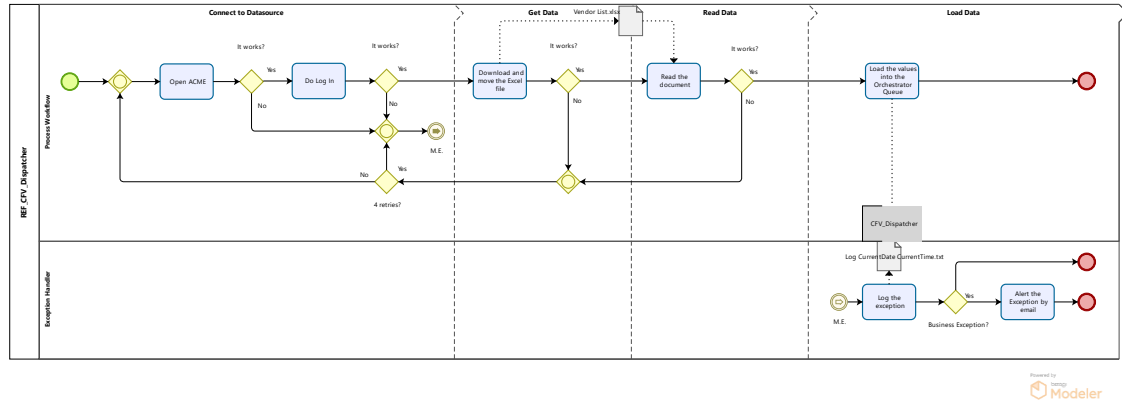
Requirement	Description
1	Connect to datasource
2	Get the data
3	Read the data
4	Load the data

1.3.2. Non-Functional Requirements

Requirement	Description
1	Log in/ Log out
2	Authentication
3	Exception Handling
4	Exception Alert

2. General Solution Description

2.1. Solution Flowchart



2.2. Solution Description

The process consists of the following dependencies:

#	Sequences	Description
1	LogIn_ACME.xaml	Do Login in the ACME website and prepare the browser to the process transactions.
2	LogOut_ACME.xaml	Close the session in the ACME website at process' end.
3	Download_Excel.xaml	Download "Vendor List.xlsx" from the site.
4	Load_Queue.xaml	Load the values from "Vendor List.xlsx" into the Orchestrator Queue.
5	Manage_Exceptions.xaml	Handle the possible Exceptions during execution.

3. Process Description

The process, "REF_CFV_Dispatcher", use the Robotics Enterprise Framework (REF) from UIPATH to automate a very common office task: get an Excel file with the information about the providers, clean the data and once finished, send the information. The errors that occur during the execution are recorder in a log, and only those configured are notified by email.

3.1. Initialization

3.1.1. InitAllSettings

Argument	Value
out_Config["ExcetPathOld"]	Environment.GetFolderPath(Environment.SpecialFolder.Desktop).Replace("Desktop","Downloads\\")
out_Config["ExcelPathOld"]	System.DateTime.Now.ToString("HH.mm")

3.1.2. KillAllProcesses

3.1.2.1. Kill "firefox"

3.1.3. InitAllApplications

3.1.3.1. Open Firefox

3.1.3.2. Invoke LogOut_ACME.xaml

3.1.3.3. Invoke LogIn_ACME.xaml

Argument	Value
in_OrchestratorName	in_Config["OrchestratorName"].ToString()
in_OrchestratorAssetName	in_Config["OrchestratorAssetName"].ToString()

3.1.4. Transitions

3.1.4.1. Business Exception to End Process

3.1.4.1.1. Invoke Manage_Exceptions.xaml

Argument	Value
in_ErrorCode	BusinessException.Message
in_ExceptionsLogs	Config["ExceptionsLogsPath"].ToString()
in_ExceptionsListPath	Config["ExceptionsListPath"].ToString()
in_ExceptionsListSheet	Config["ExceptionsListSheet"].ToString()
in_ExceptionsListFile	Config["ExceptionsListFile"].ToString()
in_TimeStamp	Config["TimeStamp"].ToString()

3.1.4.2. System Exception to End Process

3.1.4.2.1. Invoke Manage_Exceptions.xaml

Argument	Value
in_ErrorCode	SystemException.Message
in_ExceptionsLogs	Config["ExceptionsLogsPath"].ToString()
in_ExceptionsListPath	Config["ExceptionsListPath"].ToString()
in_ExceptionsListSheet	Config["ExceptionsListSheet"].ToString()
in_ExceptionsListFile	Config["ExceptionsListFile"].ToString()
in_TimeStamp	Config["TimeStamp"].ToString()

3.1.4.3. Successful to Get Transaction Data

3.2. Get Transaction Data

3.2.1. GetTransactionData

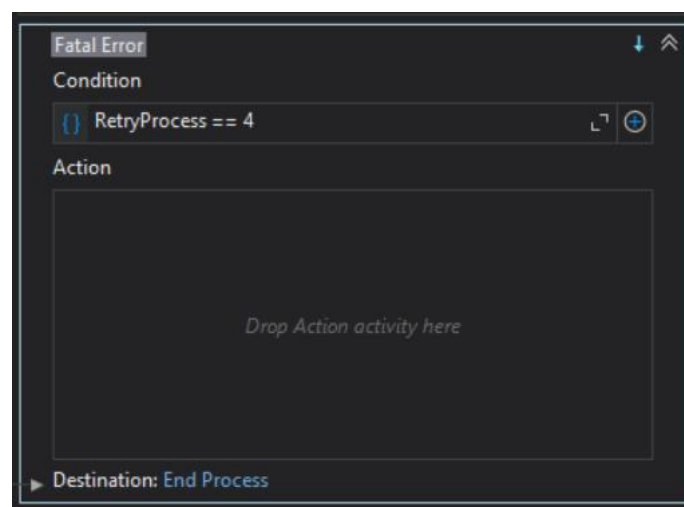
Argument	Value
out_Case	TransactionCase

3.2.1.1. Disable Activity "Retry Get transaction item"

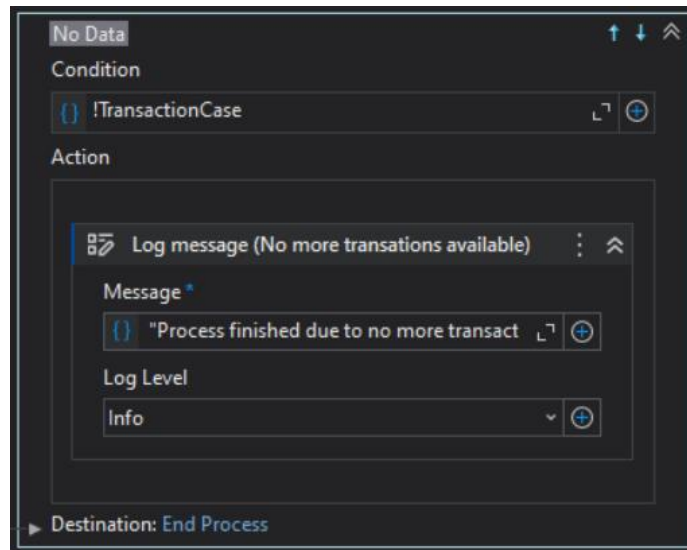
3.2.1.2. Assign out_Case = true

3.2.2. Transitions

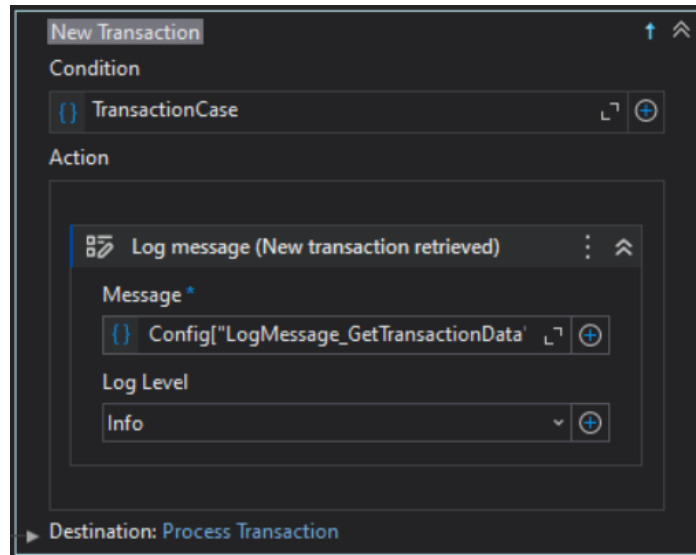
3.2.2.1. Fatal Error to end Process



3.2.2.2. No Data to End Process



3.2.2.3. New Transaction to Process Transaction



3.3. Process Transaction

3.3.1. Process

3.3.1.1. Invoke Download_Excel.xaml

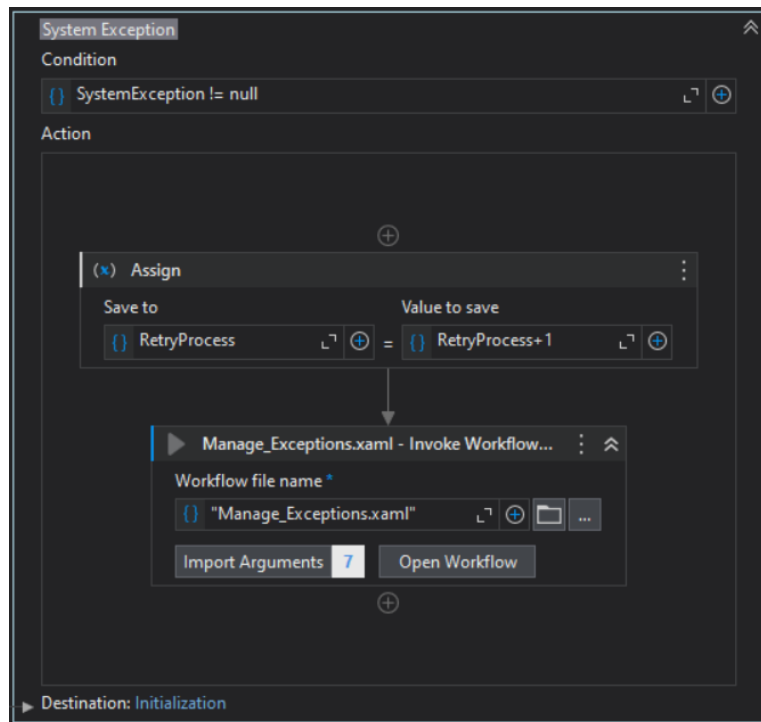
Argument	Value
In_ExcelPathOld	in_Config["ExcelPathOld"].ToString()
In_ExcelPathNew	in_Config["ExcelPathNew"].ToString()
In_ExcelName	in_Config["ExcelName"].ToString()

3.3.1.2. Invoke Load_Queue.xaml

Argument	Value
in_ExcelPath	in_Config["ExcelPathNew"].ToString()
in_ExcelSheet	in_Config["ExcelSheet"].ToString()
in_OrchestratorQueueName	in_Config["OrchestratorQueueName"].ToString()
in_OrchestratorName	in_Config["OrchestratorName"].ToString()
in_ExcelName	in_Config["ExcelName"].ToString()

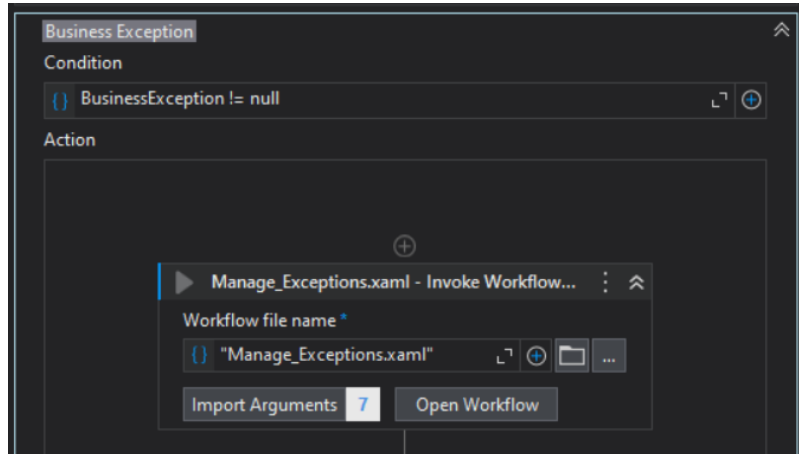
3.3.2. Transitions

3.3.2.1. System Exception to Initialization



Argument	Value
in_ErrorCode	SystemException.Message
in_ExceptionsLogs	Config["ExceptionsLogsPath"].ToString()
in_ExceptionsListPath	Config["ExceptionsListPath"].ToString()
in_ExceptionsListSheet	Config["ExceptionsListSheet"].ToString()
in_ExceptionsListFile	Config["ExceptionsListFile"].ToString()
in_TimeStamp	Config["TimeStamp"].ToString()

3.3.2.2. Business Exception to End Process



i. Invoke Manage_Exceptions.xaml

Argument	Value
in_ErrorCode	BusinessException.Message
in_ExceptionsLogs	Config["ExceptionsLogsPath"].ToString()
in_ExceptionsListPath	Config["ExceptionsListPath"].ToString()
in_ExceptionsListSheet	Config["ExceptionsListSheet"].ToString()
in_ExceptionsListFile	Config["ExceptionsListFile"].ToString()
in_TimeStamp	Config["TimeStamp"].ToString()

ii. Use Desktop Outlook App

- a. Account = Config["ExceptionsMailSender"].ToString()
- b. Send Outlook Mail Message
 - i. To = Config["ExceptionsMail"].ToString()
 - ii. Subject = "Business Exception "+BusinessException.Message
 - iii. Body = BusinessException.Source

3.3.2.3. Success to End Process

3.4. End Process

3.4.1. CloseAllApplications

3.4.1.1. Invoke LogOut_ACME.xaml

3.4.1.2. Kill "firefox"

3.4.2. Invoke KillAllProcesses

3.4.2.1. Kill "firefox"

3.5. LogIn_ACME

Get the Credentials from the Orchestrator Assets, Type user and password, Do click on the button

3.5.1. Variables:

3.5.1.1. c_acme_user = String, This

3.5.1.2. c_acme_pass = SecureString, This

3.5.2. <Try>

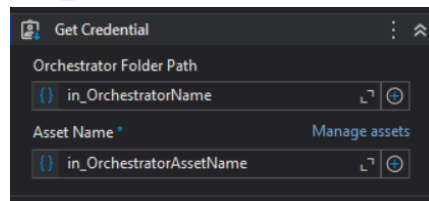
3.5.2.1. <Use Browser "Firefox"> = ""

3.5.2.1.1. <Try>

3.5.2.1.1.1. <Get Credential> = in_OrchestratorName, in_OrchestratorAssetName

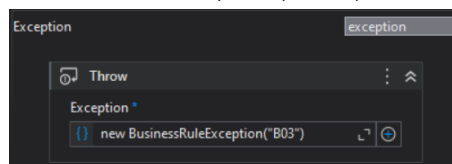
3.5.2.1.1.1.1. Username = c_acme_user

3.5.2.1.1.1.2. Password = c_acme_pass

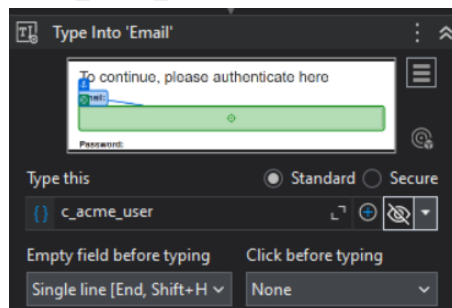


3.5.2.1.2. <Catch>

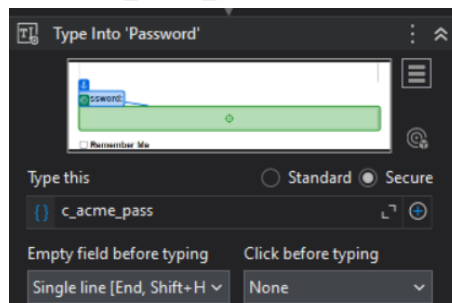
3.5.2.1.2.1. <Throw> = new BusinessException("B03")



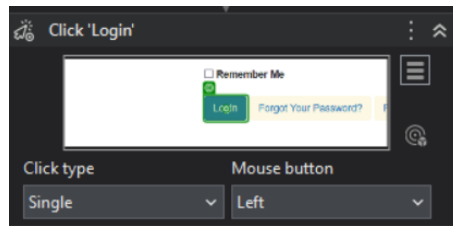
3.5.2.1.3. <Type Into "Email"> = c_acme_user



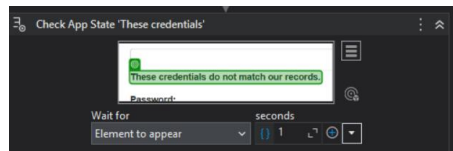
3.5.2.1.4. <Type Into "Password"> = c_acme_pass



3.5.2.1.5. <Click "Login">

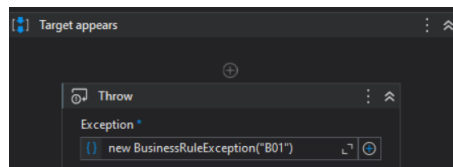


3.5.2.1.6. <Check App State "These Credentials"> = Element to appear, 1s



3.5.2.1.6.1. <Target appears>

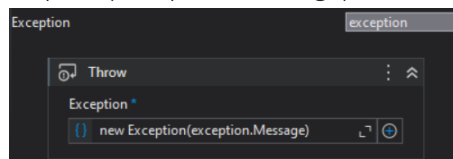
3.5.2.1.6.1.1. <Throw> = new BusinessException("B01")



3.5.3. <Catches>

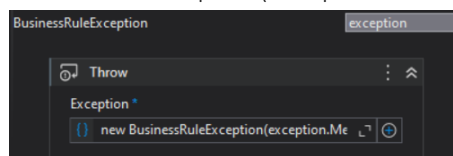
3.5.3.1. Exception = exception

3.5.3.1.1. <Throw> = new Exception(exception.Message)



3.5.3.2. BusinessException = exception

3.5.3.2.1. <Throw> = new BusinessException(exception.Message)



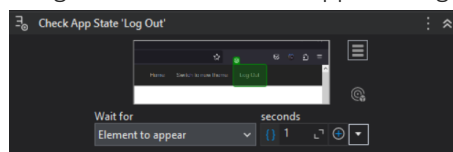
3.6. LogOut_ACME

Do Log Out from the ACME webpage, Check if the session is active, THEN click on LOG OUT

3.6.1. <Try>

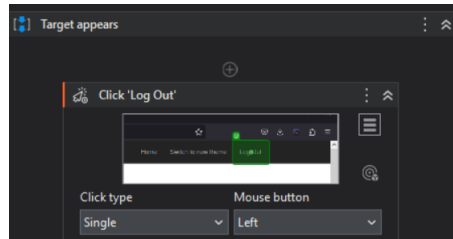
3.6.1.1. <Use Browser "Firefox"> = ""

3.6.1.1.1. <Check App State "Log Out"> = Element to appear "Log Out", 1s



3.6.1.1.1.1. <Target appears>

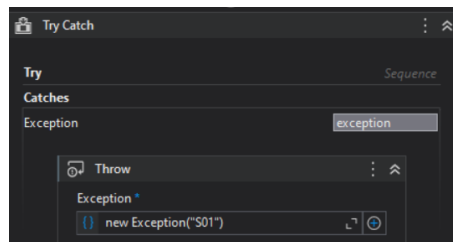
3.6.1.1.1.1.1. <Click "Log Out">



3.6.2. <Catch>

3.6.2.1. Exception

3.6.2.1.1. <Throw> = new Exception("S01")



3.7. Download_Excel

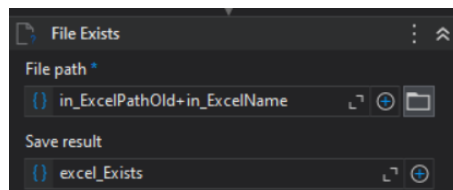
Access to the ACME webpage, Download Vendor list Excel, Move the file to the Vendor folder

3.7.1. Variables:

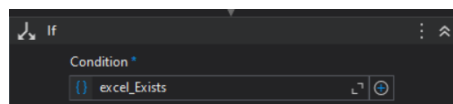
3.7.1.1. excel_Exists = Boolean, This

3.7.1.2. folder_Exists = Boolean, This

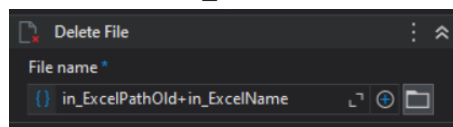
3.7.2. <File Exists> = in_ExcelPathOld + in_ExcelName, excel_Exists



3.7.2.1. IF excel_Exists THEN



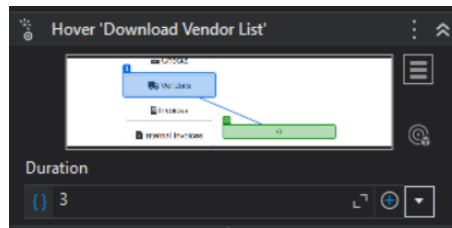
3.7.2.1.1. <Delete File> in_ExcelPathOld + in_ExcelName



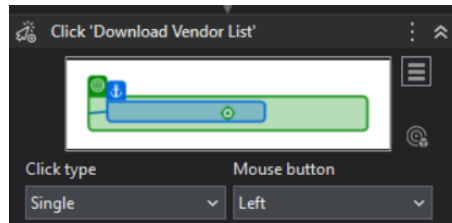
3.7.3. <Try>

3.7.3.1. <Use Browser> Firefox = ""

3.7.3.1.1. <Hover> 'Download Vendor List'



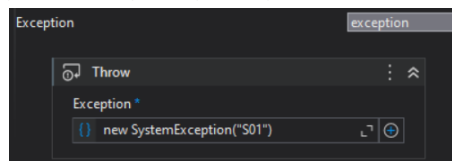
3.7.3.1.2. <Click> 'Download Vendor List'



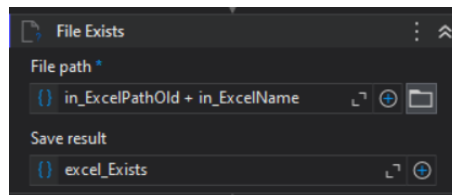
3.7.4. <Catch>

3.7.4.1. Exception = exception

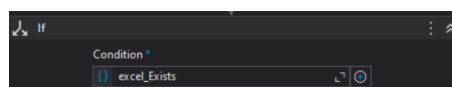
3.7.4.1.1. <Throw> = new SystemException("S01")



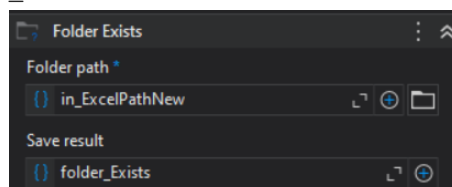
3.7.5. <File Exists> = in_ExcelPathOld + in_ExcelName, excel_Exists



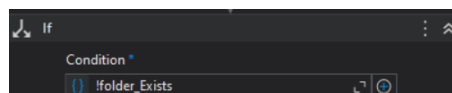
3.7.6. IF excel_Exists THEN



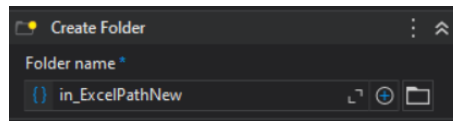
3.7.6.1. <Folder Exists> = in_ExcelPathNew



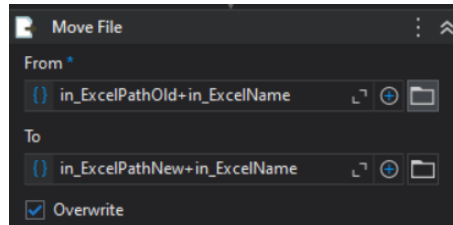
3.7.6.2. IF !folder_Exists THEN



3.7.6.2.1. <Create Folder> = in_ExcelPathNew

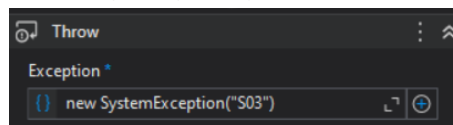


3.7.6.3. <Move File> = in_ExcelPathOld + in_ExcelName, in_ExcelPathNew + in_ExcelName, Overwrite



3.7.7. ELSE

3.7.7.1. <Throw> = new SystemException("S03")



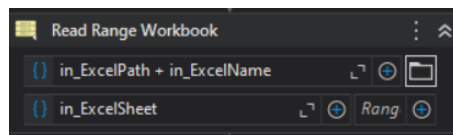
3.8. Load_Queue

Read the Excel File "Vendor list.xlsx", Upload the values to the Orchestrator Queue

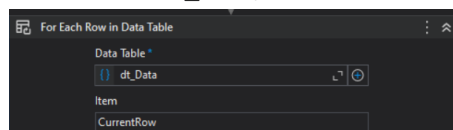
3.8.1. <Try>

3.8.1.1. <Read Range Workbook> = in_ExcelPath + in_ExcelName, in_ExcelSheet

3.8.1.1.1. DataTable = dt_Data



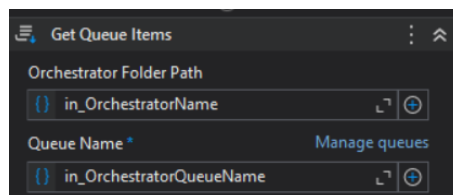
3.8.1.2. <For Each Row in Data Table> = dt_Data, CurrentRow



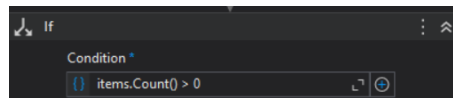
3.8.1.2.1. <Get Queue Items> = in_OrchestratorName, in_OrchestratorQueueName(New, In progress, Successful)

3.8.1.2.1.1. Reference = CurrentRow["Vendor code"].ToString()

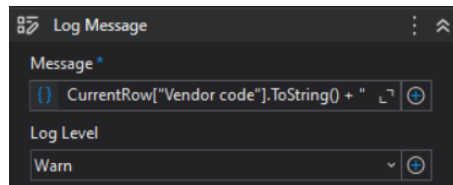
3.8.1.2.1.2. Output = items



3.8.1.2.2. IF items.Count() > 0 THEN



3.8.1.2.2.1. <Log Message> = CurrentRow["Vendor code"].ToString()

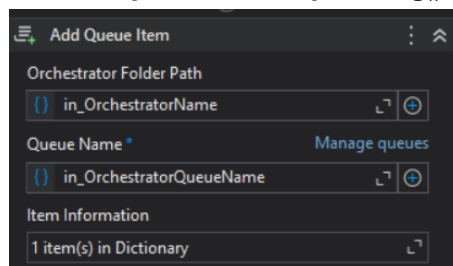


3.8.1.2.3. ELSE

3.8.1.2.3.1. <Add Queue Item> in_OrchestratorName, in_OrchestratorQueueName

3.8.1.2.3.1.1. Dictionary = VendorID, String, CurrentRow["Vendor code"].ToString()

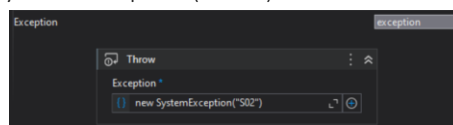
3.8.1.2.3.1.2. Reference = CurrentRow["Vendor code"].ToString()



3.8.2. <Catch>

3.8.2.1. Exception = exception

3.8.2.1.1. <Throw> = new SystemException("S02")



3.9. Manage_Exceptions

Check the Error Code in the Exceptions Excel. If the Error Code is unknown, it will be indicated as unexcepted Exception; If the Excel file cannot be open or cannot be found, it will notify it as an error instead the error message in the log file. Create a new log file with the error message.

3.9.1. Variables:

3.9.1.1. timeStamp_Message = String, This

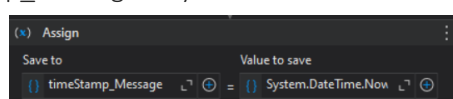
3.9.1.2. Dt_Data = DataTable, This

3.9.1.3. excel_Exists = Boolean, This

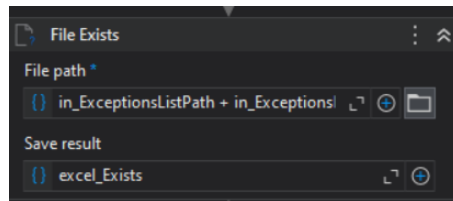
3.9.1.4. folder_Exists = Bollean, This

3.9.1.5. exception_Message = String, This, " unexpected exception"

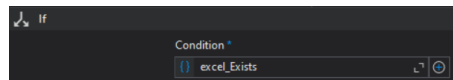
3.9.2. <Assign> = timeStamp_Message = System.DateTime.Now.ToString("HH.mm.ss ")



3.9.3. <File Exists> = in_ExceptionsListPath + in_ExceptionsListFile, excel_Exists

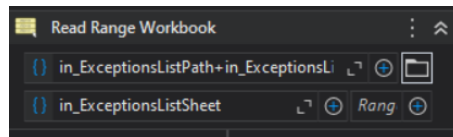


3.9.4. IF excel_Exists THEN

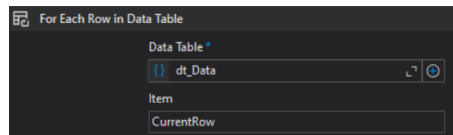


3.9.4.1. <Read Range Workbook> = in_ExceptionsListPath + in_ExceptionsListFile,
in_ExceptionsListSheet

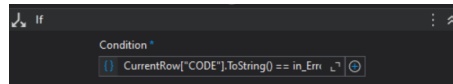
3.9.4.1.1. DataTable = dt_Data



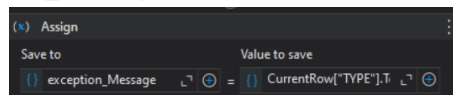
3.9.4.2. <For Each Row in Data Table> = dt_Data, CurrentRow



3.9.4.2.1. IF CurrentRow["CODE"].ToString() == in_ErrorCode



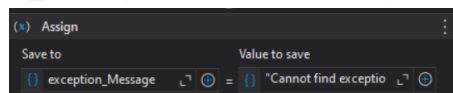
3.9.4.2.1.1. <Assign> exception_Message = CurrentRow["MESSAGE"].ToString()



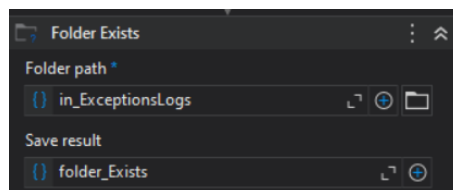
3.9.4.2.1.2. <Break>

3.9.5. ELSE

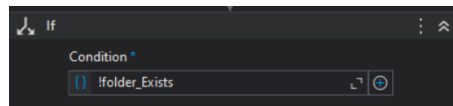
3.9.5.1. <Assign> = exception_Message = "Cannot find exception list"



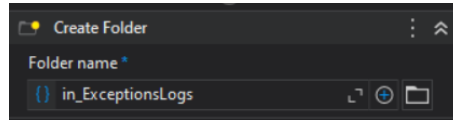
3.9.6. <Folder Exists> = in_ExceptionsLogs, folder_Exists



3.9.7. IF !folder_Exists THEN



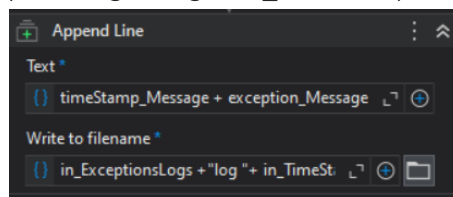
3.9.7.1. <Create Folder> = in_ExceptionsLogs



3.9.8. <Append Line>

3.9.8.1. Text = timeStamp_Message+ exception_Message

3.9.8.2. Filename = in_ExceptionsLogs +"log "+ in_TimeStamp + ".txt"



4.Environment Variables

4.1. Config

Name	Value
OrchestratorQueueName	CFV_Queue
OrchestratorName	orchestrator@company.com's workspace
OrchestratorAssetName	C_ACME
ExceptionsListPath	Data\
ExceptionsListFile	CFV_Exceptions.xlsx
ExceptionsListSheet	Dispatcher
ExceptionsLogsPath	Logs\
ExceptionsMailReceiver	receiver@company.com
ExceptionsMailSender	sender@company.com
URL_ACME	https://acme-test.uipath.com/login
ExcelURL	https://acme-test.uipath.com/testdata/vendor/Vendor%20List.xlsx
ExcelName	Vendor List.xlsx
ExcelPathOld	
ExcelPathNew	C:\Vendors\
ExcelSheet	Vendor list
TimeStamp	orchestrator@company.com's workspace

4.2. Global Variables

Variable	Type	Scope	Default
RetryProcess	Int32	General Business	0
TransactionCase	Boolean	General Business	

4.3. Assets

Asset	Type	Path
C_ACME	Credential	orchestrator@company.rpa

4.4. Queue

Queue	ID	Path
CFV_Queue	VendorID	orchestrator@company.rpa

5.Exceptions

The Exception list, stored in the Excel file CVV_Exceptions, includes the following exceptions according to their case:

5.1. Business Exceptions

Code	Description
B01	Wrong credentials in ACME
B02	Cannot use the Excel document
B03	Wrong Orchestrator Name or Asset in Config

5.2. System Exceptions

Code	Description
S01	Cannot open the ACME webpage
S02	Cannot load the values to the Orchestrator Queue
S03	Cannot download the Excel File

6.Settings and Alerts

6.1. Scheduling and execution settings

The process Schedule will mark the start time of this at: Still to be defined.

6.2. Alerts

The following handled Exceptions will be notified according to:

Exception	Action	Notes
Business Exceptions	Send an email to the person in charge:	receiver@company.com
System Exceptions	Perform up to N retries before ending	Currently 4 times.
Exceptions (all)	Write to the errors log file	"log yyyy.MM.dd, HHJ.mm.txt"

7.Data and credentials

7.1. Data storage

The data are stored on local disk and the email sent/in box.

Path	Description
\Downloads\	Default user's downloads folder
C:\Vendors	Folder to store the data before to be processed
C:\Output_Vendors\	Folder to store the data after being processed
\Logs\	Folder to store the exception logs

7.2. Data privacy

Certain data privacy policies are being carried out: It requires a password to access to the system, to the email manager and the Orchestrator Queue.

7.3. Data preservation

The data will be saved for the next four years, according to the Spanish tax law.

Folders could be created by the robot even if the user deletes them.

In charge of the possible credential reset: Still to be defined.

7.4. Credentials

Credentials are stored in the Orchestrator Assets.