SOLUTION DEFINITION DOCUMENT

<REF_CFV_ Performer>

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 14
3.7. Extract_Data 15
3.8. Insert_Data 16
3.9. Manage_Exceptions 18
4. Environment Variables 21
4.1. Config 21
4.2. Global Variables 21
4.3. Assets 21
4.4. Queue 21
5. Exceptions 22
5.1. Business Exceptions 22
5.2. System Exceptions 22
6. Settings and Alerts 22
6.1. Scheduling and execution settings 22

6.2. Alerts 22
7. Data and credentials 23
7.1. Data storage 23
7.2. Data privacy 23
7.3. Data preservation 23
7.4. Credentials 23

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: - Tests - Pro	1 127.0.0.1 127.0.0.1
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

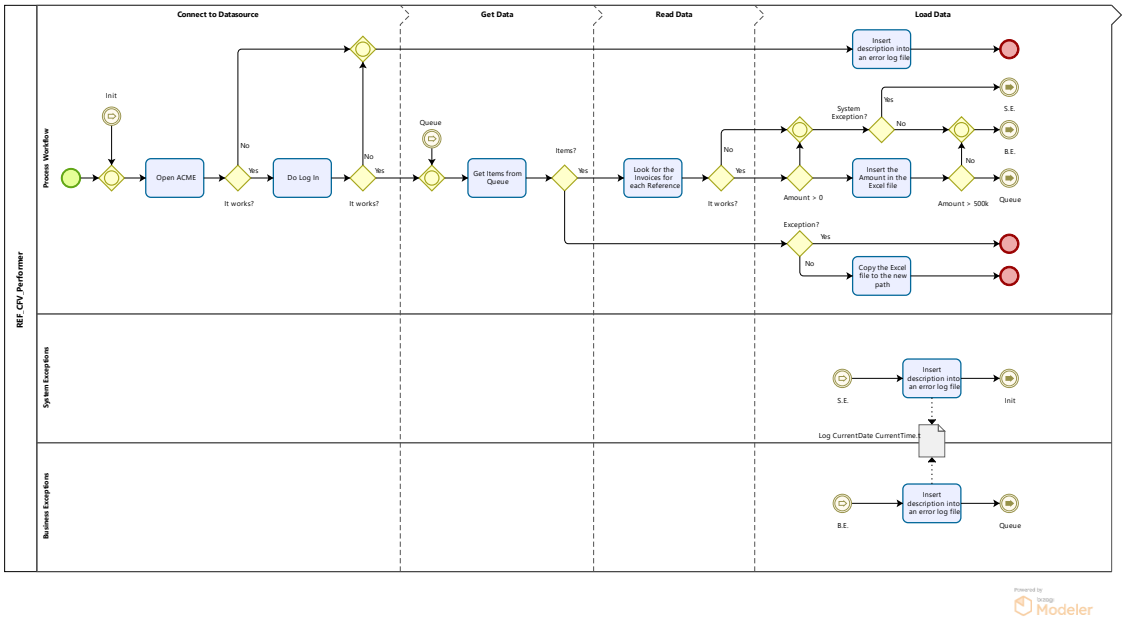
Requirement	Description
1	Connect to datasource
2	Get the data
3	Read the data
4	Load the data

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	Extract_Data.xaml	Get a list of Vendor ID from the Orchestrator Queue and look for, in the ACME webpage, the invoices for each ID.
4	Insert_Data.xaml	Analyse the data and load the right values, according to the Business rules, into the Excel file "Vendor List.xlsx".
5	Manage_Exceptions.xaml	Handle the possible Exceptions during execution.

3. Process Description

The process, “REF_CFV_Performer”, 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

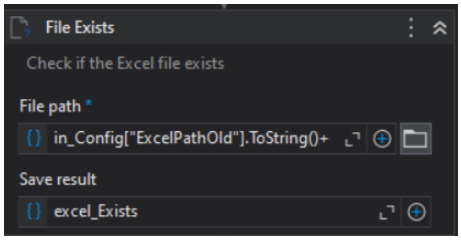
Argument	Value
out_Config["ExcelPathOld"]	System.DateTime.Now.ToString("HH.mm")

3.1.2. KillAllProcesses

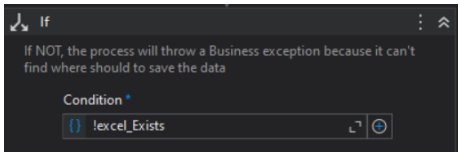
3.1.2.1. Kill “firefox”

3.1.3. InitAllApplications

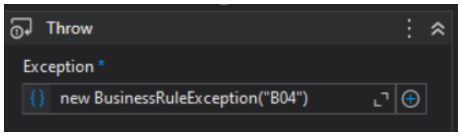
3.1.3.1. File Exists = excel_Exists



3.1.3.2. If !excel_Exists THEN



3.1.3.2.1. Throw a BusinessRuleException(“B04”)



3.1.3.3. Open Firefox

3.1.3.4. Invoke LogOut_ACME.xaml

3.1.3.5. 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

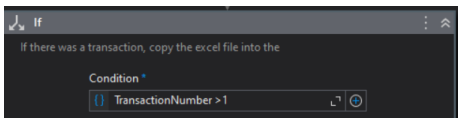
3.2. Get Transaction Data

3.2.1. GetTransactionData

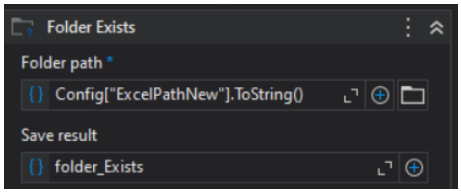
3.2.2. Transitions

3.2.2.1. No Data to End Process

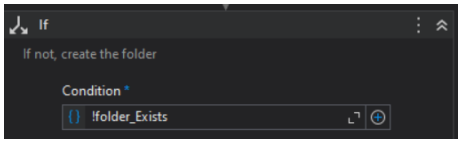
3.2.2.1.1. If TransactionNumer > 1 THEN



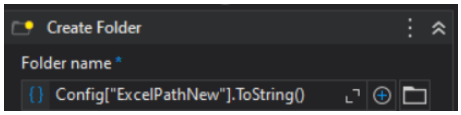
3.2.2.1.1.1. Folder Exists = folder_Exists



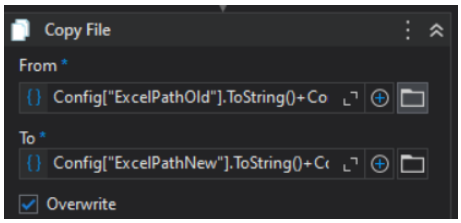
3.2.2.1.1.2. If !folder_Exists THEN



3.2.2.1.1.3. Create Folder



3.2.2.1.2. Copy File



3.2.2.2. New Transaction to Process Transaction

3.3. Process Transaction

3.3.1. Process

3.3.1.1. Invoke Extract_Data.xaml

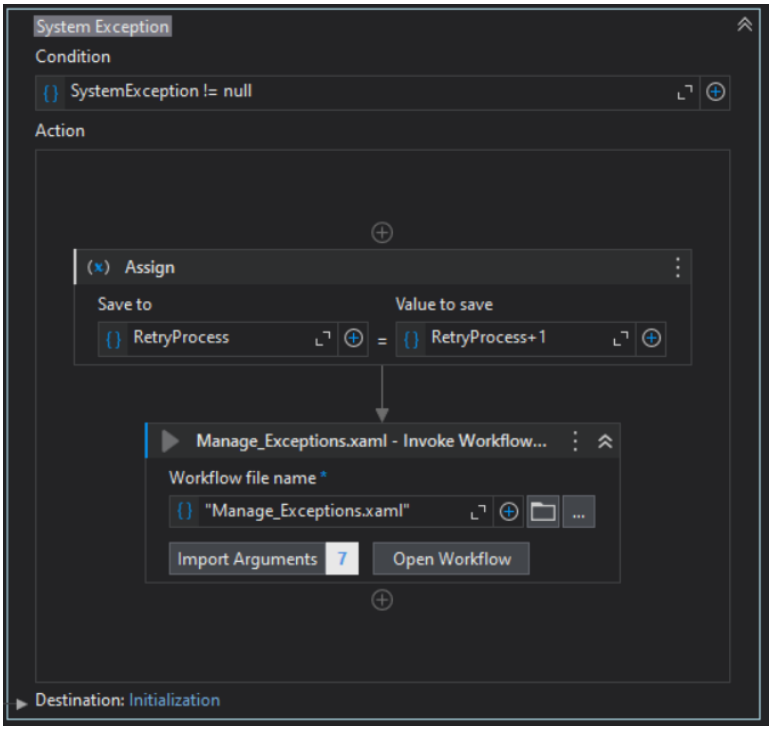
Argument	Value
in_ID	in_TransactionItem.SpecificContent["VendorID"].ToString()
out_dt_Data	dt_Data

3.3.1.2. Invoke Insert_Data.xaml

Argument	Value
in_ID	in_TransactionItem.SpecificContent["VendorID"].ToString()
in_dt_Data	dt_Data
In_ExcelPathOld	in_Config["ExcelPathOld"].ToString()
In_ExcelPathNew	in_Config["ExcelPathNew"].ToString()
In_ExcelName	in_Config["ExcelName"].ToString()
In_MaxAmount	double.Parse(in_Config["MaxAmount"].ToString())

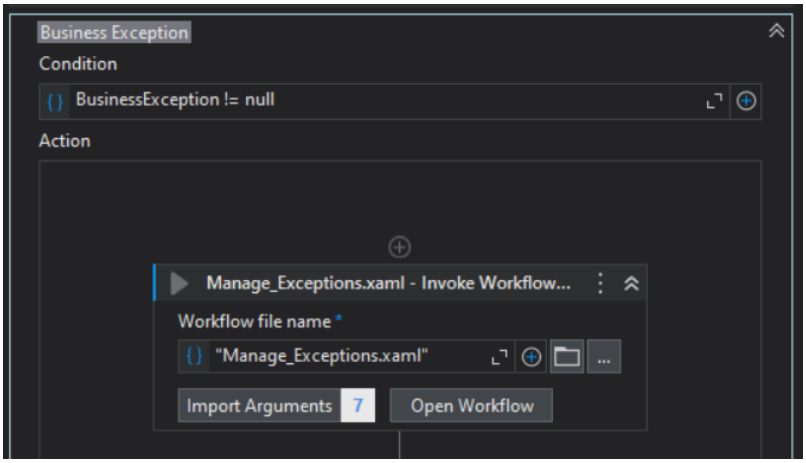
3.3.2. Transitions

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 Mesage
      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>

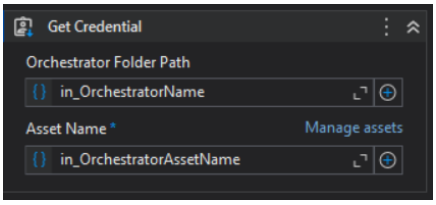
3.5.2.1. <Use Browser "Firefox"> = ""

3.5.2.1.1. <Try>

3.5.2.1.1.1. <Get Credential> = in_OrchestratorName, in_OrchestratorAssetName

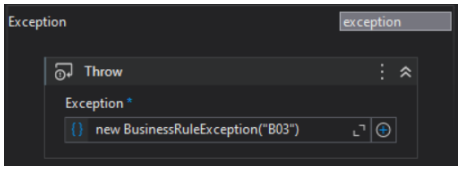
3.5.2.1.1.1.1. Username = c_acme_user

3.5.2.1.1.1.2. Password = c_acme_pass

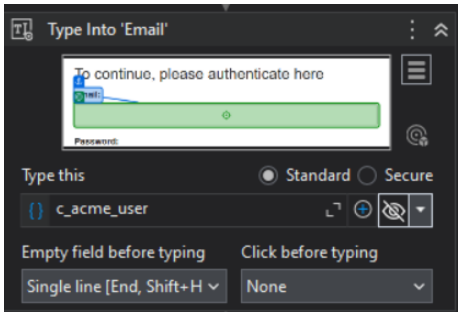


3.5.2.1.2. <Catch>

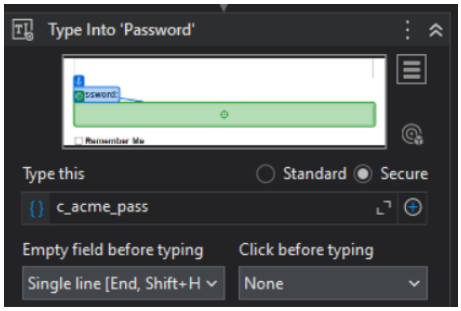
3.5.2.1.2.1. <Throw> = new BusinessRuleException("B03")



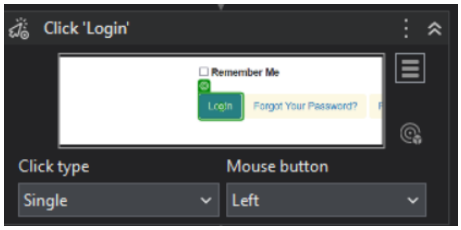
3.5.2.1.3. <Type Into "Email"> = c_acme_user



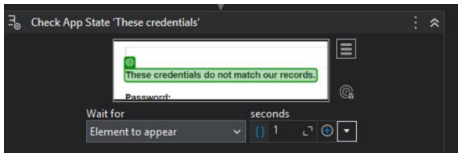
3.5.2.1.4. <Type Into "Password"> = c_acme_pass



3.5.2.1.5. <Click "Login">

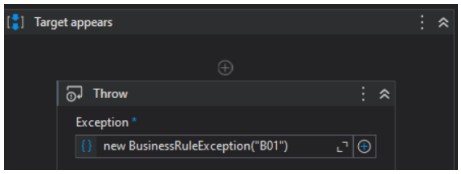


3.5.2.1.6. <Check App State "These Credentials"> = Element to appear, 1s

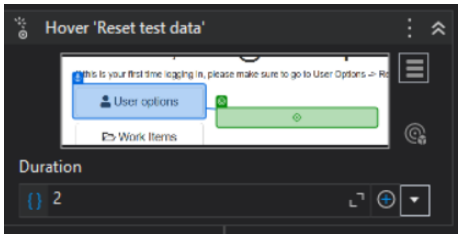


3.5.2.1.6.1. <Target appears>

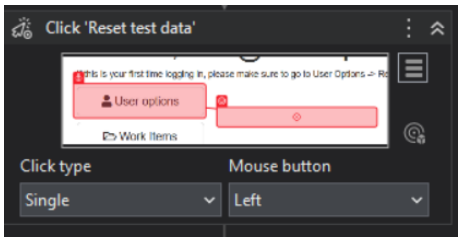
3.5.2.1.6.1.1. <Throw> = new BusinessRuleException("B01")



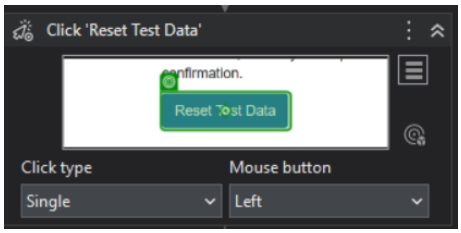
3.5.3. <Hover> “Reset test data”



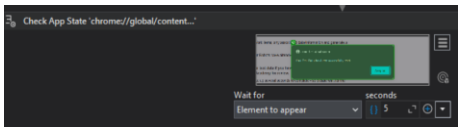
3.5.4. <Click> “Reset test data”



3.5.5. <Click> “Reset Test Data” button

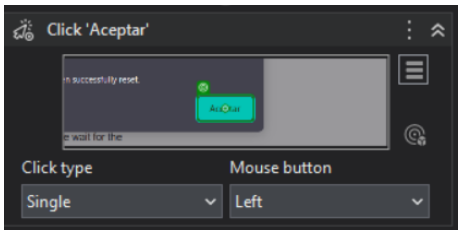


3.5.6. <Check App State> “Acept”

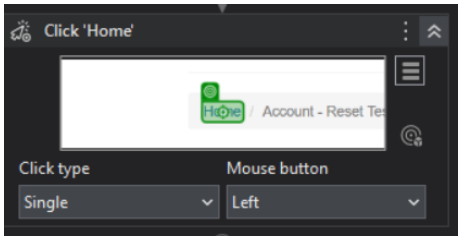


3.5.6.1. <Target appears>

3.5.6.1.1. <Click> “Aceptar”



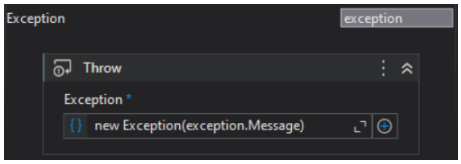
3.5.7. <Click> “Home”



3.5.8. <Catches>

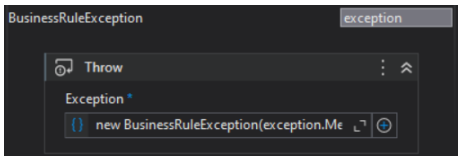
3.5.8.1. Exception = exception

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



3.5.8.2. BusinessRuleException = exception

3.5.8.2.1. <Throw> = new BusinessRuleException(exception.Message)



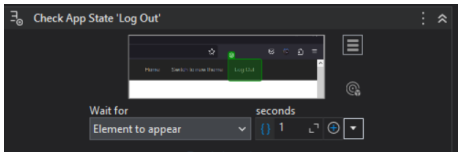
3.6. LogOut_ACME

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

3.6.1. <Try>

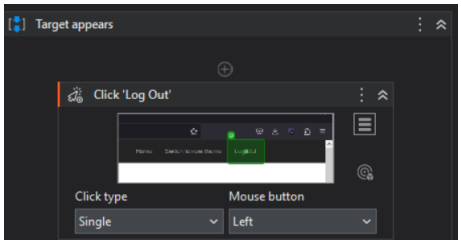
3.6.1.1. <Use Browser "Firefox"> = ""

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



3.6.1.1.1.1. <Target appears>

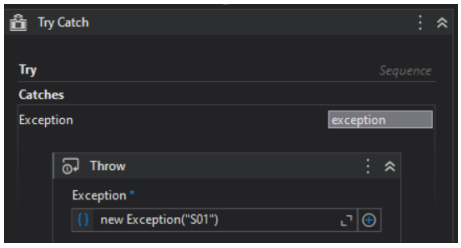
3.6.1.1.1.1.1. <Click "Log Out">



3.6.2. <Catch>

3.6.2.1. Exception

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

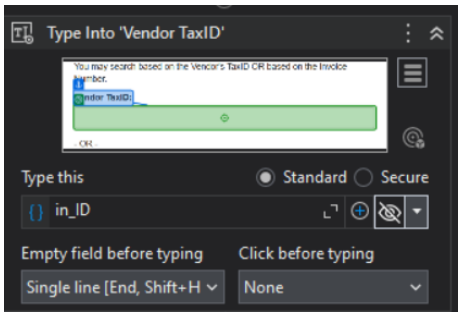


3.7. Extract_Data

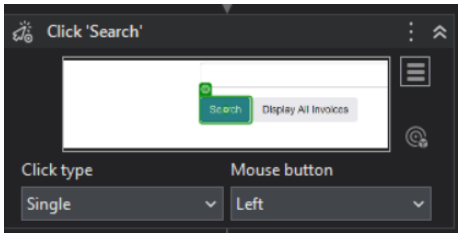
3.7.1. <Try>

3.7.1.1. <Use Browser “Firefox”>

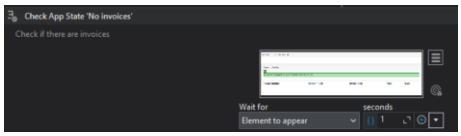
3.7.1.1.1. <Type Into “Vendor TaxID”> = in_ID



3.7.1.1.1.1. <Click> “Search”

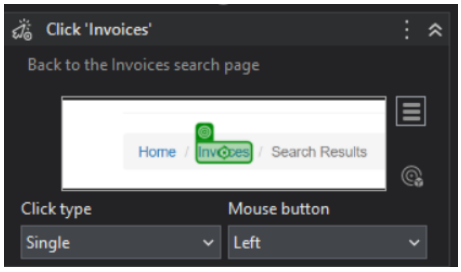


3.7.1.1.2. <Check App State> “No invoices”

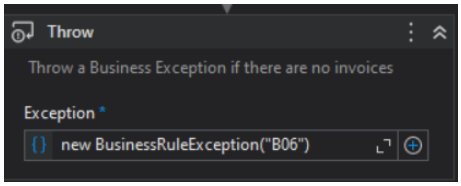


3.7.1.1.3. <Target appears>

3.7.1.1.3.1. <Click> “Invoices”

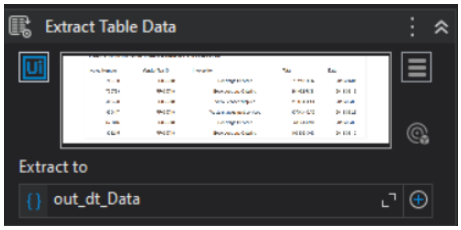


3.7.1.1.3.2. <Throw> = new BusinessRuleException(“B06”)

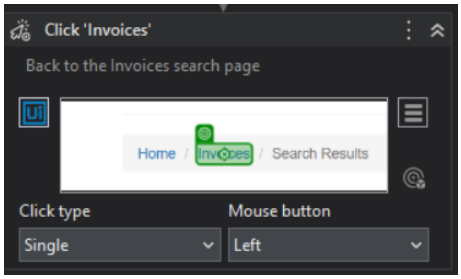


3.7.1.1.4. <Target does not appear>

3.7.1.1.4.1. <Etract Table Data> = out_dt_Data



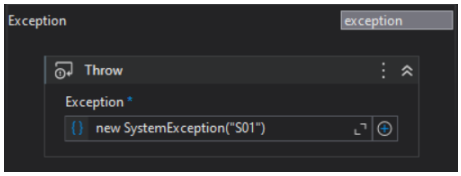
3.7.1.1.4.2. <Click> “Invoices”



3.7.2. <Catches>

3.7.2.1. Exception = exception

3.7.2.1.1. <Throw> = new SystemException(“S01”)



3.7.2.2. BusinessRuleException = exception

3.7.2.2.1. <Throw> = exception

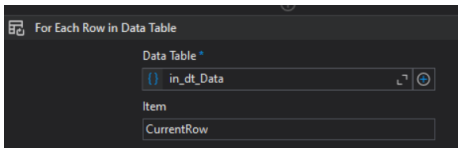
3.8. Insert_Data

3.8.1. Variables

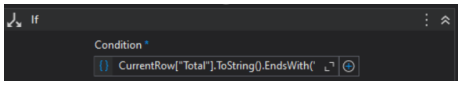
3.8.1.1. Result = Double, This, “0”

3.8.1.2. Dt_Data = DataTable, This

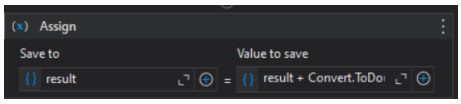
3.8.2. <For Each Row in Data Table> = in_dt_Data, CurrentRow



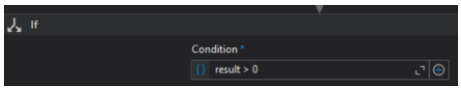
3.8.2.1. If CurrentRow["Total"].ToString().EndsWith("EUR") THEN



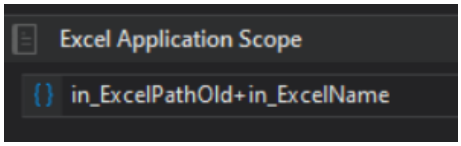
3.8.2.1.1. <Assign> result = result + Convert.ToDouble(CurrentRow["Total"].ToString().Replace(" EUR",""))



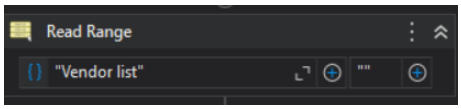
3.8.3. If result > 0 THEN



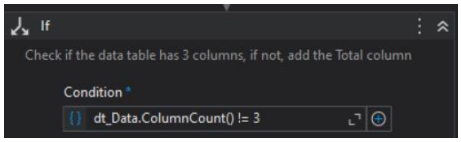
3.8.3.1. <Excel Application Scope> = in_ExcelPathOld + in_ExcelName



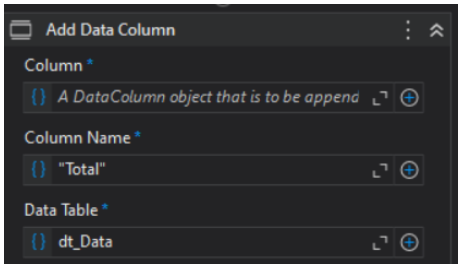
3.8.3.1.1. <Read Range> = “Vendor list”



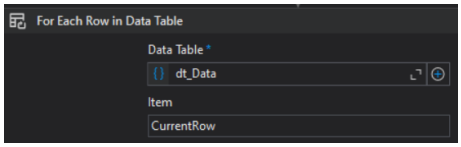
3.8.3.1.2. dt_Data.ColumnCount() != 3 THEN



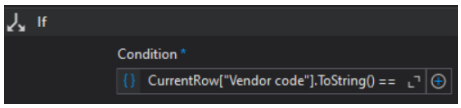
3.8.3.1.2.1. <Add Data Column>



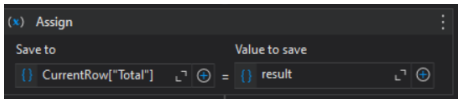
3.8.3.1.3. <For Each Row in Data Table> = dt_Data, CurrentRow



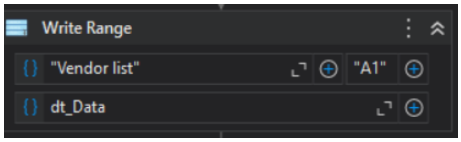
3.8.3.1.3.1. If CurrentRow["Vendor code"].ToString() == in_ID.ToString() THEN



3.8.3.1.3.1.1. <Assign> CurrentRow["Total"] = result

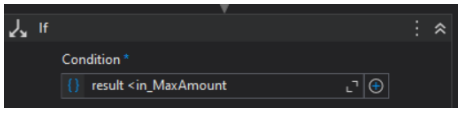


3.8.3.1.3.1.2. <Write Range> = “Vendor list”, dt_Data

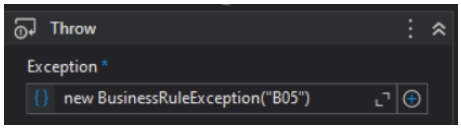


3.8.3.1.3.1.3. <Break>

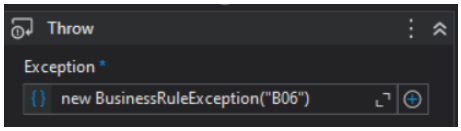
3.8.3.2. If result <in_MaxAmount THEN



3.8.3.2.1. <Throw> = new BusinessRuleException(“B05”)



3.8.4. ELSE



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

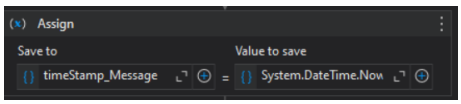
3.9.1.2. dt_Data = DataTable, This

3.9.1.3. excel_Exists = Boolean, This

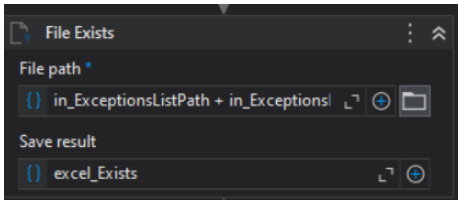
3.9.1.4. folder_Exists = Bollean, This

3.9.1.5. exception_Message = String, This, “ unexpected exception”

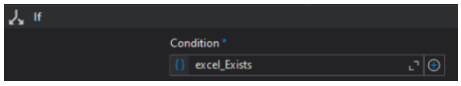
3.9.2. <Assign> = timeStamp_Message = System.DateTime.Now.ToString("HH.mm.ss ")



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

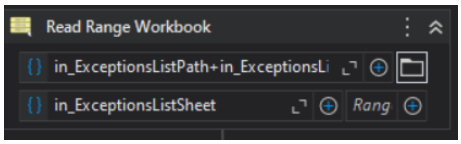


3.9.4. IF excel_Exists THEN

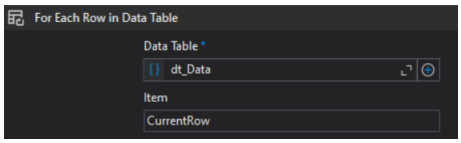


3.9.4.1. <Read Range Workbook> = in_ExceptionsListPath + in_ExceptionsListFile, in_ExceptionsListSheet

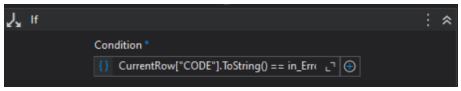
3.9.4.1.1. DataTable = dt_Data



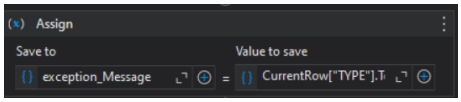
3.9.4.2. <For Each Row in Data Table> = dt_Data, CurrentRow



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



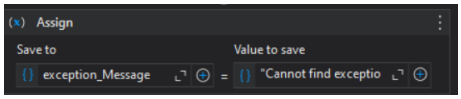
3.9.4.2.1.1. <Assign> exception_Message = CurrentRow["MESSAGE"].ToString()



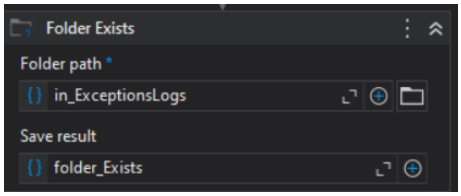
3.9.4.2.1.2. <Break>

3.9.5. ELSE

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

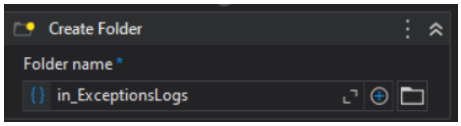


3.9.6. <Folder Exists> = in_ExceptionsLogs, folder_Exists



3.9.7. IF !folder_Exists THEN

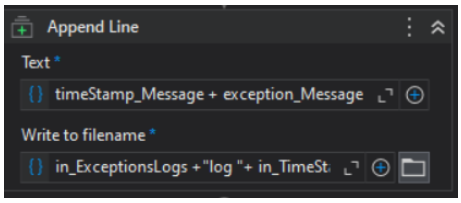
3.9.7.1. <Create Folder> = in_ExceptionsLogs



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	Performer
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
MaxAmount	500000

4.2. Global Variables

No variables have been defined.

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 CFV_Exceptions, includes de 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
B04	Cannot find the Excel file to save the values
B05	The amount is below 500.000€
B06	No EUR invoices has been found

5.2. System Exceptions

Code	Description
S01	Cannot open the ACME webpage
S02	Cannot load the values to the Orchestrator Queue

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
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.