

# AA Enterprise Framework Sample Illustration-1

## Problem statement

Create a bot to get the temperature from Google for a list of cities stored in a csv file and write it to another csv file. The bot should recover automatically (atmost 2 times) in case of exceptions and log all actions and exceptions.

Steps:

1. Deploy framework in the respective folders in the Automation Anywhere Application path. Do retain the folder structure.
2. Populate/Add the necessary key-values in the Config.xlsx file.
  - a. Set the value for **MaxRetryCount** as 3.
  - b. Set the value for **TransactionItemsPath** to the path of the csv file containing the list of cities.
    - i. Set in this case as *C:\Users\U43360\Documents\Automation Anywhere Files\Automation Anywhere\Data\EnterpriseFramework Sample-1\ListOfTransactionItems.csv*
3. Open the workflow file **EnterpriseFramework** for edit in Automation Anywhere Client.
4. Open the task **Initialize** and do the following:
  - a. Kill/Close all applications used in the automation use case.
    - i. For this, Open the task **CloseAllApplications** and do the following:

```
1  Comment: *****CloseAllApplications*****
2  Begin Error Handling; Action: Stop Task; Options: Log to File, Task Status: Fail
3  Log to File: Info: Closing all applications in "$AAApplicationPath$\LogFiles\EnterpriseFramework Sample-1\EnterpriseFrameworkLog.txt"
4  IF If Window Exists ("weather in") Then (Wait up to 2 seconds -for Window to exist)
5      Close Window: "weather in"
6  End If
7  IF If Window Exists ("OutputTransactionProcessing") Then (Wait up to 2 seconds -for Window to exist)
8      Close Window: "OutputTransactionProcessing"
9  End If
10 End Error Handling
11 Comment: *****End of CloseAllApplications*****
```

- i. For this, Open the task **OpenAllApplications** and do the following:

```
1  Comment: *****OpenAllApplications*****
2  Begin Error Handling; Action: Stop Task; Options: Log to File, Task Status: Fail
3  Log to File: Info: Opening all applications in "$AAApplicationPath$\LogFiles\EnterpriseFramework Sample-1\EnterpriseFrameworkLog.txt"
4  Launch Website "https://www.google.com/search?safe=active&ei=OShXJKZEFDAz7sPq5C6qAQ&q=weather+in+city"
5  Maximize Window: "weather in"
6  End Error Handling
7  Comment: *****End of OpenAllApplications*****
```

5. Open the task **GetTransaction** and do the following:
  - a. Here, the transaction item is the city for which temperature should be retrieved for. The transaction id is the row index of the city in the csv file. The result is assigned to *vOutTransactionItem* as shown below. The status of the transaction is set to *Completed* when no item is found at the row whose index corresponds to the transaction id. This is expected when all transaction items are processed.

```

1  Comment: *****Get Transaction: Retrieve transaction item*****
2  Comment: Expected input- vInTransactionId: Transaction id of the transaction item to be retrieved
3  Comment: Expected input- vInConfigXML: XML Config string
4  Comment: Returned output- vOutTransactionItem: Transaction item corresponding to the transaction id that is then passed to Process transaction for further processing
5  Comment: Returned output- vOutTransactionStatus: Status of the transaction. Will be 'In Progress' by default. Value will be changed to 'Completed' upon processing of all transactions
6  Begin Error Handling; Action: Stop Task; Options: Log to File, Task Status: Fail
7  Log to File: Info: Executing GetTransaction in "$AAApplicationPath$\LogFiles\EnterpriseFrameworkLog.txt"
8  Comment: Example: Config can be accessed like below whenever required
9  Comment: XPath expression: /Data/[KeyName] (or /Data/[SheetName]/[KeyName]) if XML created with sheet specific nodes
10 XML : Start XML Session using text <...> Session:Config
11 XML : Get Single Node value from XPath location "/Data/TransactionItemsPath" and assign to variable "$vTransactionItemSourcePath$". Session:Config
12 XML : End XML Session. Session:Config
13 Comment: Logic
14 Variable Operation: "" To $vOutTransactionItem$
15 Variable Operation: In Progress To $vOutTransactionStatus$
16 If $vTransactionItemSourcePath$ Not Equal To (<)> "" Then
17   Read From CSV file: "$vTransactionItemSourcePath$" Delimiter: "Comma" Header: "Yes" Trim Leading Space: "Yes" Trim Trailing Space: "Yes" Session: 'GetTransaction'
18   Start Loop "Each row in a CSV/Text file of Session: Get Transaction"
19     If $Counter$ Equal To (+) $vInTransactionId$ Then
20       Variable Operation: $Filedata Column(1)$ To $vOutTransactionItem$
21     Exit Loop
22   End If
23 End Loop
24 End If
25 If $vOutTransactionItem$ Not Equal To (<)> "" Then
26   Log to File: Info: GetTransaction finds the TransactionItem $Filedata Column(1)$ corresponding to the TransactionId $vInTransactionId$ in "$AAApplicationPath$\LogFiles\EnterpriseFramework Sample-1\EnterpriseFrameworkLog.txt"
27 Else
28   Log to File: Info: GetTransaction has processed all transaction items in "$AAApplicationPath$\LogFiles\EnterpriseFramework Sample-1\EnterpriseFrameworkLog.txt"
29   Variable Operation: Completed To $vOutTransactionStatus$
30 End If
31 Log to File: Info: GetTransaction returns TransactionItem: $vOutTransactionItem$ and TransactionStatus: $vOutTransactionStatus$ in "$AAApplicationPath$\LogFiles\EnterpriseFramework Sample-1\EnterpriseFrameworkLog.txt"
32 End Error Handling
33 Comment: *****End of Get Transaction*****

```

6. Open the task **ProcessTransaction** and do the following:
  - a. Add commands to retrieve temperature from Google for the city retrieved in step #5 (transaction item) and write it to the corresponding csv file as shown below:

```

1  Comment: *****Process Transaction: Add business logic to perform necessary actions with the transaction item received as input*****
2  Comment: Expected input- vInTransactionItem: Transaction item to be processed
3  Comment: Expected input- vInConfigXML: XML Config string
4  Begin Error Handling; Action: Stop Task; Options: Log to File, Variable Assignment, Task Status: Fail
5  Log to File: Info: Executing ProcessTransaction for transaction item $vInTransactionItem$ in "$AAApplicationPath$\LogFiles\EnterpriseFramework Sample-1\EnterpriseFrameworkLog.txt"
6  Variable Operation: False To $vOutErrorInProcessTransaction$
7  Comment: Example: Config can be accessed like below whenever required
8  Comment: XPath expression: /Data/[KeyName] (or /Data/[SheetName]/[KeyName]) if XML created with sheet specific nodes
9  XML : Start XML Session using text <...> Session:Config
10 XML : End XML Session. Session:Config
11 Comment: Logic
12 Object Cloning: Set Text of TextBox "q" in window 'weather in "; Value: "weather in $vInTransactionItem$[ENTER]"; Source: Window; Play Type: Object
13 Object Cloning: Get Property 'HTML InnerText' of 'StaticText' "" from window 'weather in "; Assign to variable "$Prompt-Assignment$"; Source: Window; Play Type: Object
14 Excel: Open Spreadsheet "C:\Users\U43360\Documents\Automation Anywhere Files\Automation Anywhere\EnterpriseFramework Sample-1\OutputTransactionProcessing.csv". ActiveSheet: "Default". Contains Header. Session: WriteResult
15 Excel: Go to beginning of the row of the active cell. Session: WriteResult
16 Excel: Go to end of the column of the active cell. Session: WriteResult
17 Excel: Go to one cell below the active cell. Session: WriteResult
18 Excel: Set value of Active Cell with "$vInTransactionItem$". Session: WriteResult
19 Excel: Go to one cell right of active cell. Session: WriteResult
20 Excel: Set value of Active Cell with "$Prompt-Assignment$". Session: WriteResult
21 Excel: Save Spreadsheet. Session: WriteResult
22 Log to File: Info: Completed ProcessTransaction for transaction item $vInTransactionItem$ in "$AAApplicationPath$\LogFiles\EnterpriseFramework Sample-1\EnterpriseFrameworkLog.txt"
23 End Error Handling
24 Comment: *****End of Process Transaction*****

```

7. The task **UpdateTransactionId** will automatically run to iterate to the next transaction.
8. The task **SetRetryCount** will automatically initiate recovery mechanism.



**Solution:**

AA Enterprise Framework Sample Illustration - 1.zip