AppPerfect

ExpensifyBot-AppPerfect Corporation -Readme

Version 1.0

August 27,2019

Table of Contents

1. Introduction

This document contains all essential information for the user to make full use of the Bot or Digital worker. This manual includes a description of the functions and capabilities and step-by-step procedures for setup & configuration of the Bot.

Overview

This Bot reads all the expenses from a CSV Template and them in the employee account. It creates a report in the given policy where all the expenses have been created from a CSV file. It exports the report to the recipient mails read from a CSV Template based on Filter Conditions & other configurable parameters. It updates the status of report from closed or approved to reimbursed where all report Id are read from a CSV Template.

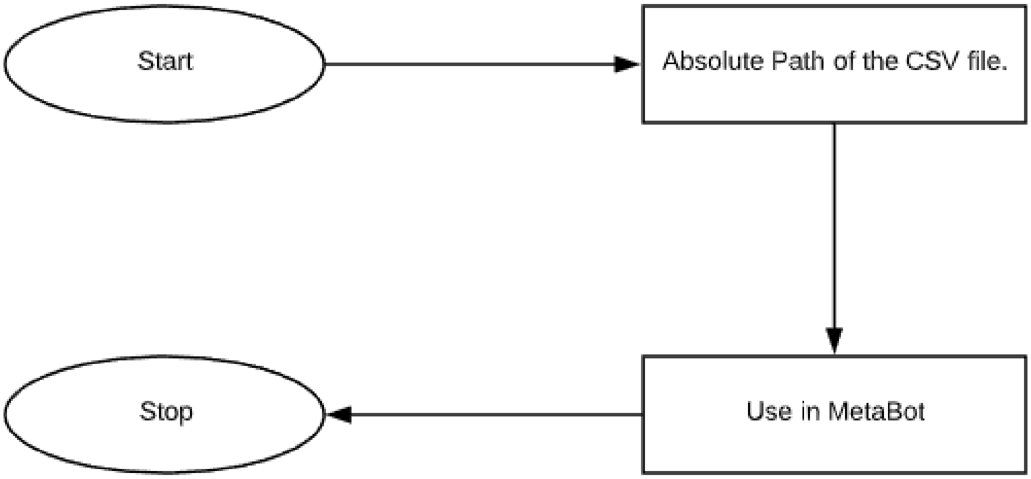
Detailed steps are as follows:

Create Expense

1) The user provides the path of expense file where all expenses have been present. The expense file has both optional and required parameters.

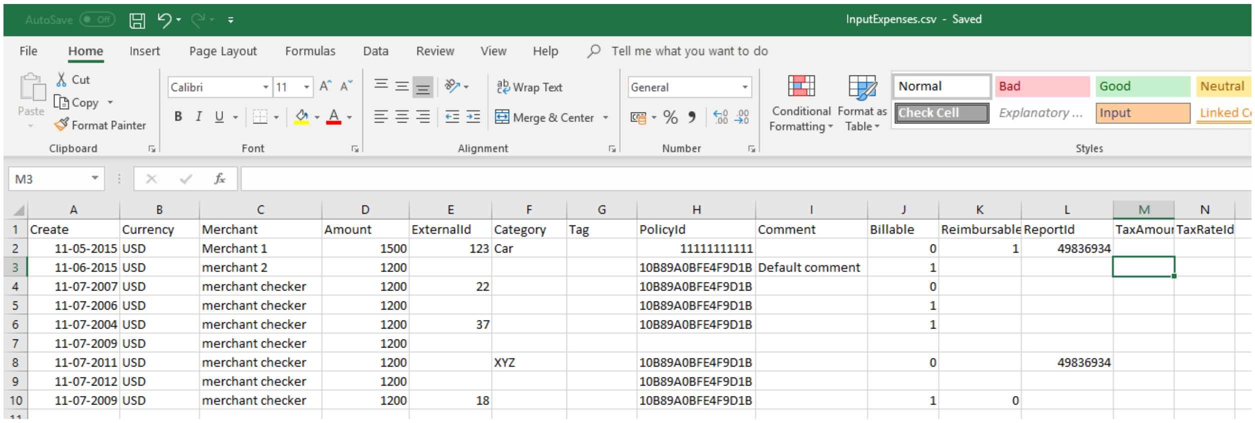
- Required Parameters in the Expense file.
- Optional Parameters in the Expense file.

- **Visual Flow –**



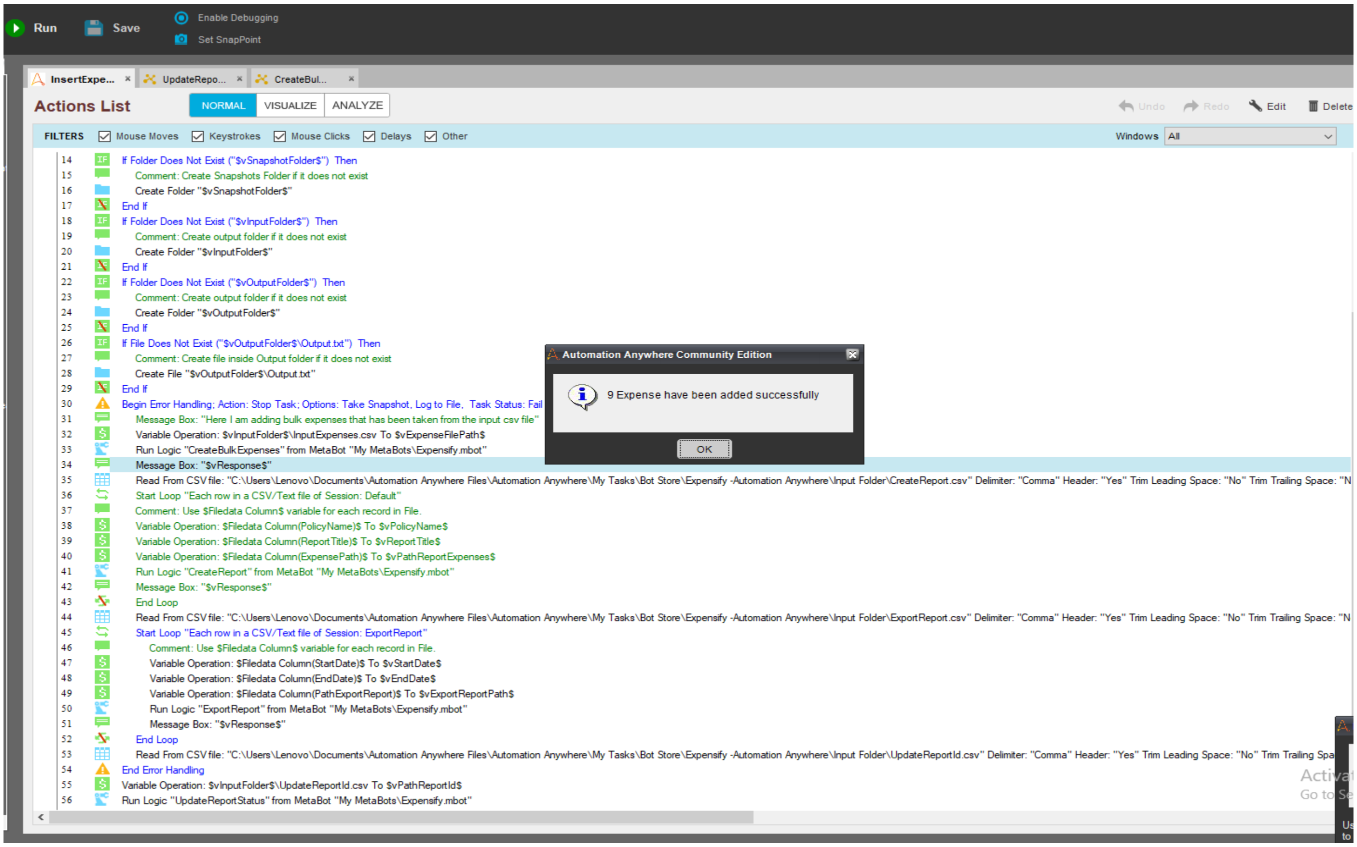
- **Input and Output Example –**

Input: Create Expenses csv



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Create	Currency	Merchant	Amount	ExternalId	Category	Tag	PolicyId	Comment	Billable	Reimbursable	ReportId	TaxAmour	TaxRateId
2	11-05-2015	USD	Merchant 1	1500	123	Car		11111111111		0	1	49836934		
3	11-06-2015	USD	merchant 2	1200				10B89A0BFE4F9D1B	Default comment	1				
4	11-07-2007	USD	merchant checker	1200	22			10B89A0BFE4F9D1B		0				
5	11-07-2006	USD	merchant checker	1200				10B89A0BFE4F9D1B		1				
6	11-07-2004	USD	merchant checker	1200	37			10B89A0BFE4F9D1B		1				
7	11-07-2009	USD	merchant checker	1200										
8	11-07-2011	USD	merchant checker	1200		XYZ		10B89A0BFE4F9D1B		0		49836934		
9	11-07-2012	USD	merchant checker	1200				10B89A0BFE4F9D1B						
10	11-07-2009	USD	merchant checker	1200	18			10B89A0BFE4F9D1B		1	0			

Output: Message box:



9 Expense have been added successfully

```
If Folder Does Not Exist ("$vSnapshotFolder$")  Then
    Comment: Create Snapshots Folder if it does not exist
    Create Folder "$vSnapshotFolder$"
End If
If Folder Does Not Exist ("$vInputFolder$")  Then
    Comment: Create output folder if it does not exist
    Create Folder "$vInputFolder$"
End If
If Folder Does Not Exist ("$vOutputFolder$")  Then
    Comment: Create output folder if it does not exist
    Create Folder "$vOutputFolder$"
End If
If File Does Not Exist ("$vOutputFolder$\Output.txt")  Then
    Comment: Create file inside Output folder if it does not exist
    Create File "$vOutputFolder$\Output.txt"
End If
Begin Error Handling; Action: Stop Task; Options: Take Snapshot, Log to File,  Task Status: Fail
    Message Box: "Here I am adding bulk expenses that has been taken from the input csv file"
    Variable Operation: $vInputFolder$\InputExpenses.csv To $vExpenseFilePath$
    Run Logic "CreateBulkExpenses" from MetaBot "My MetaBots\Expensify.mbot"
    Message Box: "$vResponse$"
    Read From CSV file: "C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\CreateReport.csv" Delimiter: "Comma" Header: "Yes" Trim Leading Space: "No" Trim Trailing Space: "N
    Start Loop "Each row in a CSV/Text file of Session: Default"
    Comment: Use $Filedata Column$ variable for each record in File.
    Variable Operation: $Filedata Column(PolicyName)$ To $vPolicyName$
    Variable Operation: $Filedata Column(ReportTitle)$ To $vReportTitle$
    Variable Operation: $Filedata Column(ExpensePath)$ To $vPathReportExpenses$
    Run Logic "CreateReport" from MetaBot "My MetaBots\Expensify.mbot"
    Message Box: "$vResponse$"
    End Loop
    Read From CSV file: "C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\ExportReport.csv" Delimiter: "Comma" Header: "Yes" Trim Leading Space: "No" Trim Trailing Space: "N
    Start Loop "Each row in a CSV/Text file of Session: ExportReport"
        Comment: Use $Filedata Column$ variable for each record in File.
        Variable Operation: $Filedata Column(StartDate)$ To $vStartDate$
        Variable Operation: $Filedata Column(EndDate)$ To $vEndDate$
        Variable Operation: $Filedata Column(PathExportReport)$ To $vExportReportPath$
        Run Logic "ExportReport" from MetaBot "My MetaBots\Expensify.mbot"
        Message Box: "$vResponse$"
    End Loop
    Read From CSV file: "C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\UpdateReportId.csv" Delimiter: "Comma" Header: "Yes" Trim Leading Space: "No" Trim Trailing Spa
End Error Handling
Variable Operation: $vInputFolder$\UpdateReportId.csv To $vPathReportId$
Run Logic "UpdateReportStatus" from MetaBot "My MetaBots\Expensify.mbot"
```

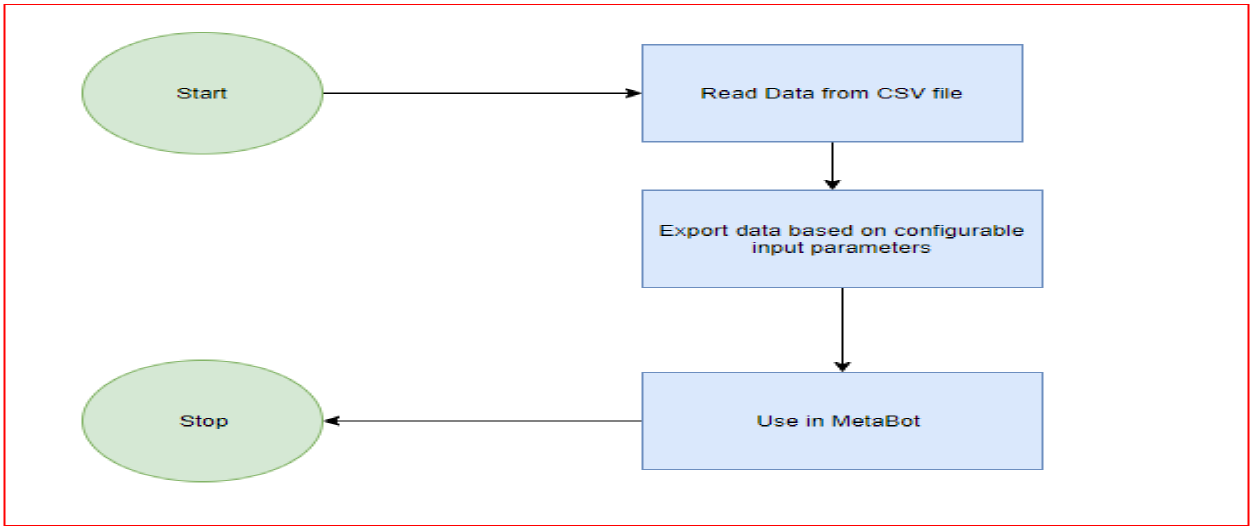
Create Report/Export Report

1) Read data from a CSV Template, e.g. CreateReport.csv
2) Export data from the CSV Template based on Filter conditions set by a user through a Configuration File.

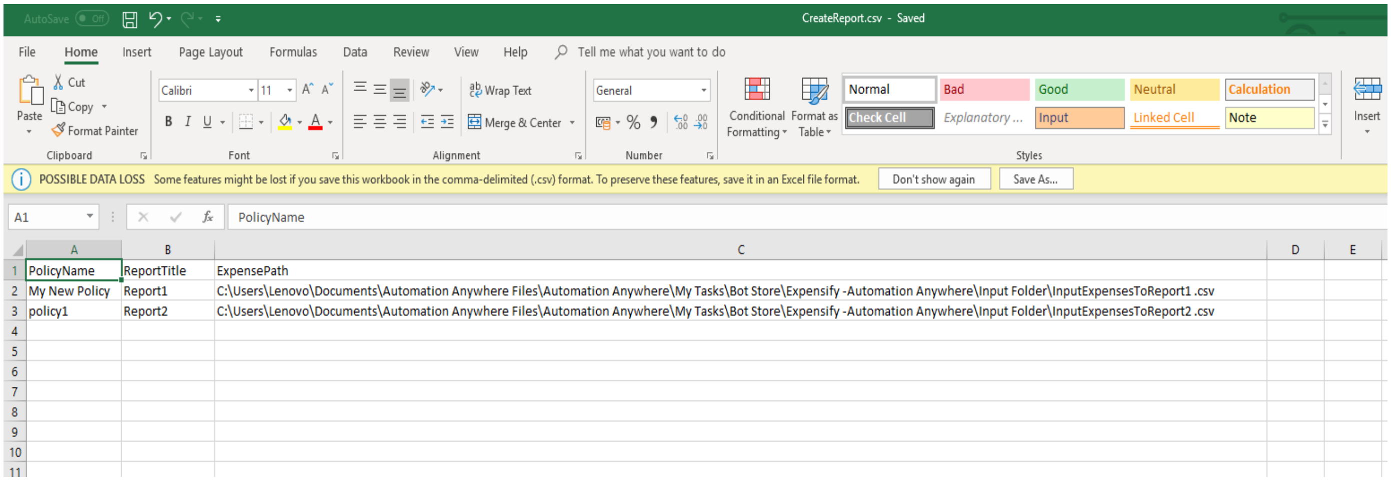
   Filter conditions are as below (Click individual to read details)

   - **Filter based on Column Value**
3) In the Export Report, the generated CSV file sends to the recipient Email.

Flow Diagram



Example Create Report

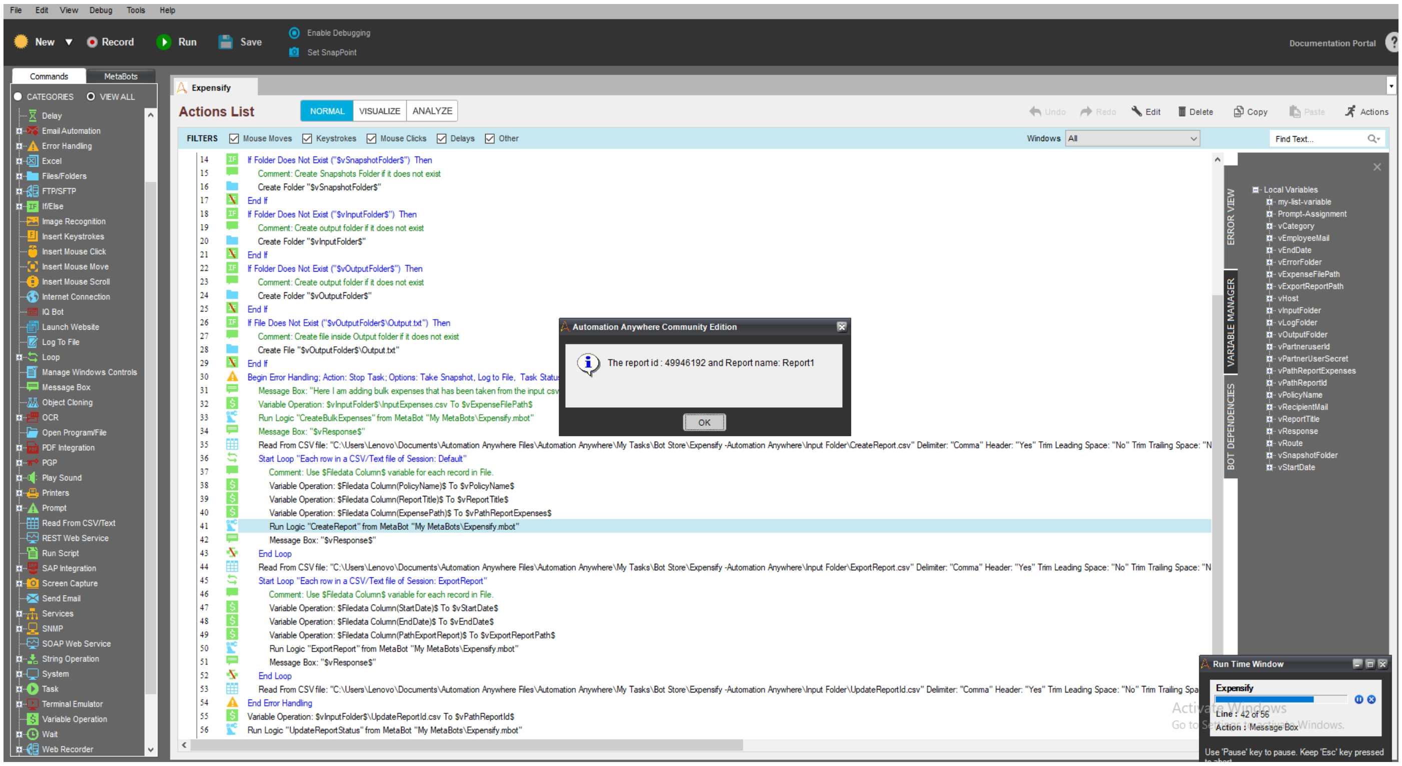


POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't show again Save As...

A1 PolicyName

	A	B	C	D	E
1	PolicyName	ReportTitle	ExpensePath		
2	My New Policy	Report1	C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\InputExpensesToReport1 .csv		
3	policy1	Report2	C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\InputExpensesToReport2 .csv		
4					
5					
6					
7					
8					
9					
10					
11					

Output: Message Box


Expensify
Actions List
NORMAL
VISUALIZE
ANALYZE
FILTERS ☑ Mouse Moves ☑ Keystrokes ☑ Mouse Clicks ☑ Delays ☑ Other
Automation Anywhere Community Edition
OK
Run Time Window

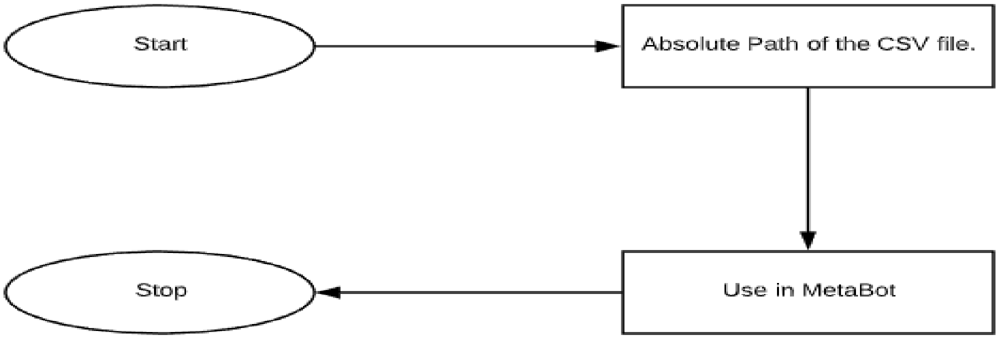
```
If Folder Does Not Exist ("$vSnapshotFolder$")  Then
    Comment: Create Snapshots Folder if it does not exist
    Create Folder "$vSnapshotFolder$"
End If
If Folder Does Not Exist ("$vInputFolder$")  Then
    Comment: Create output folder if it does not exist
    Create Folder "$vInputFolder$"
End If
If Folder Does Not Exist ("$vOutputFolder$")  Then
    Comment: Create output folder if it does not exist
    Create Folder "$vOutputFolder$"
End If
If File Does Not Exist ("$vOutputFolder$\Output.txt")  Then
    Comment: Create file inside Output folder if it does not exist
    Create File "$vOutputFolder$\Output.txt"
End If
Begin Error Handling; Action: Stop Task; Options: Take Snapshot, Log to File,  Task Statu
    Message Box: "Here I am adding bulk expenses that has been taken from the input csv
    Variable Operation: $vInputFolder$\InputExpenses.csv To $vExpenseFilePath$
    Run Logic "CreateBulkExpenses" from MetaBot "My MetaBots\Expensify.mbot"
    Message Box: "$vResponse$"
    Read From CSV file: "C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\CreateReport.csv" Delimiter: "Comma" Header: "Yes" Trim Leading Space: "No" Trim Trailing Space: "N
    Start Loop "Each row in a CSV/Text file of Session: Default"
        Comment: Use $Filedata Column$ variable for each record in File.
        Variable Operation: $Filedata Column(PolicyName)$ To $vPolicyName$
        Variable Operation: $Filedata Column(ReportTitle)$ To $vReportTitle$
        Variable Operation: $Filedata Column(ExpensePath)$ To $vPathReportExpenses$
        Run Logic "CreateReport" from MetaBot "My MetaBots\Expensify.mbot"
        Message Box: "$vResponse$"
    End Loop
    Read From CSV file: "C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\ExportReport.csv" Delimiter: "Comma" Header: "Yes" Trim Leading Space: "No" Trim Trailing Space: "N
    Start Loop "Each row in a CSV/Text file of Session: ExportReport"
        Comment: Use $Filedata Column$ variable for each record in File.
        Variable Operation: $Filedata Column(StartDate)$ To $vStartDate$
        Variable Operation: $Filedata Column(EndDate)$ To $vEndDate$
        Variable Operation: $Filedata Column(PathExportReport)$ To $vExportReportPath$
        Run Logic "ExportReport" from MetaBot "My MetaBots\Expensify.mbot"
        Message Box: "$vResponse$"
    End Loop
    Read From CSV file: "C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\UpdateReportId.csv" Delimiter: "Comma" Header: "Yes" Trim Leading Space: "No" Trim Trailing Spa
End Error Handling
Variable Operation: $vInputFolder$\UpdateReportId.csv To $vPathReportId$
Run Logic "UpdateReportStatus" from MetaBot "My MetaBots\Expensify.mbot"
```

The report id : 49946192 and Report name: Report1

Update ReportStatus

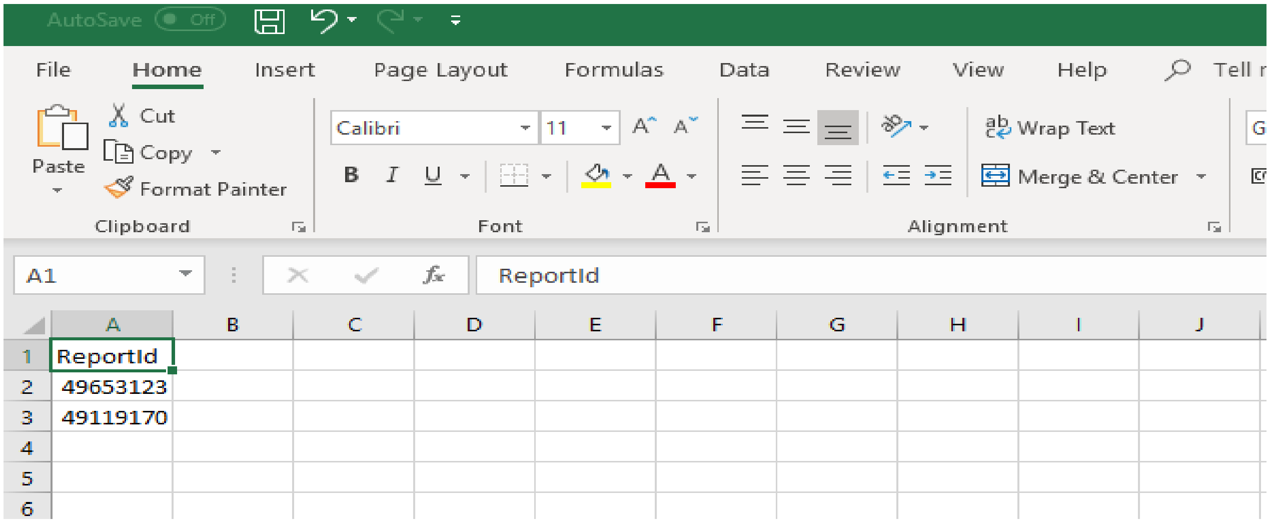
1) User provides the absolute path of ReportId.csv file where all report Id are present. ReportId.csv file has only one column where all report Ids are present.

- <u>**Visual Flow –**</u>



- **Input and Output Example –**

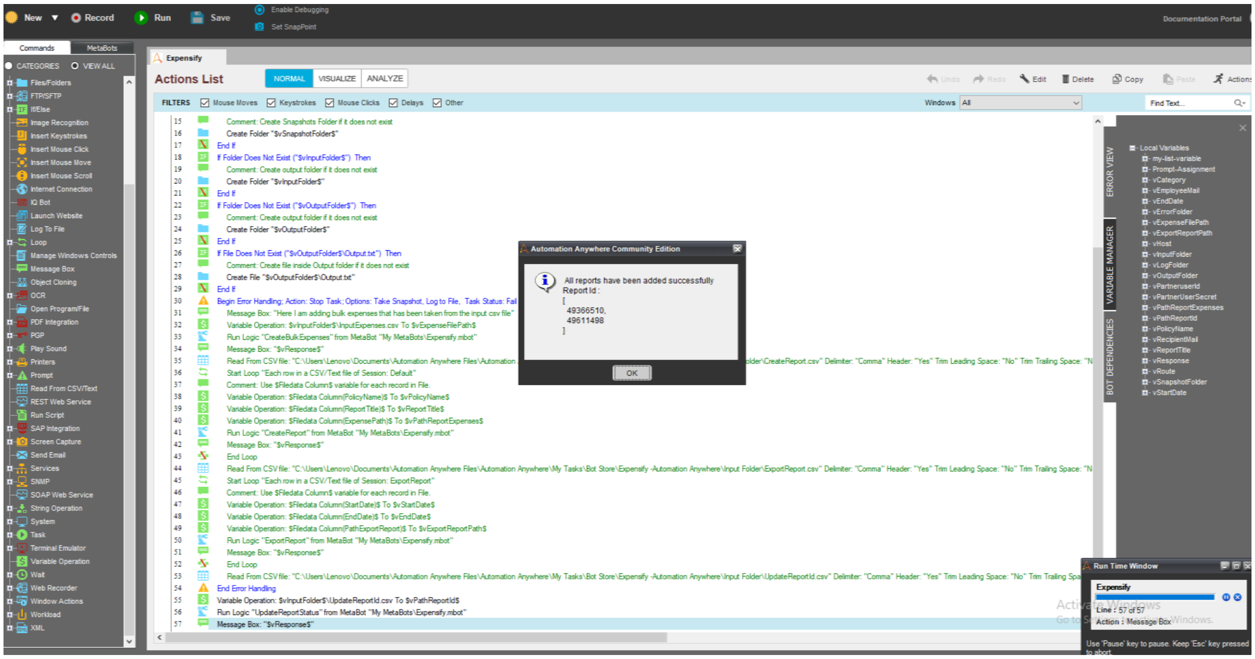
Example- ReportId.csv



A1 | ReportId

	A	B	C	D	E	F	G	H	I	J
1	ReportId									
2	49653123									
3	49119170									
4										
5										
6										

Output: Message Box



```
    Comment: Create Snapshots Folder if it does not exist
    Create Folder "$vSnapshotFolder$"
End If
If Folder Does Not Exist ("$vInputFolder$") Then
    Comment: Create output folder if it does not exist
    Create Folder "$vInputFolder$"
End If
If Folder Does Not Exist ("$vOutputFolder$") Then
    Comment: Create output folder if it does not exist
    Create Folder "$vOutputFolder$"
End If
If File Does Not Exist ("$vOutputFolder$\Output.txt") Then
    Comment: Create file inside Output folder if it does not exist
    Create File "$vOutputFolder$\Output.txt"
End If
Begin Error Handling; Action: Stop Task; Options: Take Snapshot, Log to File,  Task Status: Fail
    Message Box: "Here I am adding bulk expenses that has been taken from the input csv file"
    Variable Operation: $vInputFolder$\InputExpenses.csv To $vExpenseFilePath$
    Run Logic "CreateBulkExpenses" from MetaBot "My MetaBots\Expensify.mbot"
    Message Box: "$vResponse$"
    Read From CSV file: "C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation older\CreateReport.csv" Delimiter: "Comma" Header: "Yes" Trim Leading Space: "No" Trim Trailing Space: "N
    Start Loop "Each row in a CSV/Text file of Session: Default"
    Comment: Use $Filedata Column$ variable for each record in File.
    Variable Operation: $Filedata Column(PolicyName)$ To $vPolicyName$
    Variable Operation: $Filedata Column(ReportTitle)$ To $vReportTitle$
    Variable Operation: $Filedata Column(ExpensePath)$ To $vPathReportExpenses$
    Run Logic "CreateReport" from MetaBot "My MetaBots\Expensify.mbot"
    Message Box: "$vResponse$"
    End Loop
    Read From CSV file: "C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\ExportReport.csv" Delimiter: "Comma" Header: "Yes" Trim Leading Space: "No" Trim Trailing Space: "N
    Start Loop "Each row in a CSV/Text file of Session: ExportReport"
    Comment: Use $Filedata Column$ variable for each record in File.
    Variable Operation: $Filedata Column(StartDate)$ To $vStartDate$
    Variable Operation: $Filedata Column(EndDate)$ To $vEndDate$
    Variable Operation: $Filedata Column(PathExportReport)$ To $vExportReportPath$
    Run Logic "ExportReport" from MetaBot "My MetaBots\Expensify.mbot"
    Message Box: "$vResponse$"
    End Loop
    Read From CSV file: "C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\UpdateReportId.csv" Delimiter: "Comma" Header: "Yes" Trim Leading Space: "No" Trim Trailing Spa
End Error Handling
Variable Operation: $vInputFolder$\UpdateReportId.csv To $vPathReportId$
Run Logic "UpdateReportStatus" from MetaBot "My MetaBots\Expensify.mbot"
Message Box: "$vResponse$"
```

Automation Anywhere Community Edition

```
All reports have been added successfully
Report Id :
[
  49366510,
  49611498
]
```

OK

Common Use cases

The fields in the CSV can be changed to whatever data that needs to be added in the list

- Add Bulk expenses in user Expensify account
- Create Report in the user Expensify account
- Export report to the recipient mails.
- Update Report status from closed/approved to reimbursed in the user Expensify account

2. Requirements & Prerequisites

System Requirements

For the PC or server where the bot needs to run:

- ✓ RAM: 8GB or higher
- ✓ PROCESSOR: Intel Core i5 or higher and equivalent for any other OS
- ✓ Hard Disk: Up to 2GB of overall free space in the AA Client installation drive.

Reference below for Enterprise Client & Control Room system requirements.

Prerequisites

- ✓ ***Software's needed -***
- ✓ AA Enterprise Client 11.3.3
- ✓ AA Enterprise Control Room 11.3.3
- **Accounts/License needed –**
- ✓ Expensify Account. Expensify account can be opened at here
- ✓ AA Enterprise License

Security Measures

There are some security best practice recommendations that you may follow with your bot.

- It is not recommended to provide admin access to the Windows User Account executing the Bots, to avoid unintended data changes with the CSV Template.
- It is recommended to update Expensify API Token twice or more in a quarter to ensure data safety as well as update the credential vault with updated credential.

  Note: For regenerating PartnerUserSecret , Click here

3. Getting Started

Skill Matrix

The functionality of the Bot has been divided into a set of skills.

Below is an overview of how the task bots and meta bots map to these skills:

Skill	Task Files	MetaBot Files
Create Expenses, Create Report, Export Report, Update the status of a report from closed/approved to reimbursed	Expensify.atmx	Expensify.mbot

Installation Hierarchy

Once the bot is downloaded and installed, the installer creates the files in the folder structure as shown below.

Folder Structure on your Machine

Folder Structure	Description
<AA Application Path>	**<AA Application Path>** is the location where AA files are stored on your machine
1. My Tasks	**1. My Tasks** My Task Folder is the default directory where Bot Files are saved.
1.1. Bot Store	**1.1 Bot Store** Bot Store Folder contains the Bot Name Folder which the installer creates while installation of the Bot.
1.1.1. ExpensifyBot-AppPerfectCorporation	**ExpensifyBot- AppPerfectCorporation** This folder gets created by the installer and contains files and folders needed for the bot execution.
➢ **Error Folder** • **Logs** ○ Error logs Month-Day-Year.txt • **Snapshots** ○ Error Snap Month-Day-Year Hour Min Sec.png	➢ **Error folder** is where logs and snapshots of screens will be placed if something goes wrong with the bot during execution
➢ **Input Folder** • CreateReport.csv • ExportReport.csv • InputExpenses.csv • UpdateReportId.csv ➢ **My Tasks** • Expensify.atmx **2. My MetaBots** ➢ Expensify.mbot	➢ **Input Folder** is where the input files that the bot needs for execution of the use case is saved • **CreateReport.csv** provides policy name, Report Title and absolute file path where all expenses are present as an input to the bot for Creating Report. • **ExportReport.csv** provides Start Date, end Date and absolute file path where all the required parameters are used in exporting report is present as an input to the bot for exporting Report. • **UpdateReportId.csv** provides Report Ids whose status needs to be updated from closed/approved state to reimbursed. • **InputExpenses.csv** provides all the parameters which are given to create Expenses in the user account.

	➢ **My Tasks** folder contains all the Developed Platform Source Code ➢ **My MetaBots** folder contains the Developed Metabots need for the bot execution.

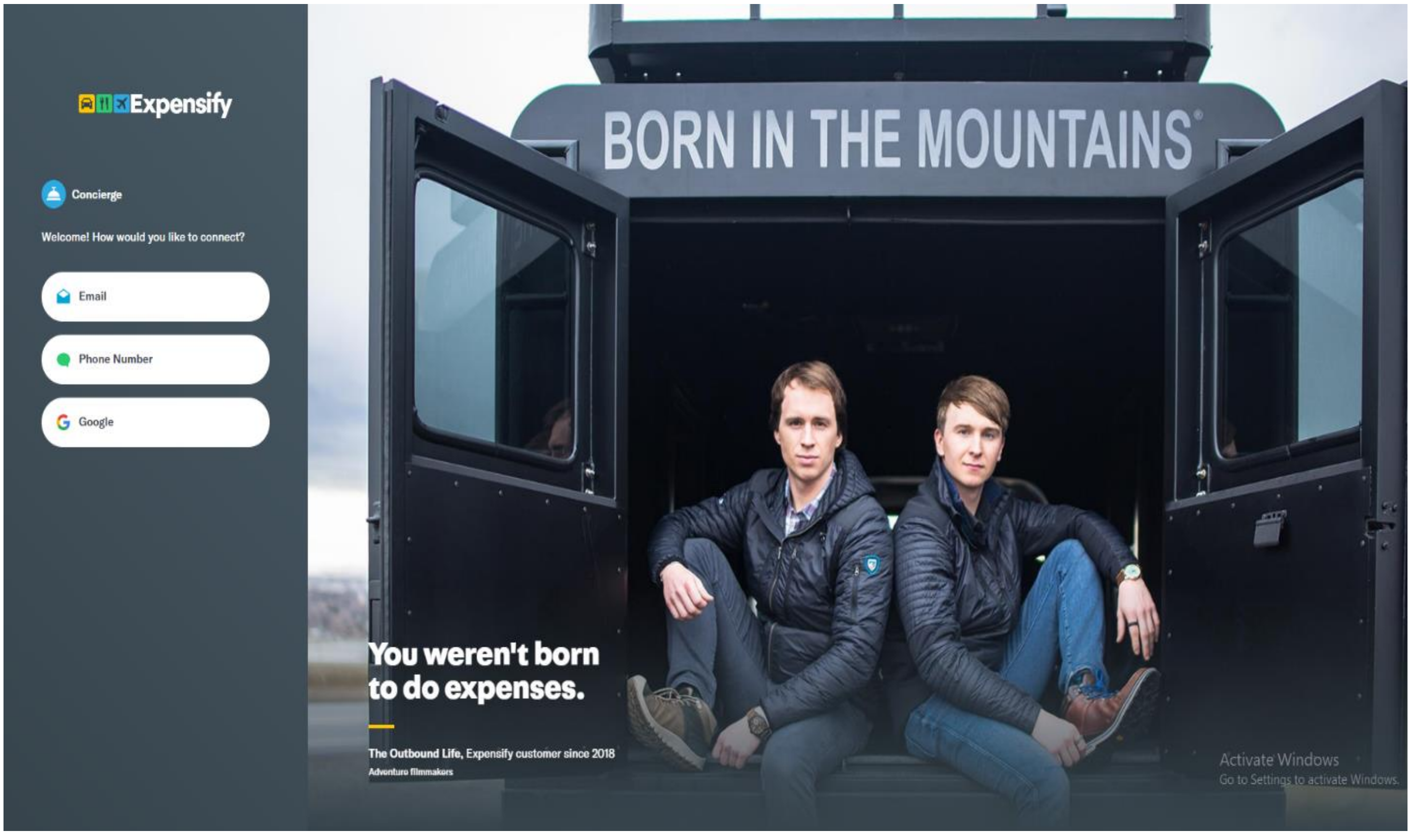
Quick Start

3.1.1 Setup

STEP 1 (Setting up an Expensify Account):

Click here

The below page will appear once the above link is clicked -


Expensify
Concierge
Welcome! How would you like to connect?
Email
Phone Number
Google
BORN IN THE MOUNTAINS
You weren't born to do expenses.
The Outbound Life, Expensify customer since 2018
Adventure filmmakers
Activate Windows
Go to Settings to activate Windows.

- If you already have an account, click on ‘**Log in**’.
- For new users, click on Email and enter require details and verify your account from email.

Step 2 (Generating URL & Keys):

Find or Generate Your API Key

- Click here to navigate to the expensify website. If you are not logged in log-in with your user credentials.
- After logged in you will find three values

Server URL

partnerUserID

partnerUserSecret

Save all these values in your computer and do not share your partnerUserId and partnerUserSecret with anyone.

3.1.2 Inputting your Access Code

Step by Step Guide for Customers: Inputting Your Access Code **click here**

3.1.3 Configuration

STEP 1 (Configuring Credential in Control Room’s Credential Vault)

Passwords and other sensitive information such as partnerUserId and partnerUserSecret included in automation tasks, should be stored as credentials centrally in the Credential Vault.

These sensitive data items become secured (stored in CR) and can’t be accessed locally which is used by the bots when it executes in the machine.

Below are the step-by-step instructions for what needs to be done to set up the URL and Token for Expensify in the Credential Vault.

- ‘Locker Consumer Role’ needs to be assigned to the AA user (bot runner) which ensures the user has full access to the locker.
- Locker needs to be created with name as “Expensify_Locker”
- Credential needs to be created with name as “Expensify _Credential”, where the URL and Token needs to be saved.

Locker Name	Credentials Name	Attribute Name	Value

Expensify_Locker	Expensify _Credential	Host	Add Server URL that has been generated from the above links
Expensify_Locker	Expensify _Credential	Route	Add Endpoint resource /Integration-Server/ExpensifyIntegrations For more Information Click Here
Expensify_Locker	Expensify _Credential	partnerUserId	Add partnerUserID value which is generated using the above links
Expensify_Locker	Expensify _Credential	partnerUserSecret	Add partnerUserSecret value generated using the above links which are generated using the above links
Expensify_Locker	Expensify _Credential	employeeEmail	The email account which is used to create expensify account

For more information on how to set up credentials in the Control Room, please visit this link on the Automation Anywhere Product Documentation portal.

STEP 2 (Configuring Parameters with an External File)

External file can be configured at -

AA Application Path > My Tasks > Bot Store > ExpensifyBot-AppPerfectCorporation > Input Folder > CreateReport.csv

AA Application Path > My Tasks > Bot Store > ExpensifyBot-AppPerfectCorporation > Input Folder > ExportReport.csv

AA Application Path > My Tasks > Bot Store > ExpensifyBot-AppPerfectCorporation > Input Folder > InputExpenses.csv

AA Application Path > My Tasks > Bot Store > ExpensifyBot-AppPerfectCorporation > Input Folder > UpdateReportId.csv

This is where, the input values for variables are stored that are needed by the bot at the time of execution. (Sensitive variables – like user name, password, and URLs will be an input in the Credential).

Below is a table that summaries what the variable name is, what it does, and an example of the input-

CreateReport –

- **INPUT VARIABLES: NEEDS TO BE CONFIGURED BY THE USERS FOR INPUT PARAMETER**

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Input
vPolicyName	Text	Yes	Name of the policy in which a report is created.	policy1
vReportTitle	Text	Yes	The title of the report (Set by you)	Report_XYZ
vPathReportExpenses	Text	Yes	Absolute Path where Expense file has been present which are added in the report	C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\ ExpensifyBot-AppPerfectCorporation \Input Folder\InputExpensesToReport .csv

- **OUTPUT VARIABLES: DISPLAYS OUTPUT RESPONSE**

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Output
vResponse	Text	Yes	Displays Report ID and Report name of the generated report	The report id: 51141694 and Report name: Report1

CreateBulkExpenses –

- **INPUT VARIABLES: NEEDS TO BE CONFIGURED BY THE USERS FOR INPUT PARAMETER**

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Input
vExpenseFilePath	Text	Yes	Absolute file Path where Expense file is Present.	C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\ ExpensifyBot-AppPerfectCorporation \Input Folder\InputExpenses.csv

- **OUTPUT VARIABLES: DISPLAYS OUTPUT RESPONSE**

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Output
vResponse	Text	Yes	Displays No. of Expenses that has been updated successfully.	10 Expense have updated successfully.

ExportReport -

- **INPUT VARIABLES: NEEDS TO BE CONFIGURED BY THE USERS FOR INPUT PARAMETER**

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Input
vExportReportPath	Text	Yes	Absolute file Path where all Export Report parameters are present.	C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\ ExpensifyBot-AppPerfectCorporation \Input Folder\ExportReportParameters.csv
vStartDate	Text	No	Filters out all reports submitted or created before	2015-07-08

			the given date, whichever occurred last (yyyy-mm-dd)	
vEndDate	Text	No	Filters out all reports submitted or created after the given date, whichever occurred last (yyyy-mm-dd)	2019-06-05

- **OUTPUT VARIABLES: DISPLAYS OUTPUT RESPONSE**

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Output
vResponse	Text	Yes	Displays name of the generated report that has been send on the Recipient mail.	The generated file is myExportbf5ead4c-3848-4ea3-ab3c-1e9908a6503b.csv

UpdateReportStatus –

- **INPUT VARIABLES: NEEDS TO BE CONFIGURED BY THE USERS FOR INPUT PARAMETER**

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Input
vPathReportId	Text	Yes	Absolute file Path where all report ids are present in a CSV file.	C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\ ExpensifyBot-AppPerfectCorporation \Input Folder\UpdateReportId.csv

- **OUTPUT VARIABLES: DISPLAYS OUTPUT RESPONSE**

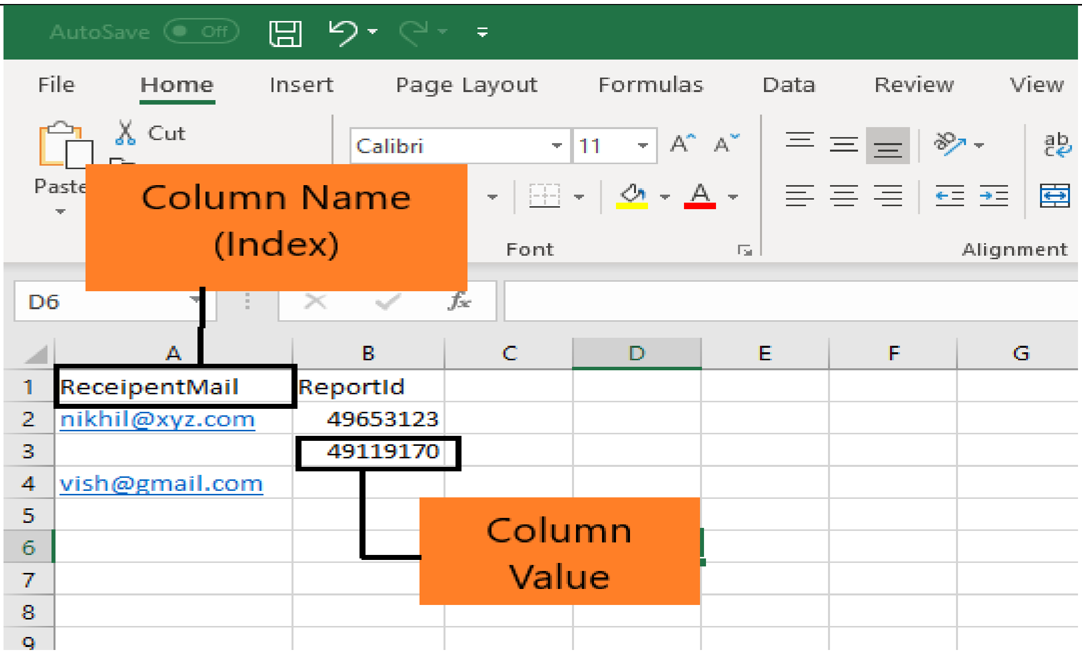
Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Output
vResponse	Text	Yes	Displays message about the report Id's that has been updated successfully	All reports have been updated successfully ReportId [45678901]

Guidance for configuration Filter conditions

a. Filter based on Column Value

Column Name is the column name that is present in the first row of the CSV file. Please do not delete or alter Column Name.
Column Value is the value that needs to be matched against the mentioned Column name. You can add or delete parameters as per your requirements.

Example



	A	B	C	D	E	F	G
1	ReceipentMail	ReportId					
2	nikhil@xyz.com	49653123					
3		49119170					
4	vish@gmail.com						
5							
6							
7							
8							
9							

Input CSV Files-

InputExpenses.csv

- **Required Parameters-**

File Parameter Name	Type	Description	Example Input
Create	Text	The date of the expense (format yyyy-mm-dd).	2015-07-08
Currency	Text	The three-letter currency code of the expense	USD
Merchant	Text	The name of the expense's merchant	Merchant try
Amount	Text	The amount of the expense, in cents	1500

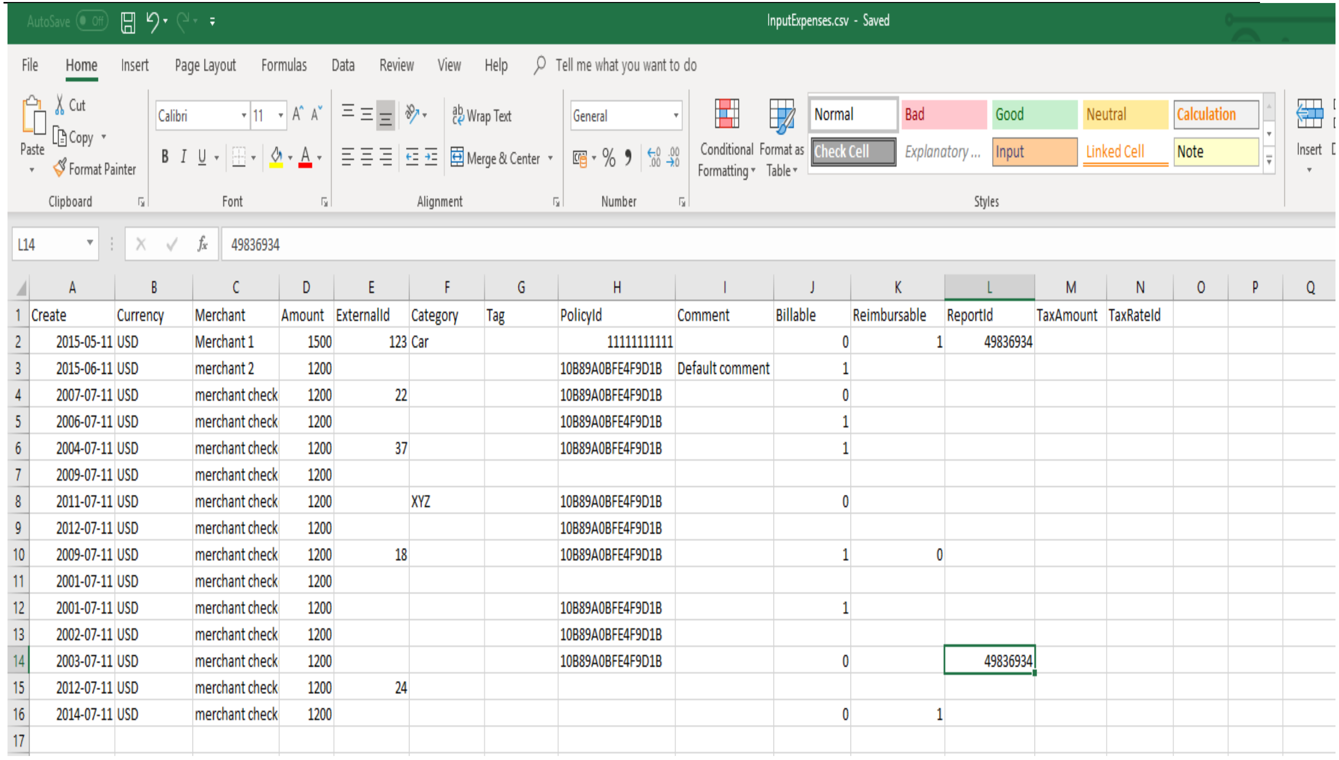
- **Optional Parameters -**

(If you don't want to add any value in the parameter leave it empty)

File Parameter Name	Type	Description	Example Input

ExternalId	Text	A unique, custom string that you specify - this will help identify the expense after being exported	12345
Category	Text	The name of the category to assign to the expense	Rent
Tag	Text	The name of the tag to assign to the expense	Tag Xyz
PolicyId	Text	The ID of the policy the tax belongs to	123456789
Comment	Text	An expense comment	Expense Comment
Billable	Text	Whether to mark the expense as billable or not **0 for false** **1 for true**	1
Reimbursable	Text	Whether to mark the expense as reimbursable or not **0 for false** **1 for true**	0
ReportId	Text	The ID of the report you want to attach the expense to	49700998
TaxAmount	Text	The tax amount paid on the expense	1500
TaxRateId	Text	The tax rateID, as defined in the policy.	5

Example-



L14 49836934

	Create	Currency	Merchant	Amount	ExternalId	Category	Tag	PolicyId	Comment	Billable	Reimbursable	ReportId	TaxAmount	TaxRateId
2	2015-05-11	USD	Merchant 1	1500	123	Car		11111111111		0	1	49836934		
3	2015-06-11	USD	merchant 2	1200				10B89A0BFE4F9D1B	Default comment	1				
4	2007-07-11	USD	merchant check	1200	22			10B89A0BFE4F9D1B		0				
5	2006-07-11	USD	merchant check	1200				10B89A0BFE4F9D1B		1				
6	2004-07-11	USD	merchant check	1200	37			10B89A0BFE4F9D1B		1				
7	2009-07-11	USD	merchant check	1200										
8	2011-07-11	USD	merchant check	1200		XYZ		10B89A0BFE4F9D1B		0				
9	2012-07-11	USD	merchant check	1200				10B89A0BFE4F9D1B						
10	2009-07-11	USD	merchant check	1200	18			10B89A0BFE4F9D1B		1	0			
11	2001-07-11	USD	merchant check	1200										
12	2001-07-11	USD	merchant check	1200				10B89A0BFE4F9D1B		1				
13	2002-07-11	USD	merchant check	1200				10B89A0BFE4F9D1B						
14	2003-07-11	USD	merchant check	1200				10B89A0BFE4F9D1B		0		49836934		
15	2012-07-11	USD	merchant check	1200	24									
16	2014-07-11	USD	merchant check	1200						0	1			

ExportReport.csv

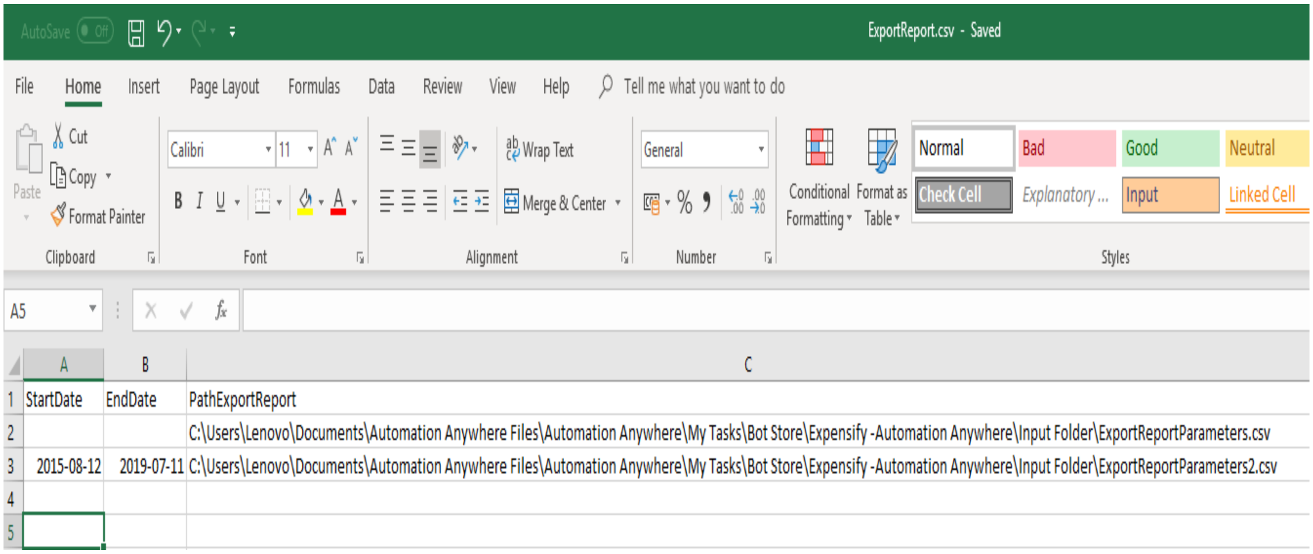
- **Required Parameters-**

File Parameter Name	Type	Description	Example Input
PathExportReport	Text	Absolute file path where export report file parameters are present.	C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\ ExpensifyBot-AppPerfectCorporation \Input Folder\ExportReportParameters.csv

- **Optional Parameters –**

File Parameter Name	Type	Description	Example Input

StartDate	Text	Filters out all reports submitted or created before the given date, whichever occurred last (yyyy-mm-dd)	2015-07-08
EndDate	Text	Filters out all reports submitted or created after the given date, whichever occurred last (yyyy-mm-dd)	2015-07-08



	A	B	C
1	StartDate	EndDate	PathExportReport
2			C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\ExportReportParameters.csv
3	2015-08-12	2019-07-11	C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\ExportReportParameters2.csv
4			
5			

ExportReportParameters.csv

- **Required Parameters-**

File Parameter Name	Type	Description	Example Input
RecipientMail	Text	Email of all receiving persons	nikhil@abc.com

- **Optional Parameters -**

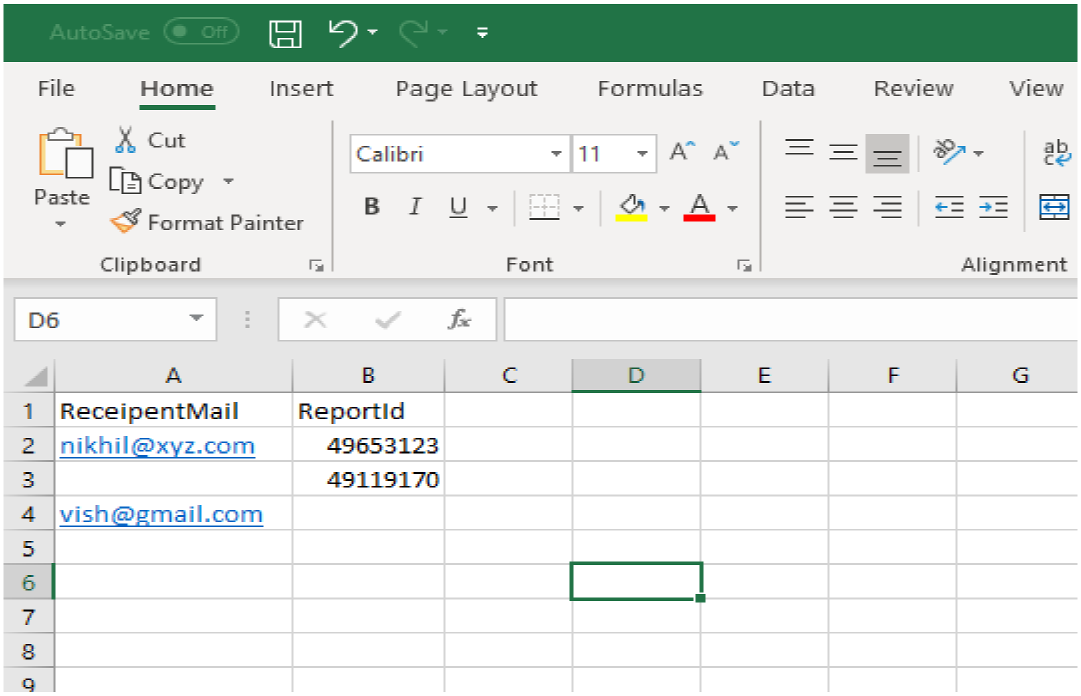
File Parameter Name	Type	Description	Example Input
ReportId	Text	The ID of the report	49700998

- **Points To Note**
  - Report Id and Start date can never be empty simultaneously.

(Example- If you leave report Id column empty in ExportReport.csv then their respective row in Export Report.csv file can't have Start date empty).

- There is no relation between Recipient Mail and Report Id parameter
- To find report Id User can use this utility and sends start Date, end Date and leaves ReportId column empty which sends a CSV file on recipient mail that contains column entry for report Name and their respective report Id.

Example -



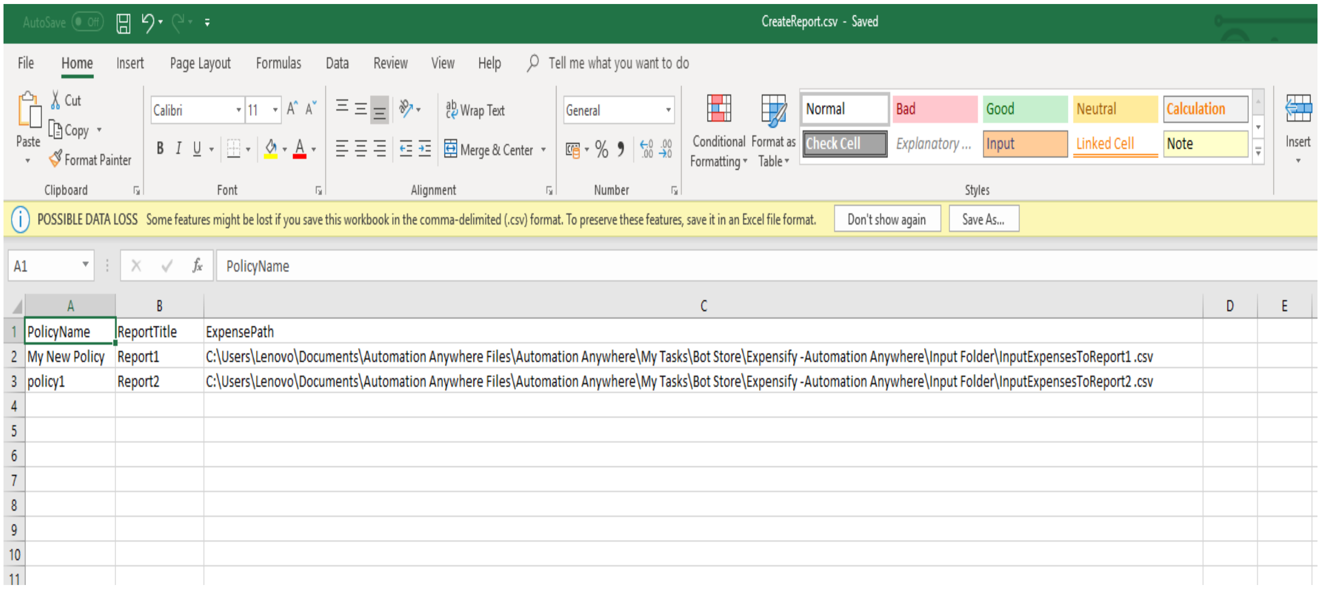
	A	B	C	D	E	F	G
1	ReceipentMail	ReportId					
2	nikhil@xyz.com	49653123					
3		49119170					
4	vish@gmail.com						
5							
6							
7							
8							
9							

2. CreateReport.csv

- **Required Parameters-**

File Parameter Name	Type	Description	Example Input
PolicyName	Text	The name of the policy where the report is created	Policy1
ReportTitle	Text	The title of the report	Report_New
ExpensePath	Text	The name of the expense's merchant	C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\ ExpensifyBot-AppPerfectCorporation \Input Folder\InputExpensesToReport .csv

Example-



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

A1 PolicyName

	A	B	C
1	PolicyName	ReportTitle	ExpensePath
2	My New Policy	Report1	C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\InputExpensesToReport1 .csv
3	policy1	Report2	C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\InputExpensesToReport2 .csv

InputExpensesToReport.csv

- **Required Parameters-**

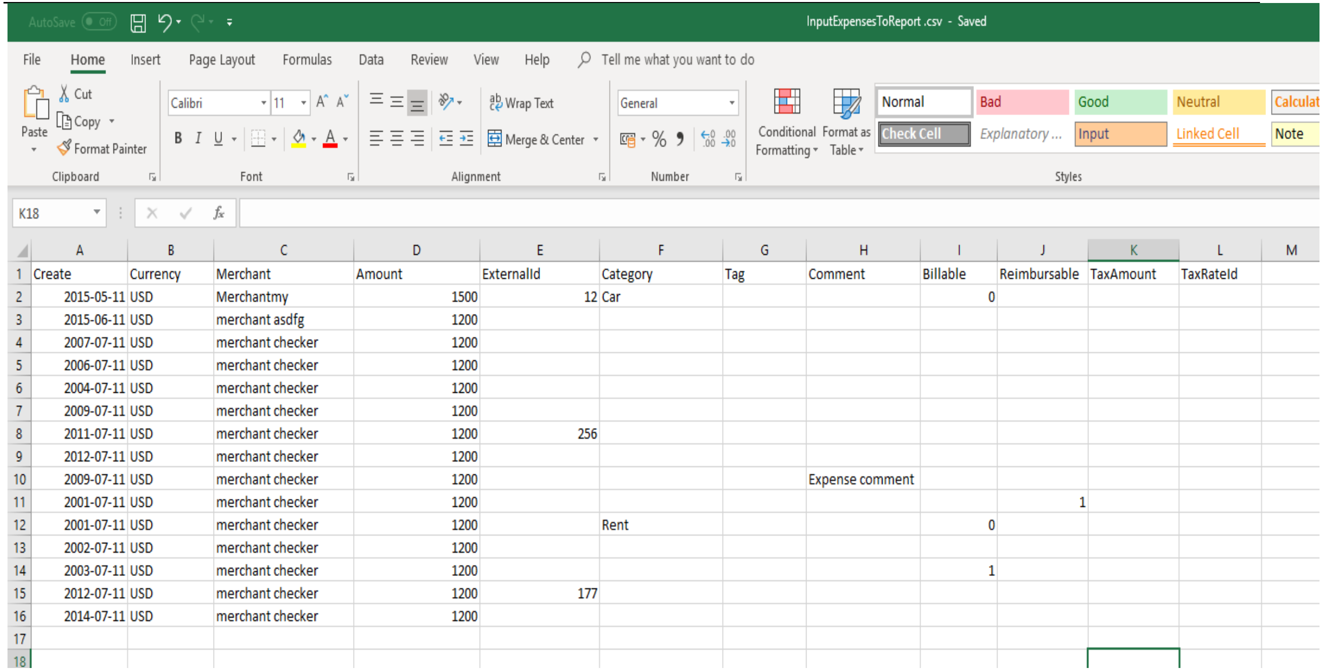
File Parameter Name	Type	Description	Example Input
Create	Text	The date of the expense (format yyyy-mm-dd).	2015-07-08
Currency	Text	The three-letter currency code of the expense	USD
Merchant	Text	The name of the expense's merchant	Merchant try
Amount	Text	The amount of the expense, in cents	1500

- **Optional Parameters -**

(If you don't want to add any value in the parameter leave it empty)

File Parameter Name	Type	Description	Example Input
ExternalId	Text	A unique, custom string that you specify - this will help identify the expense after being exported	12345
Category	Text	The name of the category to assign to the expense	Rent
Tag	Text	The name of the tag to assign to the expense	Tag Xyz
Comment	Text	An expense comment.	Expense Comment
Billable	Text	Whether to mark the expense as billable or not 0 for false 1 for true	1
Reimbursable	Text	Whether to mark the expense as reimbursable or not 0 for false 1 for true	0
TaxAmount	Text	The tax amount paid on the expense	1500
TaxRateId	Text	The tax rateID, as defined in the policy.	5

Example –



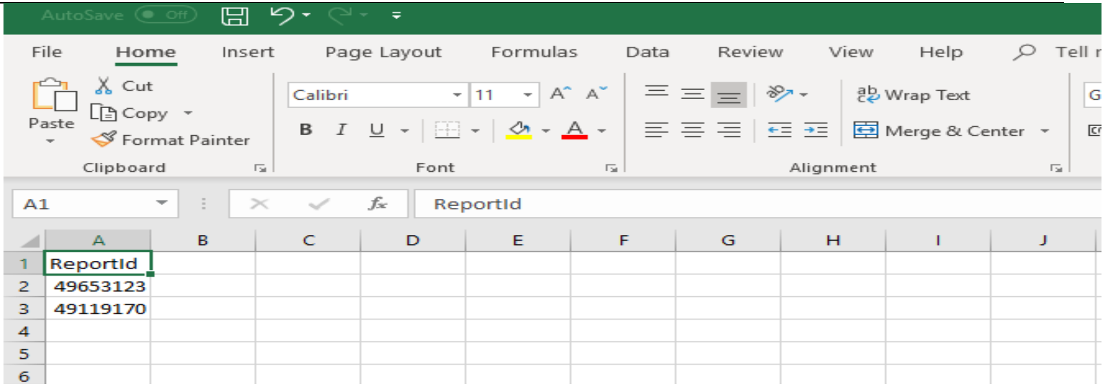
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Create	Currency	Merchant	Amount	ExternalId	Category	Tag	Comment	Billable	Reimbursable	TaxAmount	TaxRateId	
2	2015-05-11	USD	Merchantmy	1500	12	Car			0				
3	2015-06-11	USD	merchant asdfg	1200									
4	2007-07-11	USD	merchant checker	1200									
5	2006-07-11	USD	merchant checker	1200									
6	2004-07-11	USD	merchant checker	1200									
7	2009-07-11	USD	merchant checker	1200									
8	2011-07-11	USD	merchant checker	1200	256								
9	2012-07-11	USD	merchant checker	1200									
10	2009-07-11	USD	merchant checker	1200				Expense comment					
11	2001-07-11	USD	merchant checker	1200						1			
12	2001-07-11	USD	merchant checker	1200		Rent			0				
13	2002-07-11	USD	merchant checker	1200									
14	2003-07-11	USD	merchant checker	1200					1				
15	2012-07-11	USD	merchant checker	1200	177								
16	2014-07-11	USD	merchant checker	1200									
17													
18													

3. UpdateReportId.csv

- **Required Parameters-**

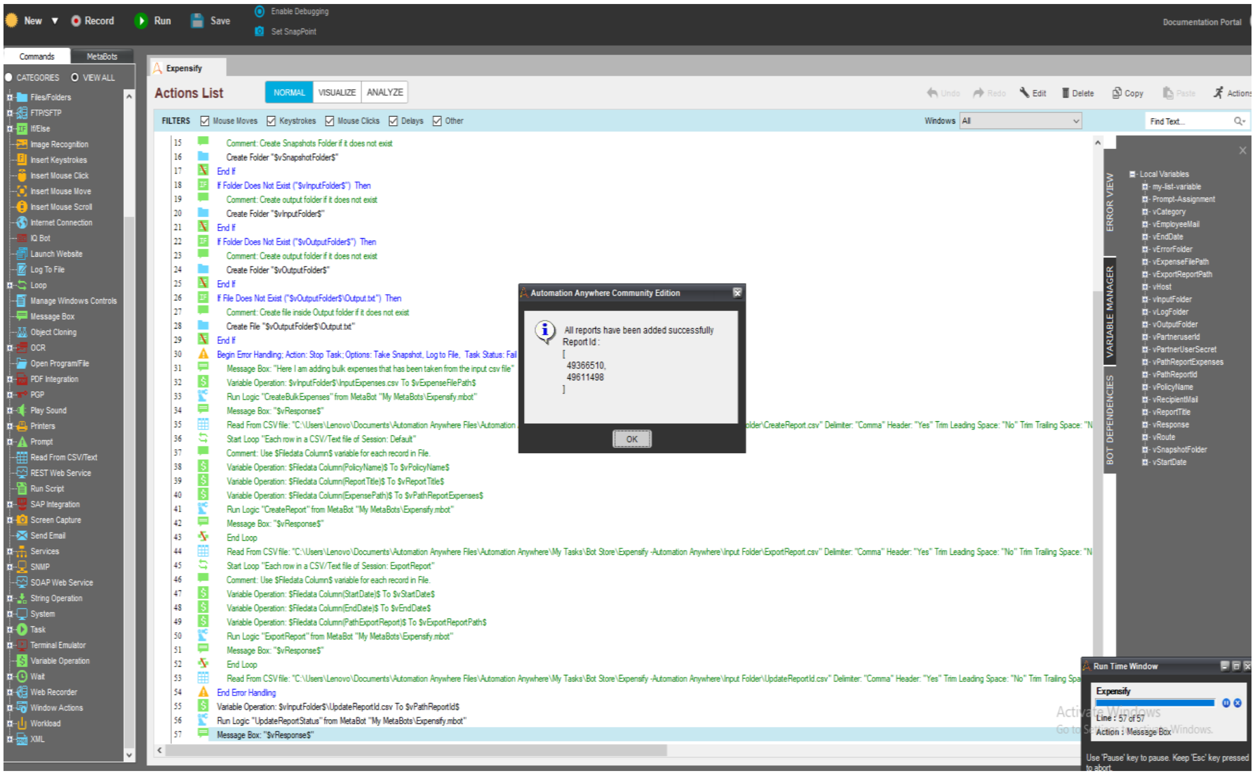
File Parameter Name	**Type**	**Description**	**Example Input**
ReportId	Text	Report Id whose status Has been updated from closed/approved to reimbursed	49700998

Example -



	A
1	ReportId
2	49653123
3	49119170
4	
5	
6	

Output: Message Box



FILTERS ☑ Mouse Moves ☑ Keystrokes ☑ Mouse Clicks ☑ Delays ☑ Other

```
    Comment: Create Snapshots Folder if it does not exist
    Create Folder "$vSnapshotFolder$"
End If
If Folder Does Not Exist ("$vInputFolder$") Then
    Comment: Create output folder if it does not exist
    Create Folder "$vInputFolder$"
End If
If Folder Does Not Exist ("$vOutputFolder$") Then
    Comment: Create output folder if it does not exist
    Create Folder "$vOutputFolder$"
End If
If File Does Not Exist ("$vOutputFolder$\Output.txt") Then
    Comment: Create file inside Output folder if it does not exist
    Create File "$vOutputFolder$\Output.txt"
End If
Begin Error Handling; Action: Stop Task; Options: Take Snapshot, Log to File,  Task Status: Fail
    Message Box: "Here I am adding bulk expenses that has been taken from the input csv file"
    Variable Operation: $vInputFolder$\InputExpenses.csv To $vExpenseFilePath$
    Run Logic "CreateBulkExpenses" from MetaBot "My MetaBots\Expensify.mbot"
    Message Box: "$vResponse$"
    Read From CSV file: "C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation                older\CreateReport.csv" Delimiter: "Comma" Header: "Yes" Trim Leading Space: "No" Trim Trailing Space: "N
    Start Loop "Each row in a CSV/Text file of Session: Default"
    Comment: Use $Filedata Column$ variable for each record in File.
    Variable Operation: $Filedata Column(PolicyName)$ To $vPolicyName$
    Variable Operation: $Filedata Column(ReportTitle)$ To $vReportTitle$
    Variable Operation: $Filedata Column(ExpensePath)$ To $vPathReportExpenses$
    Run Logic "CreateReport" from MetaBot "My MetaBots\Expensify.mbot"
    Message Box: "$vResponse$"
    End Loop
    Read From CSV file: "C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\ExportReport.csv" Delimiter: "Comma" Header: "Yes" Trim Leading Space: "No" Trim Trailing Space: "N
    Start Loop "Each row in a CSV/Text file of Session: ExportReport"
    Comment: Use $Filedata Column$ variable for each record in File.
    Variable Operation: $Filedata Column(StartDate)$ To $vStartDate$
    Variable Operation: $Filedata Column(EndDate)$ To $vEndDate$
    Variable Operation: $Filedata Column(PathExportReport)$ To $vExportReportPath$
    Run Logic "ExportReport" from MetaBot "My MetaBots\Expensify.mbot"
    Message Box: "$vResponse$"
    End Loop
    Read From CSV file: "C:\Users\Lenovo\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\Expensify -Automation Anywhere\Input Folder\UpdateReportId.csv" Delimiter: "Comma" Header: "Yes" Trim Leading Space: "No" Trim Trailing Spa
End Error Handling
Variable Operation: $vInputFolder$\UpdateReportId.csv To $vPathReportId$
Run Logic "UpdateReportStatus" from MetaBot "My MetaBots\Expensify.mbot"
Message Box: "$vResponse$"
```

Automation Anywhere Community Edition

```
All reports have been added successfully
Report Id :
[
 49366510,
 49611498
]
```

OK

4. Reports

There are no Bot Insight Reports generated for this Bot.

5. Logs

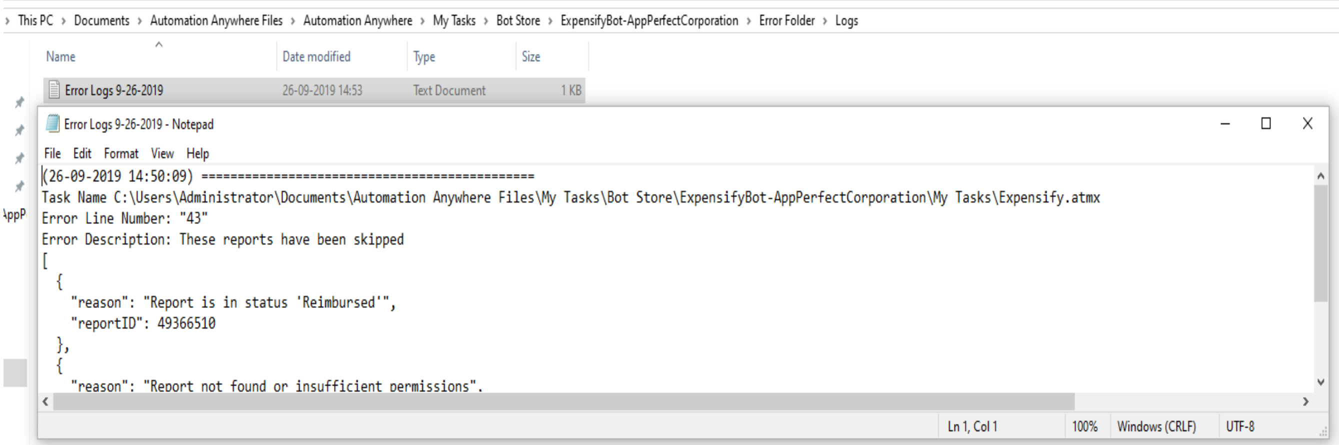
In case of Errors, Error Logs & Screenshots are generated within Error Folder (Highlighted below in yellow)

- My Tasks
    - Bot Store
        - Bot Name
            - Error Folder
                - Logs (Folder)
                    - Error Logs Month-Day-Year.txt
                - Snapshots (Folder)
                    - Error Snap Month-Day-Year HourMinSec.png

Error Logs will contain the below information -

- Generated Timestamp
- Task Name
- Error Line Number
- Error Description

Example of Errors Generated Error Logs File

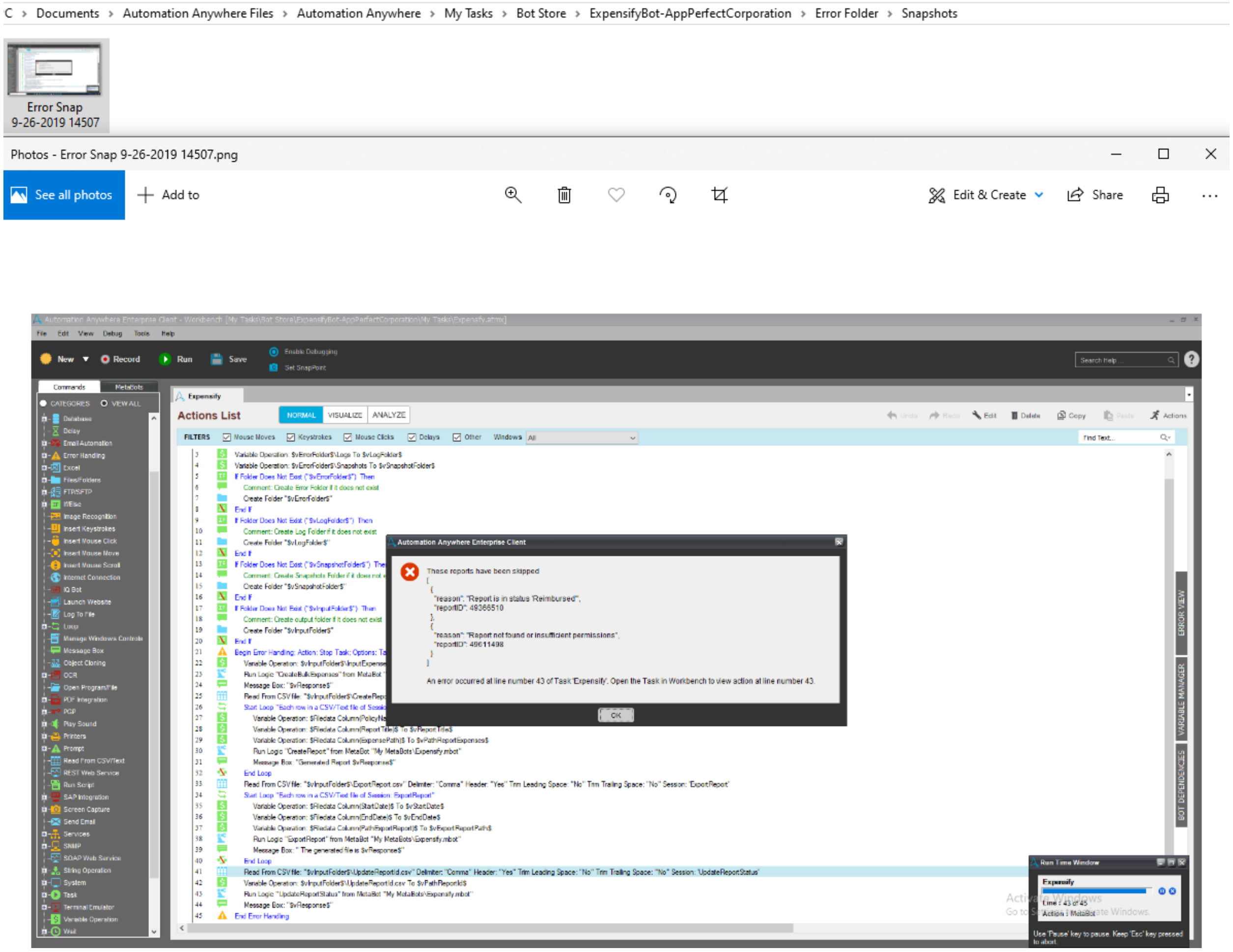


› This PC › Documents › Automation Anywhere Files › Automation Anywhere › My Tasks › Bot Store › ExpensifyBot-AppPerfectCorporation › Error Folder › Logs

Name	Date modified	Type	Size
Error Logs 9-26-2019	26-09-2019 14:53	Text Document	1 KB

```
(26-09-2019 14:50:09) =============================================
Task Name C:\Users\Administrator\Documents\Automation Anywhere Files\My Tasks\Bot Store\ExpensifyBot-AppPerfectCorporation\My Tasks\Expensify.atmx
Error Line Number: "43"
Error Description: These reports have been skipped
[
  {
    "reason": "Report is in status 'Reimbursed'",
    "reportID": 49366510
  },
  {
    "reason": "Report not found or insufficient permissions",
```

Example of Screenshot Generated Error File:



C › Documents › Automation Anywhere Files › Automation Anywhere › My Tasks › Bot Store › ExpensifyBot-AppPerfectCorporation › Error Folder › Snapshots

Error Snap 9-26-2019 14507

Photos - Error Snap 9-26-2019 14507.png

```
3   Variable Operation: $vErrorFolder$\Logs To $vLogFolder$
4   Variable Operation: $vErrorFolder$\Snapshots To $vSnapshotFolder$
5   If Folder Does Not Exist ("$vErrorFolder$") Then
6       Comment: Create Error Folder if it does not exist
7       Create Folder "$vErrorFolder$"
8   End If
9   If Folder Does Not Exist ("$vLogFolder$") Then
10      Comment: Create Log Folder if it does not exist
11      Create Folder "$vLogFolder$"
12  End If
13  If Folder Does Not Exist ("$vSnapshotFolder$") The
14      Comment: Create Snapshots Folder if it does not e
15      Create Folder "$vSnapshotFolder$"
16  End If
17  If Folder Does Not Exist ("$vInputFolder$") Then
18      Comment: Create output folder if it does not exist
19      Create Folder "$vInputFolder$"
20  End If
21  Begin Error Handling: Action: Stop Task; Options: Ta
22      Variable Operation: $vInputFolder$\InputExpense
23      Run Logic "CreateBulkExpenses" from MetaBot "
24      Message Box: "$vResponse$"
25      Read From CSV file: "$vInputFolder$\CreateRepo
26      Start Loop "Each row in a CSV/Text file of Sessio
27          Variable Operation: $Filedata Column(PolicyNa
28          Variable Operation: $Filedata Column(Report Title)$ To $vReport Title$
29          Variable Operation: $Filedata Column(ExpensePath)$ To $vPathReportExpenses$
30          Run Logic "CreateReport" from MetaBot "My MetaBots\Expensify.mbot"
31          Message Box: "Generated Report $vResponse$"
32      End Loop
33      Read From CSV file: "$vInputFolder$\ExportReport.csv" Delimter: "Comma" Header: "Yes" Trim Leading Space: "No" Trim Trailing Space: "No" Session: 'ExportReport'
34      Start Loop "Each row in a CSV/Text file of Session: ExportReport"
35          Variable Operation: $Filedata Column(StartDate)$ To $vStartDate$
36          Variable Operation: $Filedata Column(EndDate)$ To $vEndDate$
37          Variable Operation: $Filedata Column(PathExportReport)$ To $vExportReportPath$
38          Run Logic "ExportReport" from MetaBot "My MetaBots\Expensify.mbot"
39          Message Box: " The generated file is $vResponse$"
40      End Loop
41      Read From CSV file: "$vInputFolder$\UpdateReportId.csv" Delimter: "Comma" Header: "Yes" Trim Leading Space: "No" Trim Trailing Space: "No" Session: 'UpdateReportStatus'
42      Variable Operation: $vInputFolder$\UpdateReportId.csv To $vPathReportId$
43      Run Logic "UpdateReportStatus" from MetaBot "My MetaBots\Expensify.mbot"
44      Message Box: "$vResponse$"
45  End Error Handling
```

Automation Anywhere Enterprise Client

```
These reports have been skipped
[
  {
    "reason": "Report is in status 'Reimbursed'",
    "reportID": 49366510
  },
  {
    "reason": "Report not found or insufficient permissions",
    "reportID": 49611498
  }
]

An error occurred at line number 43 of Task 'Expensify'. Open the Task in Workbench to view action at line number 43.
```

OK

Run Time Window

Expensify

Line : 43 of 45

Action : MetaBot

Use 'Pause' key to pause. Keep 'Esc' key pressed to abort.

<u>Note</u>: There are no Custom Logs generated for this Bot.

6. Troubleshooting & Support

Support

Please visit our Support Portal for any assistance on Bot functionality or Feature.

Automation Anywhere provides a Product Documentation portal that can be accessed for more information about AA's products and guidance on building bots and Digital Workers.

The "Build" section of the portal includes these sections:

- Getting Started - information on building bots recommended practices (including use of the Credential Vault)
- Build Advanced Bots - details on MetaBots and the approach to integrating code into them
- Build Digital Workers - high-level architecture

FAQs

Q: The 'Create Locker' button is not visible in the Credential Vault Page. What do I do?

A: Please check if the role 'Locker Consumer' is assigned to the user and you are using an Enterprise Client version

Appendix A: Record of Changes

No.	Version Number	Date of Change (DD/MM/YYYY)	Author	Notes
1	1.0	27/08/2019	AppPerfectCorporation	Version 1 Release

Appendix B: Acronyms

No.	Acronym	Description
1	AA	Automation Anywhere
2	CSV	Comma Separated File
3	DW	Digital Worker
4	CR	Credential Vault
5	CRM	Customer Relationship Management
6	API	Application Programming Interface
7	PC	Personal Computer
8	RAM	Random Access Memory
9	OS	Operating System

Appendix C: References

No.	Topic	Reference Link
1	System Requirements - Client	*Click here*
2	System Requirements – Control Room	*Click here*
3	How to create credential & Locker?	*Click here*
4	Credential Overview	*Click here*
5	Apeople Community Site	*Click here*
6	Product Documentation portal	*Click here*