

# Automation Anywhere

## Enterprise

### Step by Step

### Command Examples Walkthroughs

## Copyright & Disclaimer statement

This training course is designed for information and education purpose only and the contents hereof are subject to change without notice. This course is not warranted to be error free, nor is it subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this training course, and no contractual obligations are formed either directly or indirectly by this course. This training course either wholly or any of its parts may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

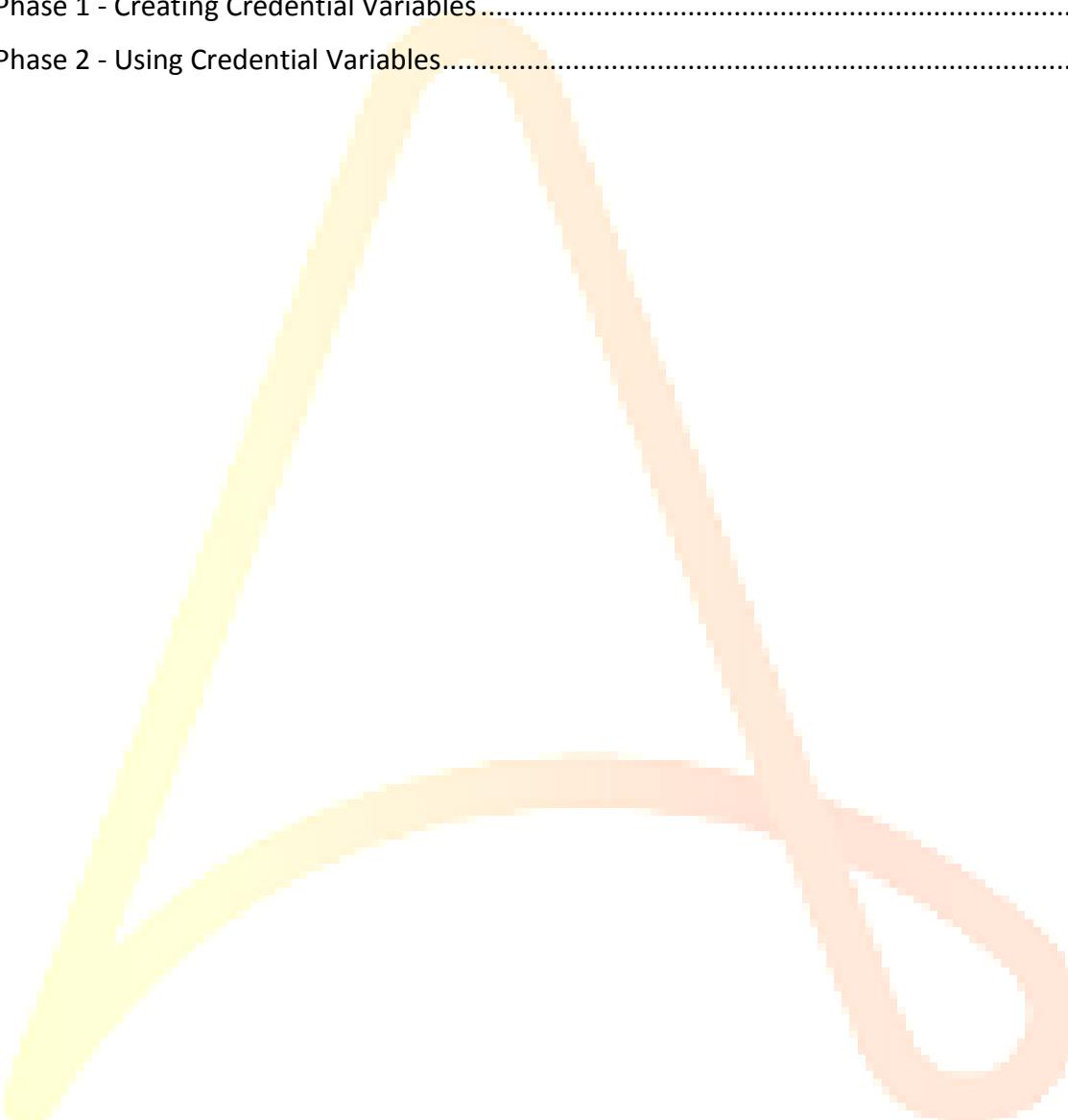
© Copyright. Automation Anywhere, Inc. 2016. All rights reserved

## Table of Contents

1) Recording and Running Simple Tasks, Properties, Schedule, Trigger .....	1
Objective .....	1
Screen Recorder .....	1
Properties/Schedule/Trigger.....	6
1) Properties – General .....	6
2) Properties – Repeat.....	7
3) Properties – Speed .....	7
4) Properties – Notification .....	7
5) Properties – Hotkey.....	10
6) Properties – Security .....	11
Schedule .....	13
1) Schedule – One Time Only .....	13
Trigger .....	14
1) Trigger – Window .....	14
2) Trigger – File .....	15
3) Trigger – Folder .....	16
4) Trigger – Performance.....	17
5) Trigger – Process .....	18
6) Trigger – Service .....	19
2) Web Recorder .....	20
Objective .....	20
Example 1 (Extract Table demo) .....	20
Example 2 (Extract Data demo).....	24
Example 3 (Extract Data - Pattern).....	28
3) Object cloning.....	41
Objective .....	41
Example: Object Cloning a Windows Application .....	41
4) Excel Commands.....	51
Objective .....	51
Example 1: Reading Data from a CSV/TXT and writing into an Excel spreadsheet .....	51

Example 2: Reading Data from an Excel File and Writing into a Database.....	62
5) Database .....	70
Objective .....	70
Example 1: Writing into a Database.....	70
Example 2: Reading from a Database into a CSV file .....	80
Example 3: Reading from a Database row by row .....	86
6) Files and Folders/Loop Each File in a Folder .....	93
Objective .....	93
Pre-Requisites: .....	93
Example: Copy/Rename/Delete Files in a Folder.....	93
7) PDF Integration.....	117
Objective .....	117
Example: PDF Integration.....	117
8) String Operations.....	126
Objective .....	126
Example: String Operation Commands .....	126
9) PGP.....	136
Objective .....	136
Example: PGP .....	136
10) Email Automation .....	142
Objective .....	142
Example: Email Automation.....	142
11) Error Handling.....	151
Objective .....	151
Example: Error Handling .....	151
12) File Transfer Protocol.....	163
Objective .....	163
Example: File Transfer Protocol .....	163
13) XML .....	177
Objective .....	177
Example: XML Document .....	177
14) Terminal Emulator .....	187
Objective .....	187

Example: Terminal Emulator .....	187
15) Services .....	198
Objective .....	198
Example: Services .....	199
16) Credential Manager .....	207
Objective .....	207
Phase 1 - Creating Credential Variables .....	207
Phase 2 - Using Credential Variables.....	212



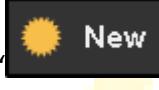
## 1) Recording and Running Simple Tasks, Properties, Schedule, Trigger

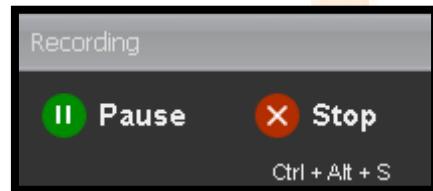
### Objective

- Step by step demonstration of Screen Recording and Executing Simple Tasks including various features like Properties, Schedule and Trigger.

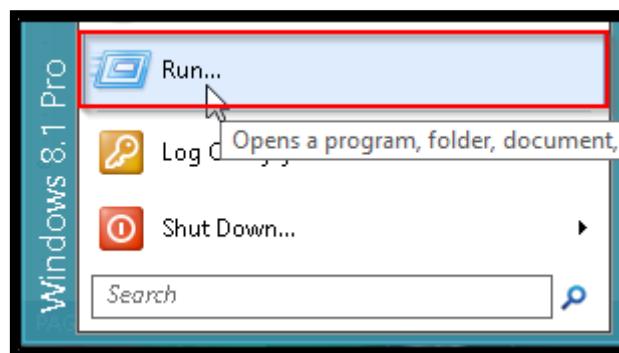
### Screen Recorder

- Start your Automation Anywhere Client, prior to that, log-in to the Automation Anywhere Control Room so that the client can be registered successfully and connects to the control room.

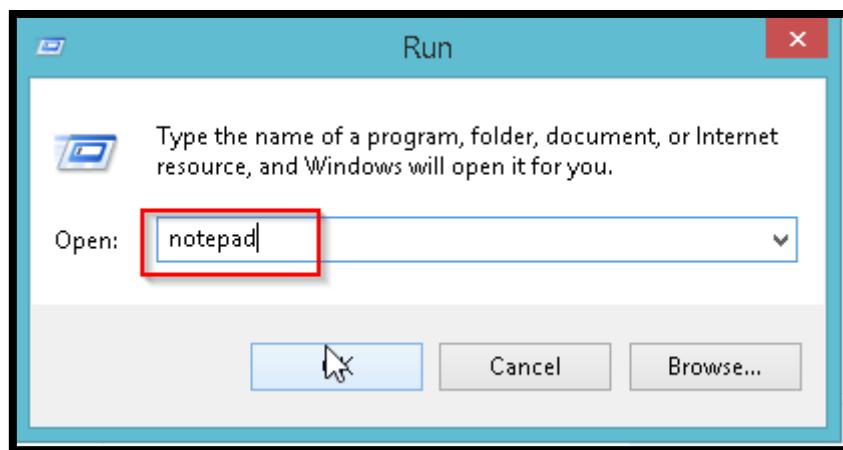
- Click on “” -> ” to start recording .
- The “Recording” floating bar appears at the right bottom of the screen as seen below.



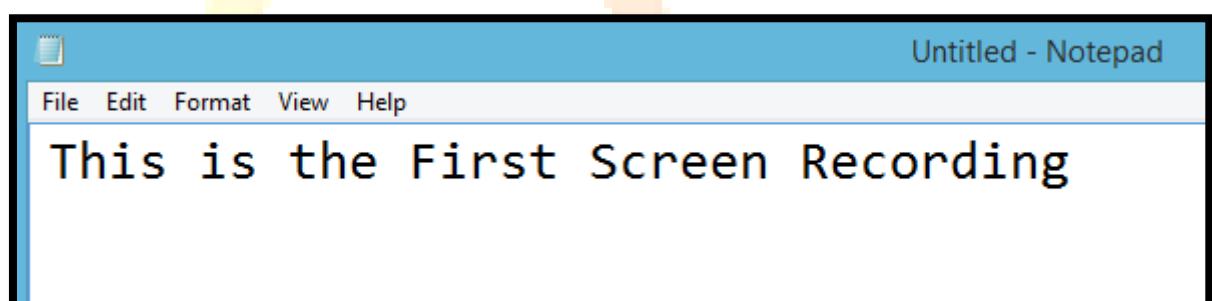
- Now, whatever actions are performed will be recorded by the A.A recorder.
- Click on Start-> Run as below.



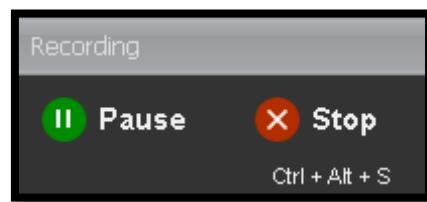
- Then, go and type notepad as below.



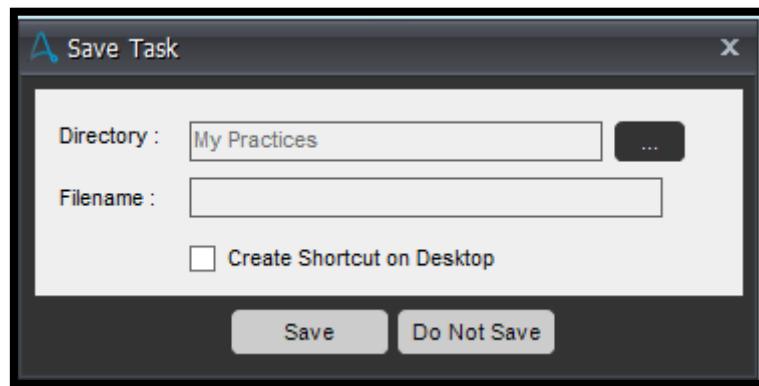
- The notepad window is launched and write a single line inside it as seen below.



- Stop the recording by clicking on the “Stop” button in the recording button as shown below.



- Subsequently, the “Save Task” dialog pops up, go in and type the name of the Task.



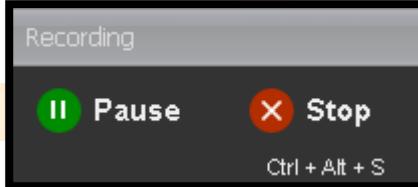
- The task is created and can be seen in the list of Tasks as seen below.

**ScreenRecorder1.atmx**

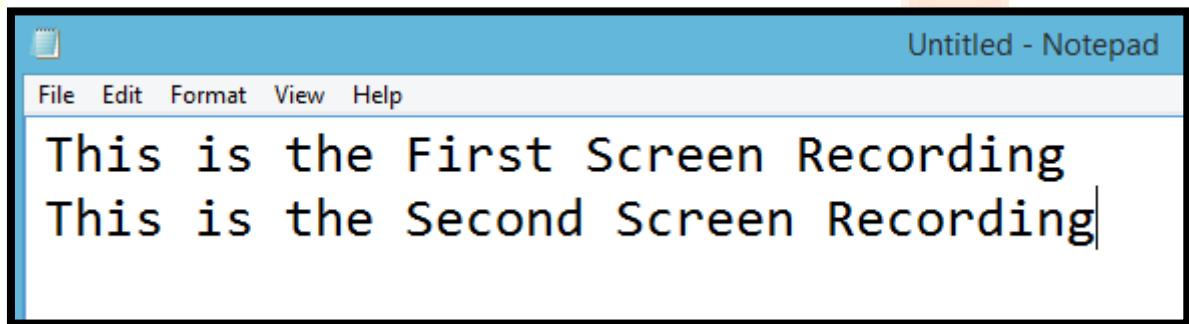
- Subsequently, start another screen recording as before, by clicking on “”

>” **Screen Recorder >** ”

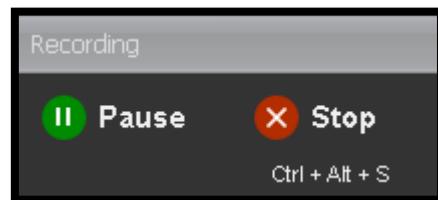
- Again the Recording Window comes up as below.



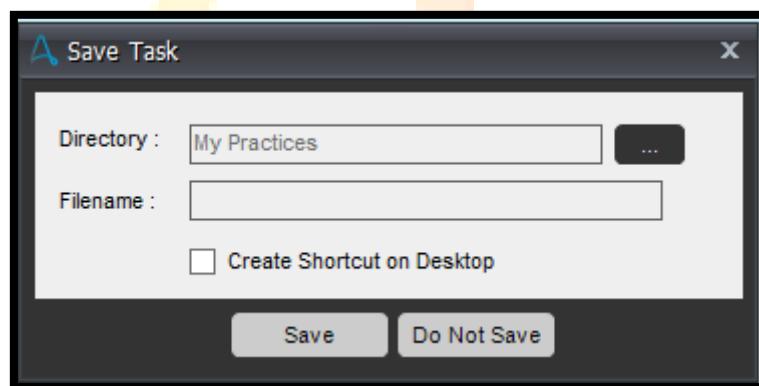
- Add another line to the notepad window that was written earlier , as seen below.



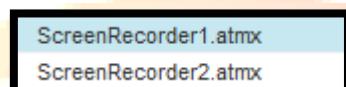
- As before, stop the recording using the “Stop” button as below.



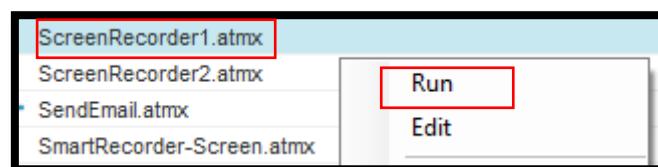
- “Save Task” dialog is popped up and the name of the script is entered and it is saved.



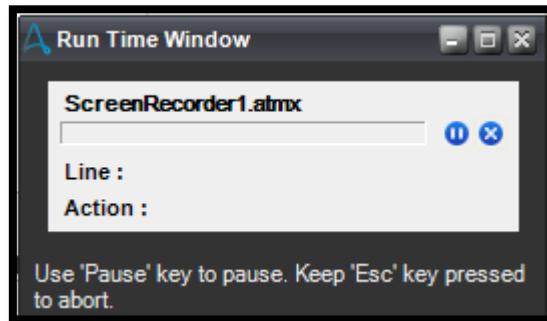
- Both the Tasks can be seen inside the Task Editor as below.



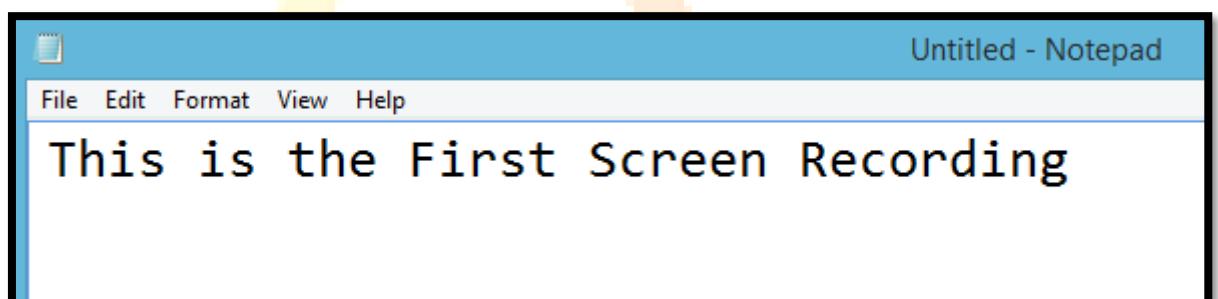
- Run both these tasks , one after the other , in the same sequence in which they were created. We right-click on the first recording that was created first and choose “Run” as below.



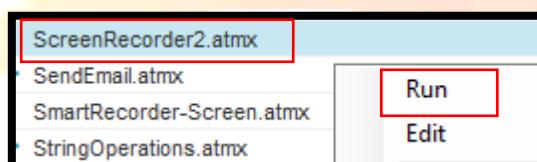
- The script starts running and the Run Time window can be seen as below.



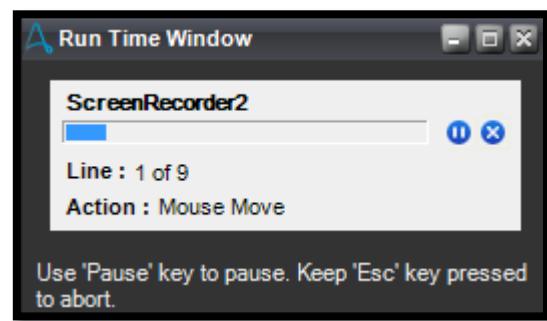
- Once the test is executed successfully, the first line getting written successfully in notepad as below.



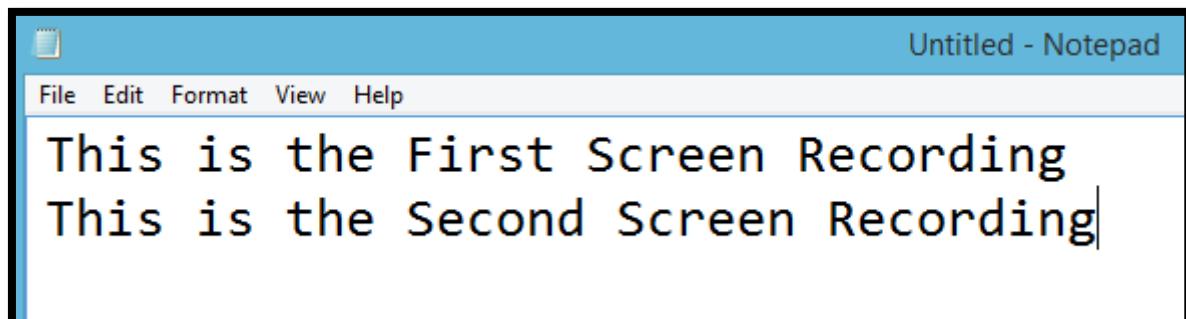
- Run the second script, as seen below.



- Again, the Runtime window come up as seen below.



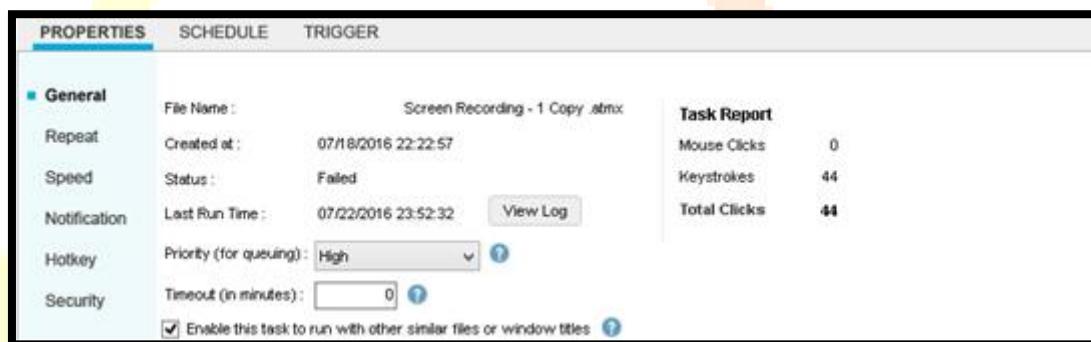
- Both the recorded lines have been written successfully inside the notepad window as below.



- Let's proceed to Schedule, Trigger & Properties.

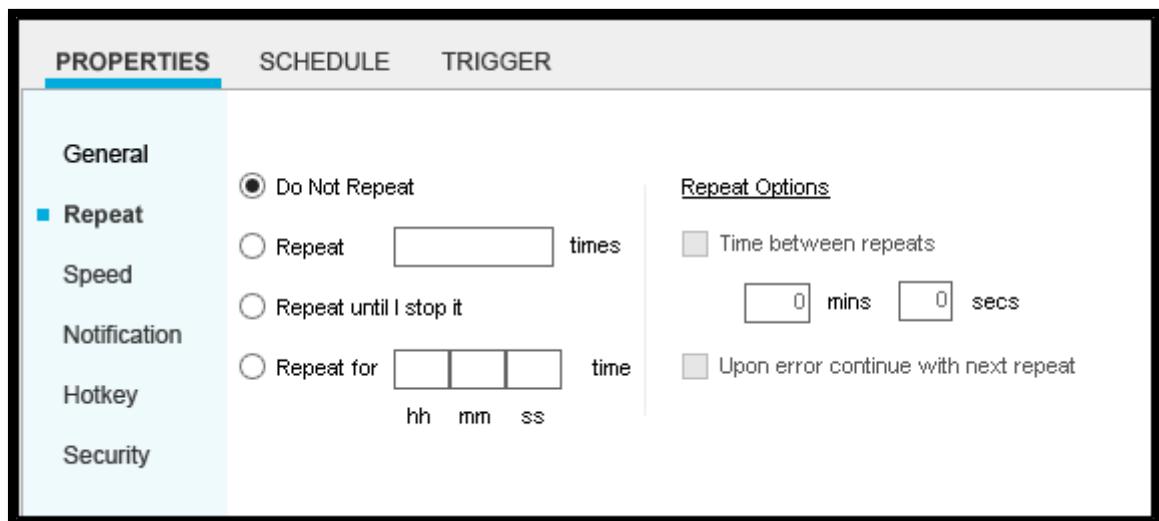
## Properties/Schedule/Trigger

### 1) Properties – General

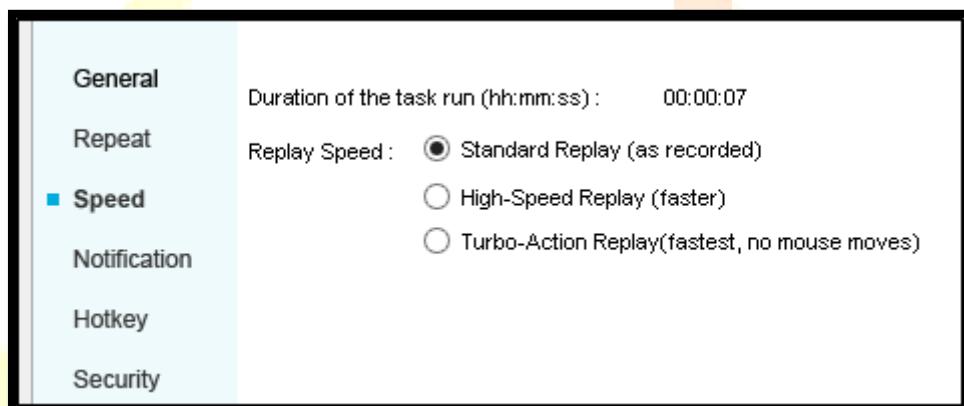


- The tab as seen above shows information like File Name, Date of Creation, Run Status, Last Run Time and Also it shows the number of Mouse Clicks/Keystrokes and Total Clicks.

## 2) Properties – Repeat

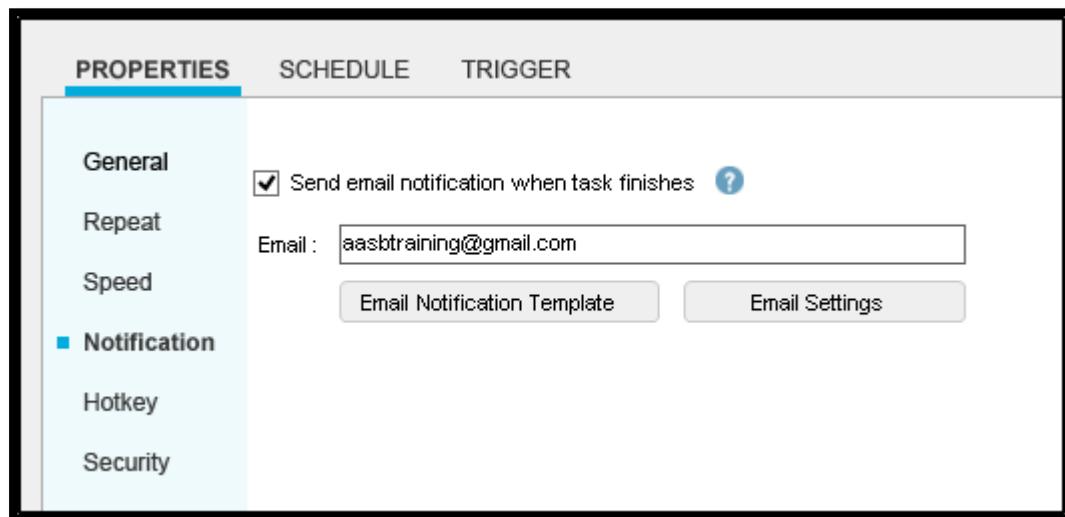


## 3) Properties – Speed

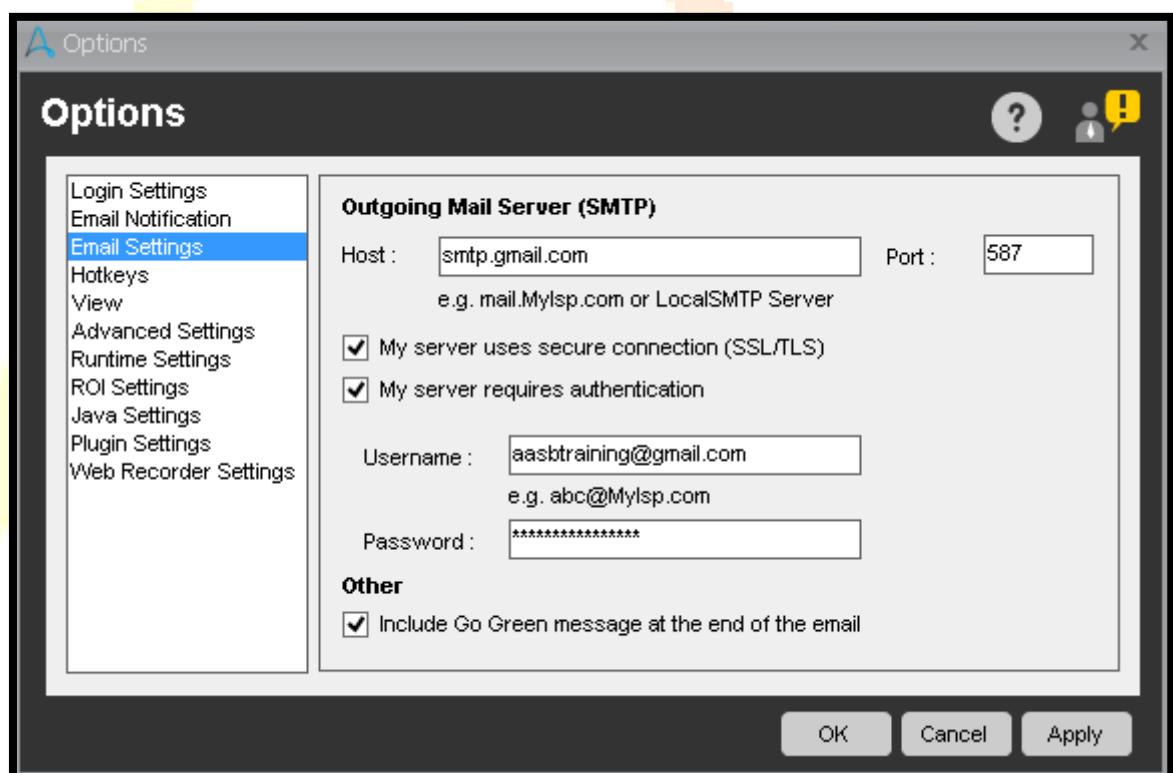


## 4) Properties – Notification

- The Notification window enables an email notification to be sent to the concerned whenever a Task Finishes, irrespective of its success or failure.



- Email Settings Tab, opens the following dialog, where details like SMTP Server, username and password to access the email can be specified as below: -

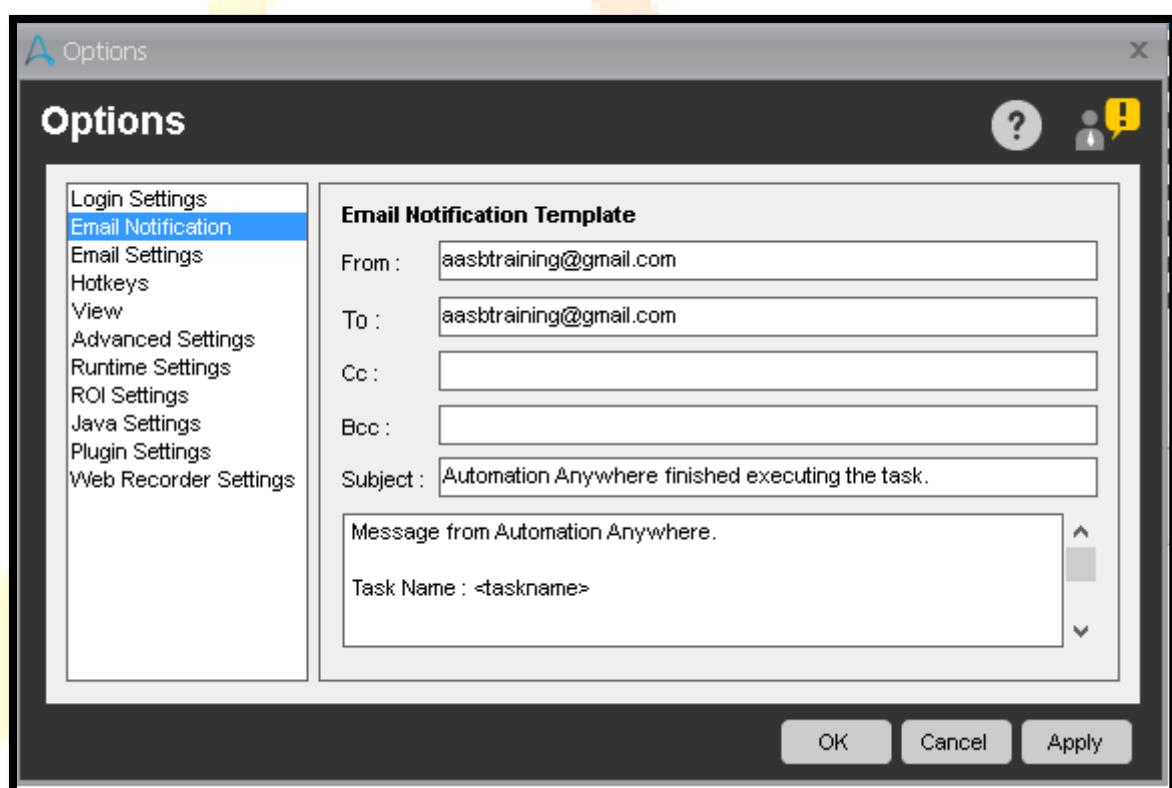


- For the sake of this demonstration we are using the smtp server settings of gmail as specified above . In case you want to send emails from your official email id , specify your office SMTP server details and your account credentials as above.
- To use this feature make sure the security of the email is less secure, else it will give an error.
- The below email credentials have been made specifically for Automation Anywhere and can be used for trying out this feature:

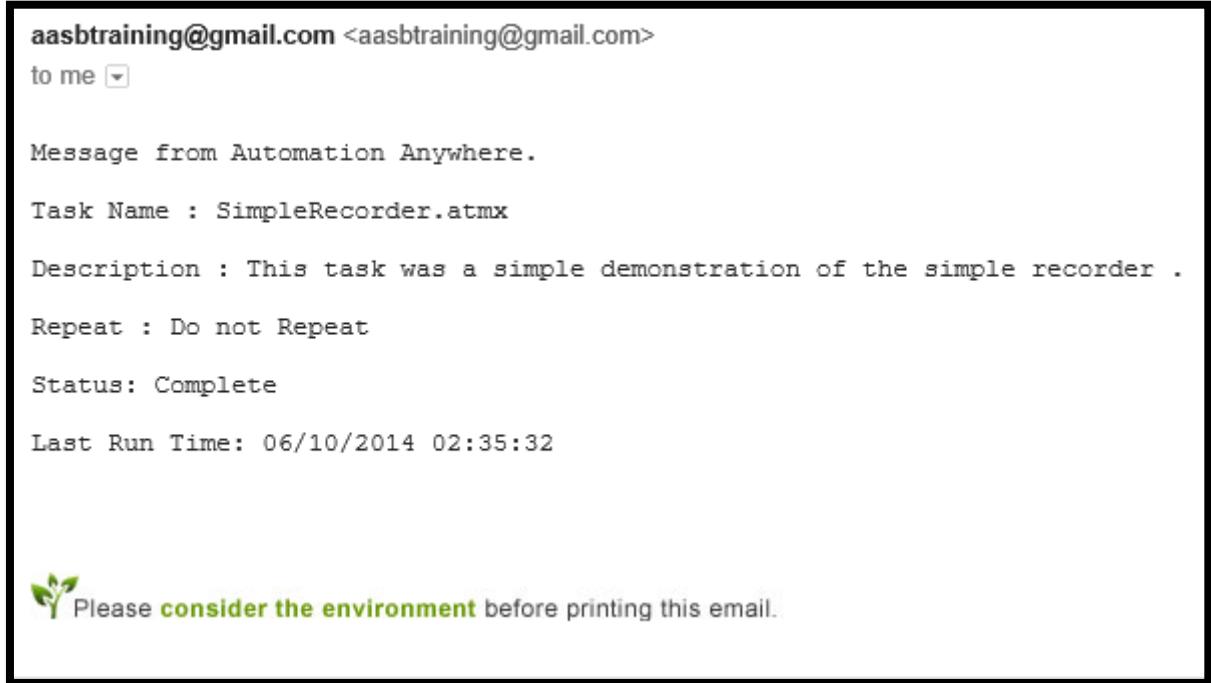
Username: [aasbtraining@gmail.com](mailto:aasbtraining@gmail.com)

Password: aasbtraining2014

- The format of the email can be specified in the Email Notification Template, clicking on that button opens the following dialog

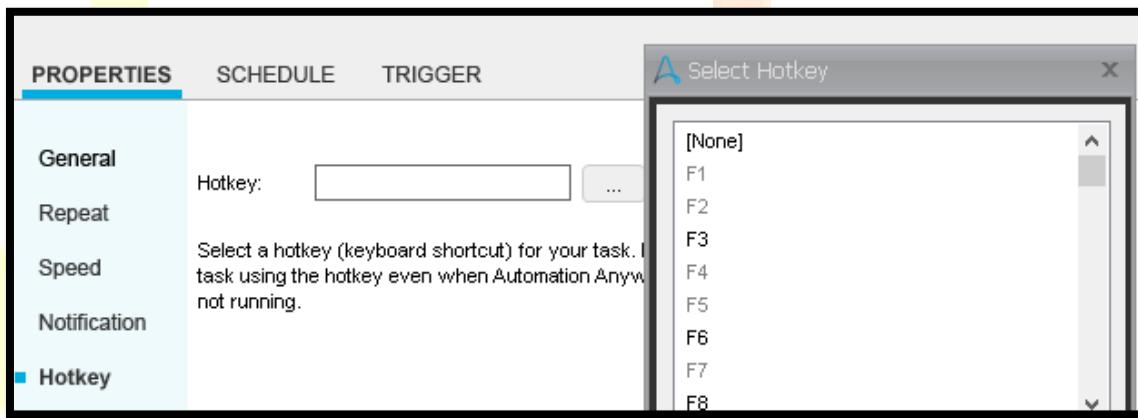


- Run our Task, when it completes an email is delivered in the inbox as below.



## 5) Properties – Hotkey

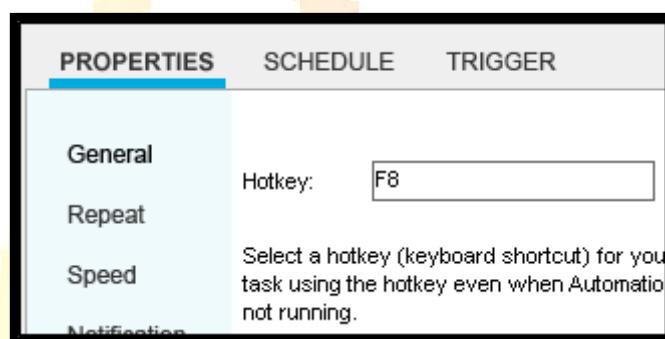
- This property helps to set up a key combination pressing which executes a Task.



- Click on Ellipsis above as shown, which shows you a list of key combinations from which any key can be chosen as shown.
- Note: The Greyed out keys are already being used, the bold keys are available for usage as below.

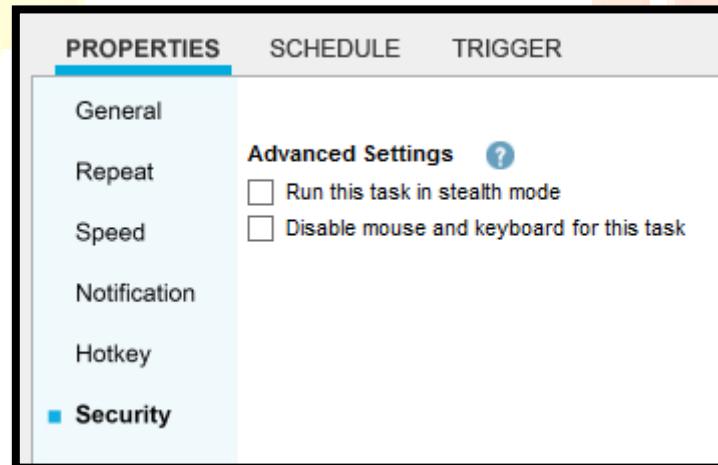


- Once a key is selected as shown, the output is as below.

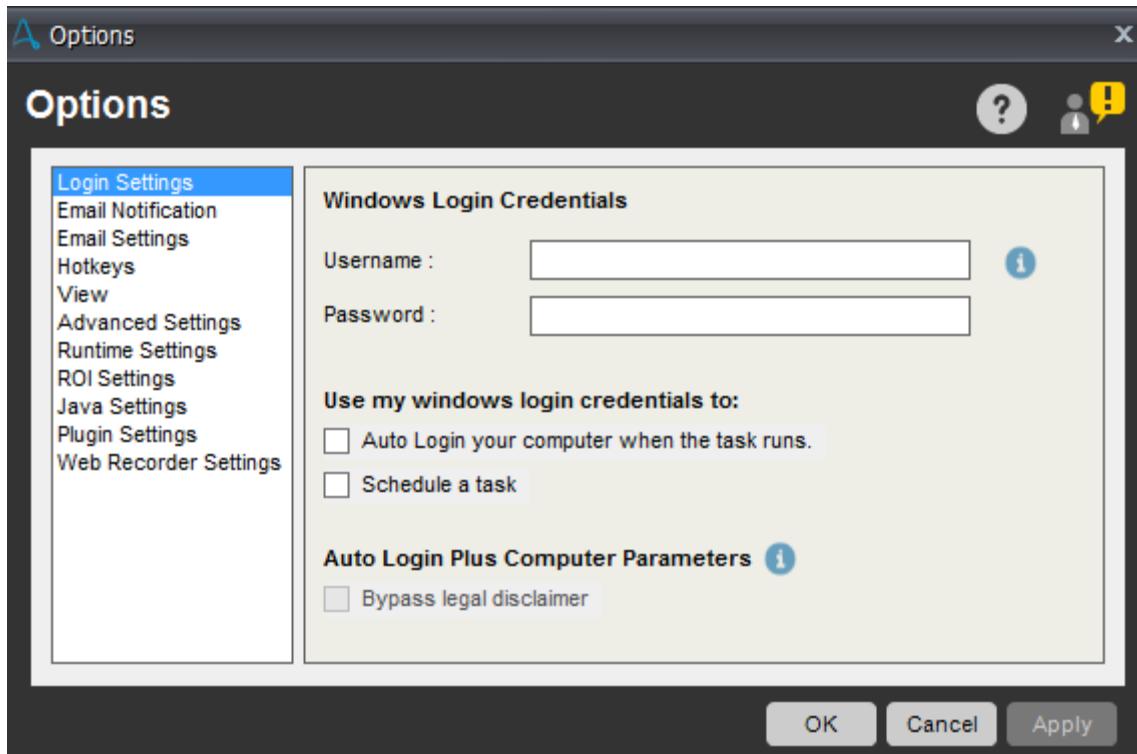


- On pressing F8, the Task Runs . So, the key F8 has been assigned successfully to execute this Task.

## 6) Properties – Security



- In “Tools – Options – Login Settings” from the main screen. The dialog is shown as below.

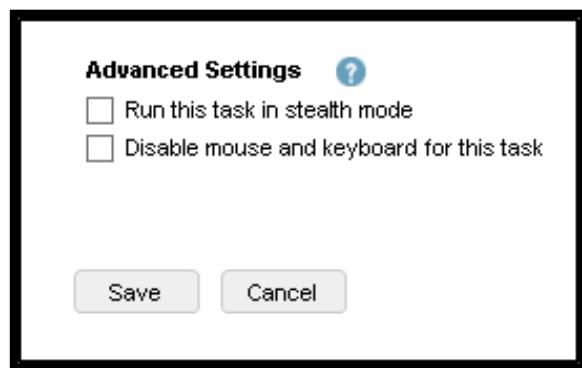


- Check Mark “Auto Login” check box above , enter the credentials in “Username” and “Password” above.

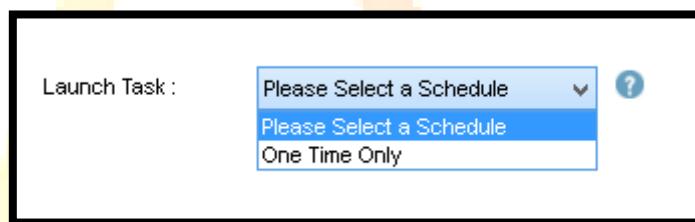
**Question-What does Auto Login Do ?**

**Answer-It allows you to run a task which has been scheduled to run without user intervention , at the pre-scheduled time of execution if the machine is locked, Auto Login will login , run the task and then lock the terminal.**

- During the above run a person may try to get unauthorized access to the terminal. To prevent the unauthorized access the followinf option is provided.



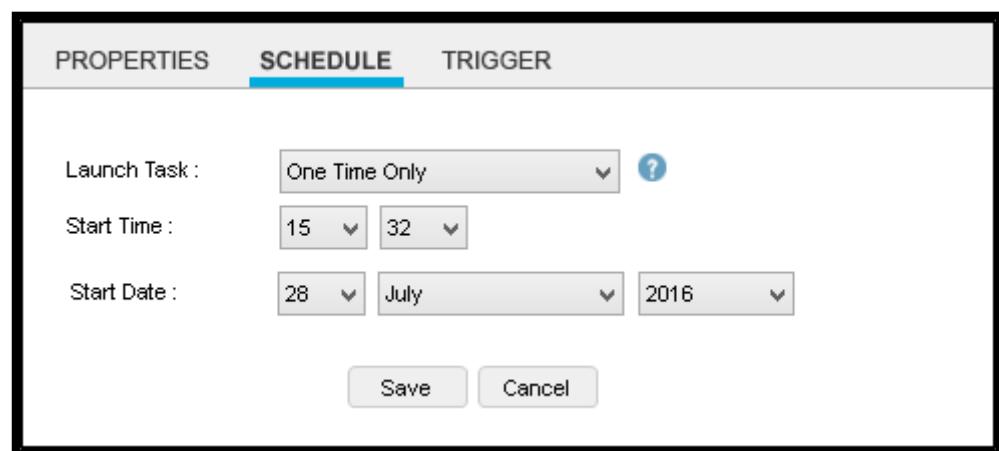
## Schedule



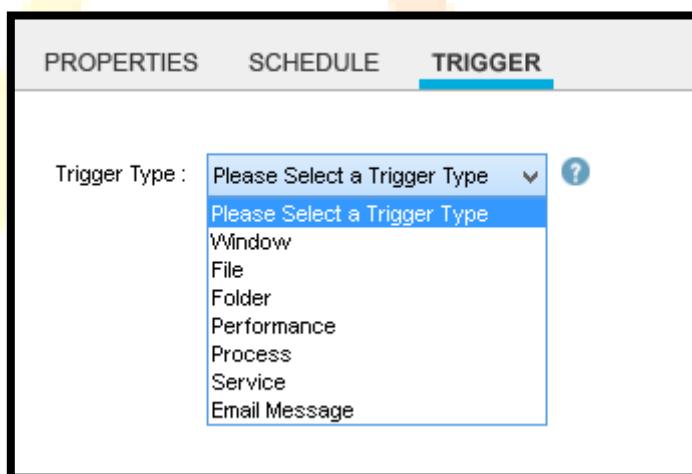
- The scheduler allows you to run your tasks at pre-determined times without user intervention.

### 1) Schedule – One Time Only

- This option helps to set up a schedule for executing the task once, for example as seen below.



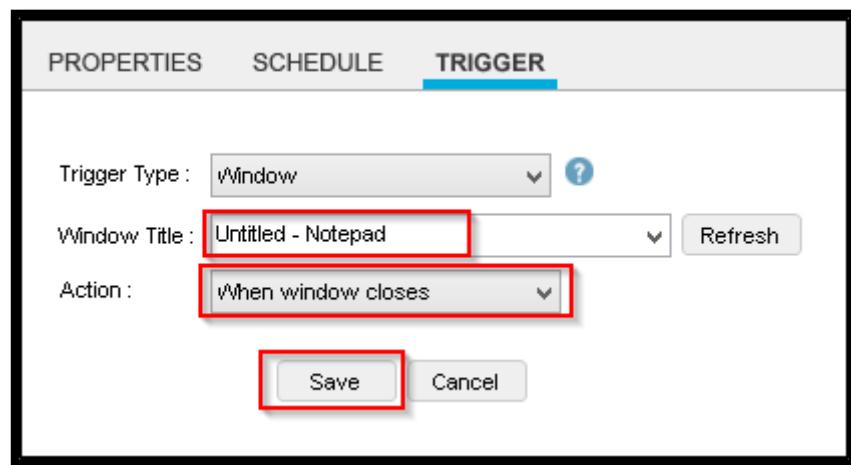
## Trigger



- The Trigger dialog helps in running the tasks based on the occurrence of a certain action or a certain condition getting fulfilled, this action or condition is known as Trigger.

### 1) Trigger – Window

- Specify a Task to get executed based on whether a window opens or closes as below.



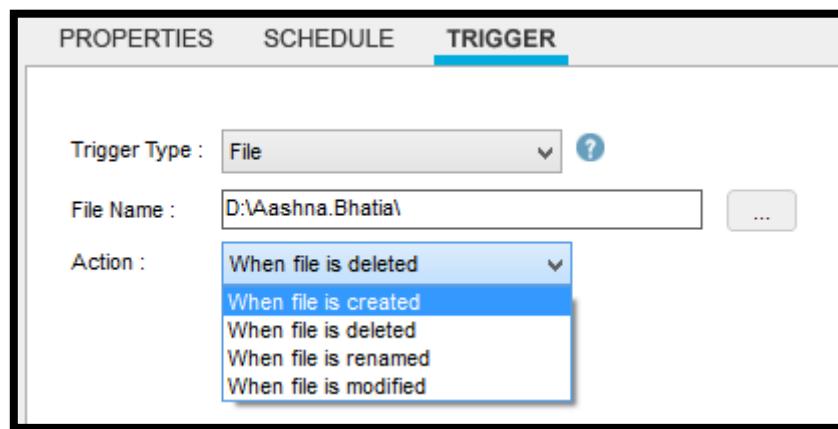
- In the above example the Task will be executed when the Untitled - Notepad window closes, clicking on save, makes the Trigger as seen below.

Trigger Type	Action
<input checked="" type="checkbox"/> Window	When 'Untitled - Notepad' window closes

Add      Delete

## 2) Trigger – File

- It helps running a task based on occurrence of any of the following events which may happen to the chosen file.



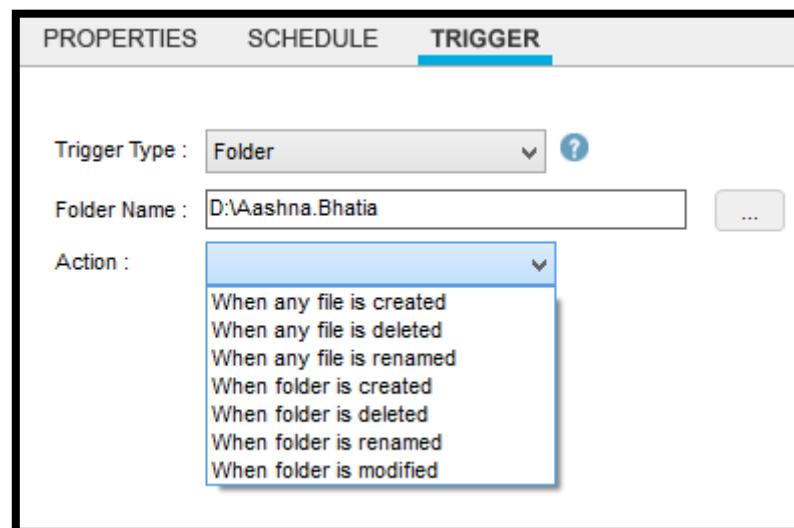
- As seen below, a Trigger on a file is created, when the file gets deleted.

Trigger Type	Action
<input checked="" type="checkbox"/> File	When 'D:\Aashna.Bhatia\' file is deleted

- So as soon as the file specified is deleted, the task is triggered as per the trigger.

### 3) Trigger – Folder

- It helps in running a task based on occurrence of any of the following events which may happen in the chosen folder

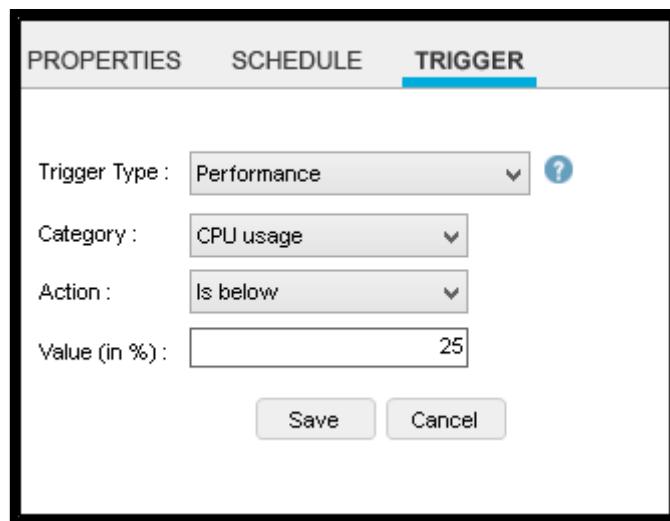


- As seen below the task will be executed whenever any file is created in the folder.

Trigger Type	Action
<input checked="" type="checkbox"/> Folder	When file is created in folder 'D:\Aashna.Bhatia'

#### 4) Trigger – Performance

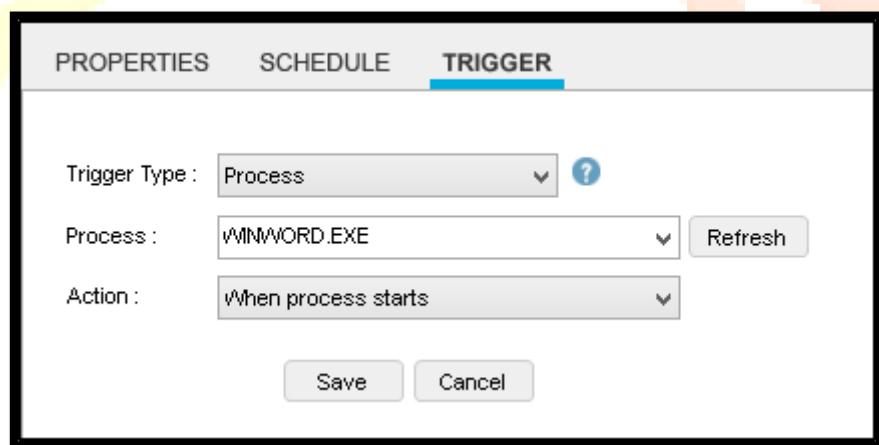
- It helps in running a task based on specific performance criteria that has been set as per the requirement and can be seen as below.



- As it can be seen above, a Trigger Type “Performance” is set on Category “CPU usage”, the Action specified is “Is Below” and the Value is “25” .
- So, whenever the CPU usage goes below 25% , the Task gets executed .

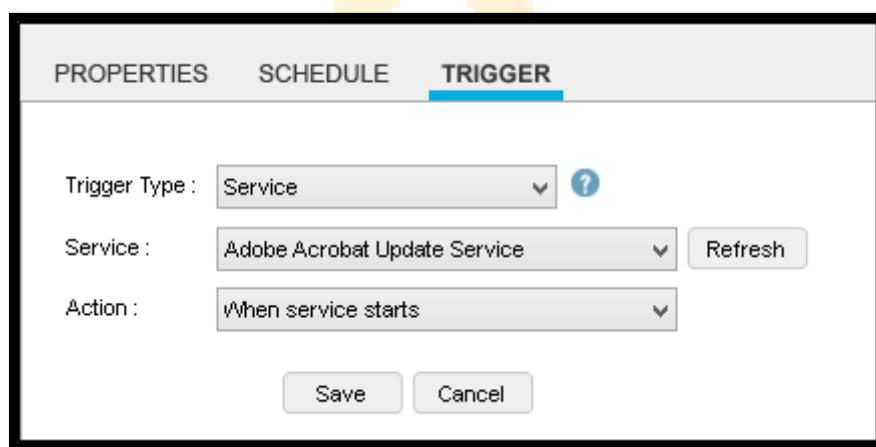
## 5) Trigger – Process

- It helps in running a task based on either the start or stop of a process as can be seen below.
- Here, a Trigger has been set that will fire when the “winword.exe” process starts, leading the task being executed .



## 6) Trigger – Service

- It helps in running a task based on either the start or stop of a service as can be seen below.
- Here, a Trigger will fire when the “Adobe Acrobat Update” service starts, leading the task getting executed.



## 2) Web Recorder

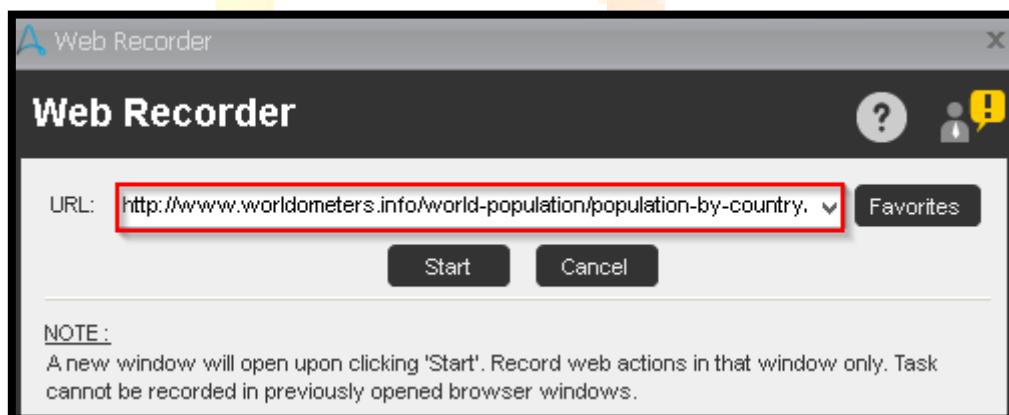
### Objective

- Step by step demonstration of Web Recorder and its various features like Extract Pattern Data, Extract Table and Extract Regular Data.

### Example 1 (Extract Table demo)

- Start Automation Anywhere Client, prior to that start your Automation Anywhere Web Control Room so that client can be registered successfully and connects with it.

- Click on “ ->” ” to start recording and the following window pops up.



- Type the URL as <http://www.worldometers.info/world-population/population-by-country/> as this is where the recording is to be done and click on start .
- The following warning is thrown , click on “OK” to proceed as below.



- The following Web Recorder bar appears at the bottom right corner of the screen.



- For the sake of this demo let's try to extract information from the web page into a table using "Extract Table" option shown in the recorder, click on "Extract Table" option from the "Web Recorder" dialog as shown above .
- Clicking on "Extract Table" prompts to click on the table on the web page from which data is to be extracted as shown below.



- Click on an area(Green Border) from which the data is to be extracted.

This list includes both **countries** and **dependent territories**. Data based on the latest *United Nations Population Division* estimates. Click on the name of the country or dependent territory to view historical data, and projected figures. See also: [World Population](#)

Web Recorder

[Go Back](#) [Extract Data](#) [Extract Table](#) [Stop Recording](#)

Search:

#	Country (or dependency)	Population (2016)	Yearly Change	Net Change	Density (P/Km <sup>2</sup> )	Area (Km <sup>2</sup> )	Migrants (net)	Fert. Rate	Med. Age	Urban Pop %	World Share
1	<a href="#">China</a>	1,382,323,332	0.46 %	6,274,389	147	9,390,784	-360,000	1.55	37	57.9 %	18.6 %
2	<a href="#">India</a>	1,326,801,576	1.2 %	15,751,049	446	2,972,892	-519,644	2.48	27	32.4 %	17.9 %
3	<a href="#">U.S.</a>	324,118,787	0.73 %	2,345,156	35	9,155,892	1,801,277	1.00	36	80.7 %	14.1 %
4	<a href="#">Indonesia</a>	260,581,100	1.17 %	3,017,285	144	1,812,108	-140,000	2.5	28	54 %	3.5 %
5	<a href="#">Brazil</a>	209,567,920	0.83 %	1,720,392	25	8,349,320	3,185	1.82	31	84.2 %	2.8 %
6	<a href="#">Pakistan</a>	192,826,502	2.07 %	3,901,628	250	770,998	-216,384	3.72	23	38.9 %	2.6 %
7	<a href="#">Nigeria</a>	186,987,563	2.63 %	4,785,601	205	910,802	-60,000	5.74	18	49 %	2.5 %
8	<a href="#">Bangladesh</a>	162,910,864	1.19 %	1,915,222	1,252	130,172	-445,296	2.23	26	34.9 %	2.2 %
9	<a href="#">Russia</a>	143,439,832	-0.01 %	-17,086	9	16,299,981	223,577	1.66	39	73.2 %	1.9 %

- Choose options in the dialog that opens below

**Extracted Table**

Preview (Up to 50 rows)

Sr. No.	Column1	Column2	Column3	Column4	Column5
1	#	Country (or dep...	Population(2016)	YearlyChange	NetCh
2	1	China	1,382,323,332	0.46 %	6,274,
3	2	India	1,326,801,576	1.2 %	15,75
4	3	U.S.	324,118,787	0.73 %	2,345,
5	4	Indonesia	260,581,100	1.17 %	3,017,
6	5	Brazil	209,567,920	0.83 %	1,720,
7	6	Pakistan	192,826,502	2.07 %	3,901,
8	7	Nigeria	186,987,563	2.63 %	4,785,
9	8	Bangladesh	162,910,864	1.19 %	1,915,
10	9	Russia	143,439,832	-0.01 %	-17,08
11	10	Mexico	128,632,004	1.27 %	1,614,
12	11	Japan	126,323,715	-0.2 %	-249,7
13	12	Philippines	102,250,133	1.54 %	1,550,
14	13	Ethiopia	101,953,268	2.48 %	2,462

The table spans across multiple pages

**Note :** To extract tabular data from multiple pages, click on 'Capture' button and then select the button/link control that loads the next page.

Next Button Details

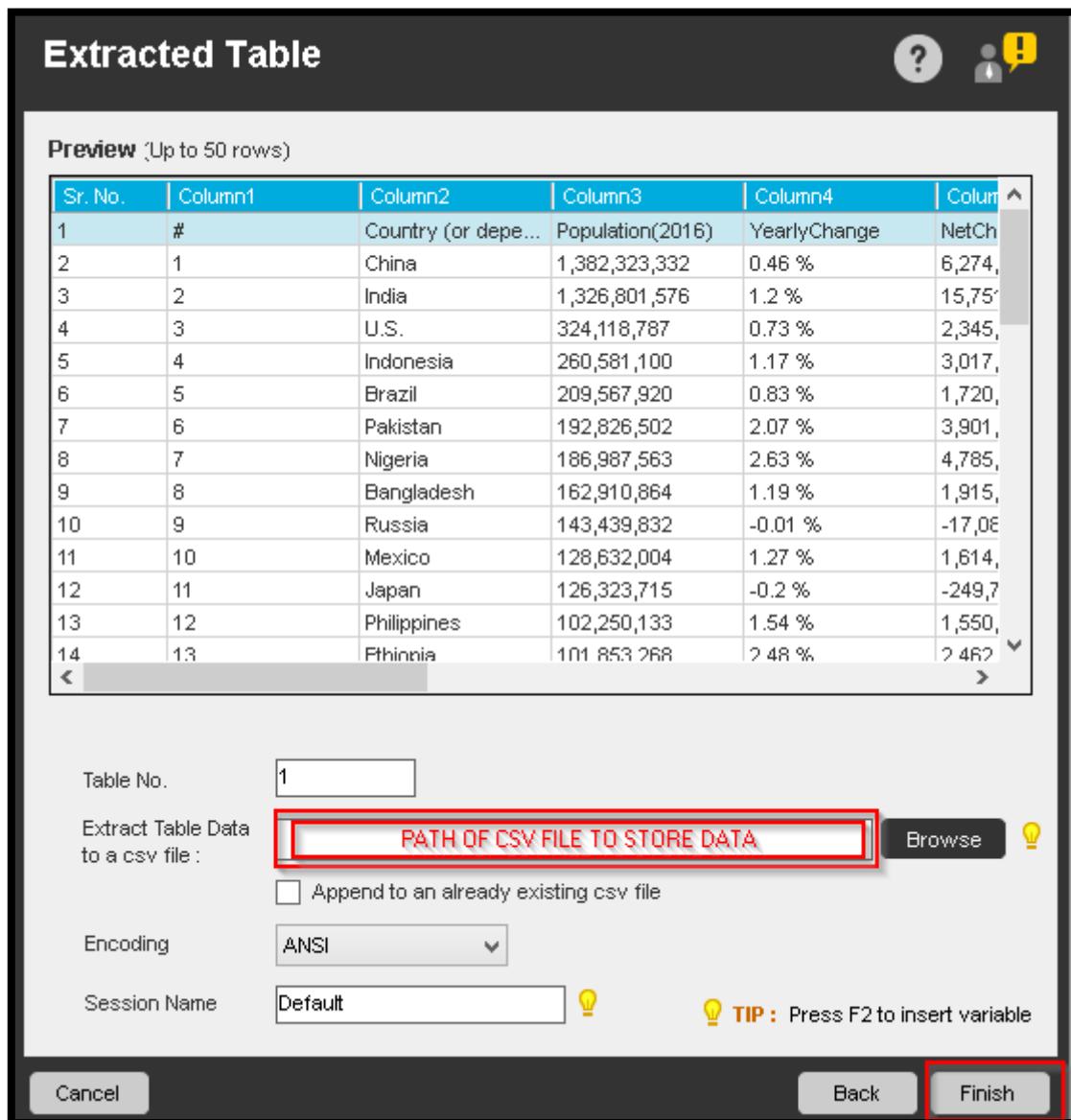
Control Type :

Value :

[Cancel](#) [Next](#)

- Click on Next in the dialog above

- The following dialog now opens where the CSV file in which the data is to be stored can be specified and then click on finish. It also shows the preview of the table data that has been extracted.



- Click on "Stop Recording" as shown below.



- Save the recording and run it .

- The Run time window appears next , showing which line is being executed , and what action is currently being played out .
- Once the Task has completed , the informationthat has been captured successfully into our CSV file as below can be checked.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Rank	Country	Population (2014)	1 Year Change	Population Change	Migrant s (net)	Median Age	Aged 60+	Fertility Rate	Area (Km <sup>2</sup> )	Density (P/Km <sup>2</sup> )	Urban Pop %	Urban Populati	Share of
2	1	China	1,393,783,836	0.59%	8,217,299	-313,996	35.7	14%	1.66	9,556,947	145	54%	#####	19.24%
3	2	India	1,267,401,849	1.22%	15,262,253	-483,402	26.6	9%	2.53	3,287,265	386	32%	#####	17.50%
4	3	U.S.A.	322,583,006	0.79%	2,532,290	1,008,835	37.5	20%	1.99	9,629,056	34	83%	#####	4.45%
5	4	Indonesia	252,812,245	1.18%	2,946,614	-141,488	28.1	8%	2.38	1,904,567	133	53%	#####	3.49%
6	5	Brazil	202,033,670	0.83%	1,671,745	-46,113	30.7	12%	1.83	8,514,209	24	85%	#####	2.79%

- Task Editor looks like as below

```

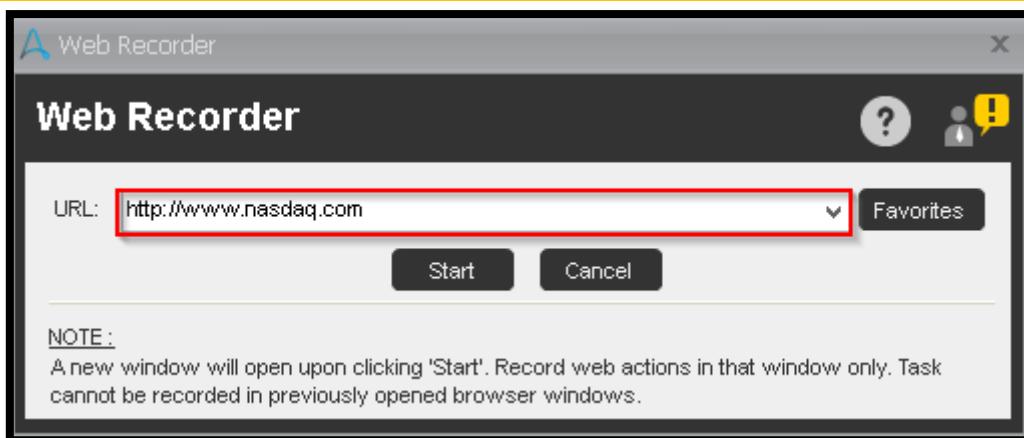
1  Open "http://www.worldometers.info/world-population/population-by-country/"
2  Extract table from 'http://www.worldometers.info/world-population/population-by-country/' website's '1' table

```

## Example 2 (Extract Data demo)

- Start Automation Anywhere Client, prior to that start your Automation Anywhere Web Control Room so that client can be registered successfully and connects with it.

- Click on “  -> 
 Page 24



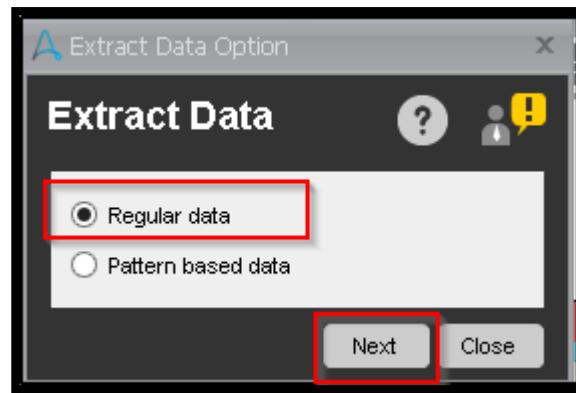
- Type the URL as <http://www.nasdaq.com> as the URL where the recording is to be done and click on start.
- The following warning is thrown , click on “OK” to proceed as below.



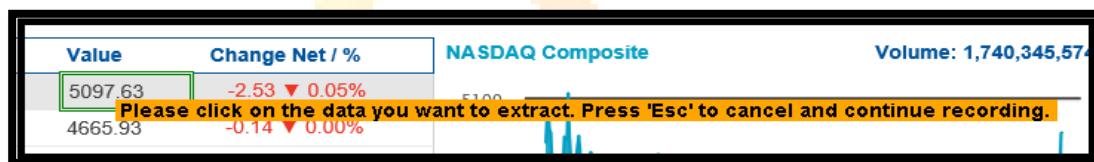
- The following Web Recorder bar appears at the bottom right corner of the screen.



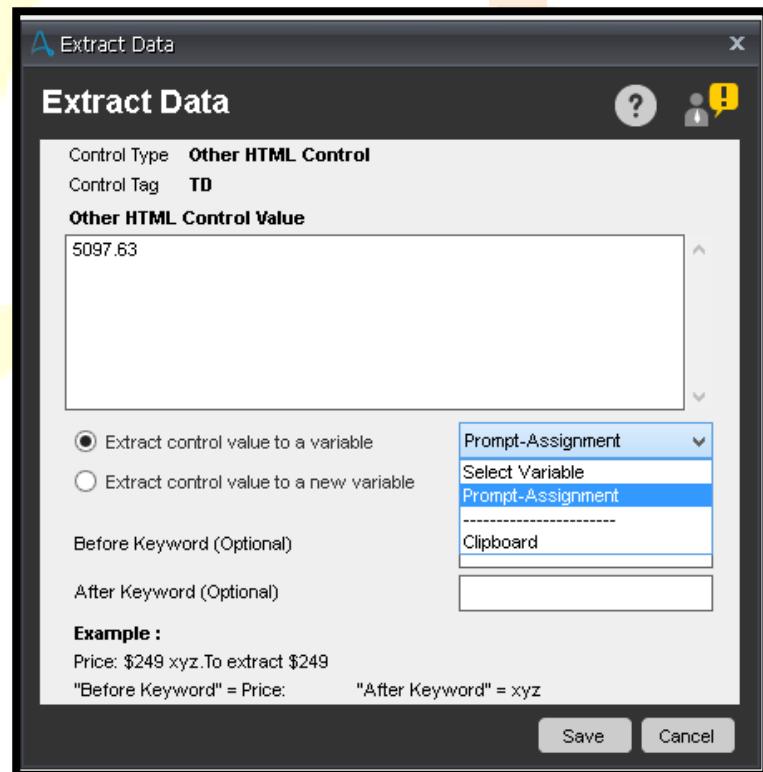
- For the sake of this demo we will try to extract information from the web page using “Extract Data” option shown in the recorder , click on “Extract Data” option from the “Web Recorder” dialog shown above.
- We choose “Regular Data” and click on “Next” as shown below.



- Click on Value for NASDAQ Index as shown below.



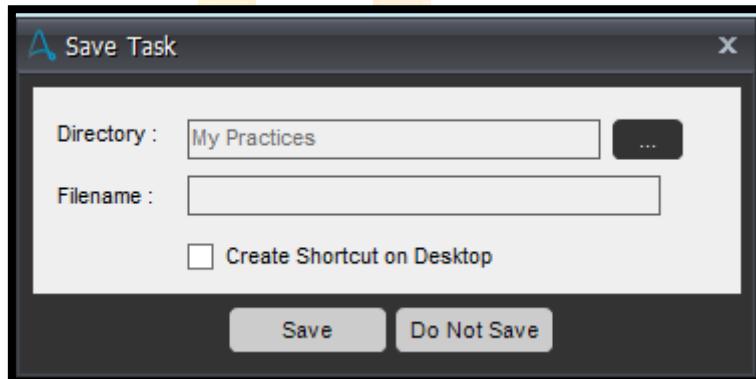
- The following dialog pops where we specify that the Value will be stored in “Prompt-Assignment” variable as below.



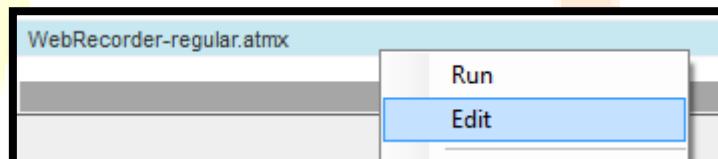
- Next, click on Stop Recording as shown below.



- Give the Recording a name in the dialog below.



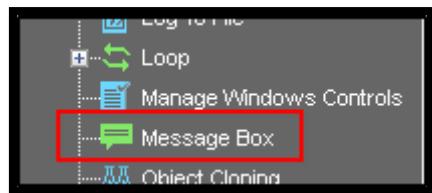
- Search for the Recording and let us edit it in the Task Editor as shown below.



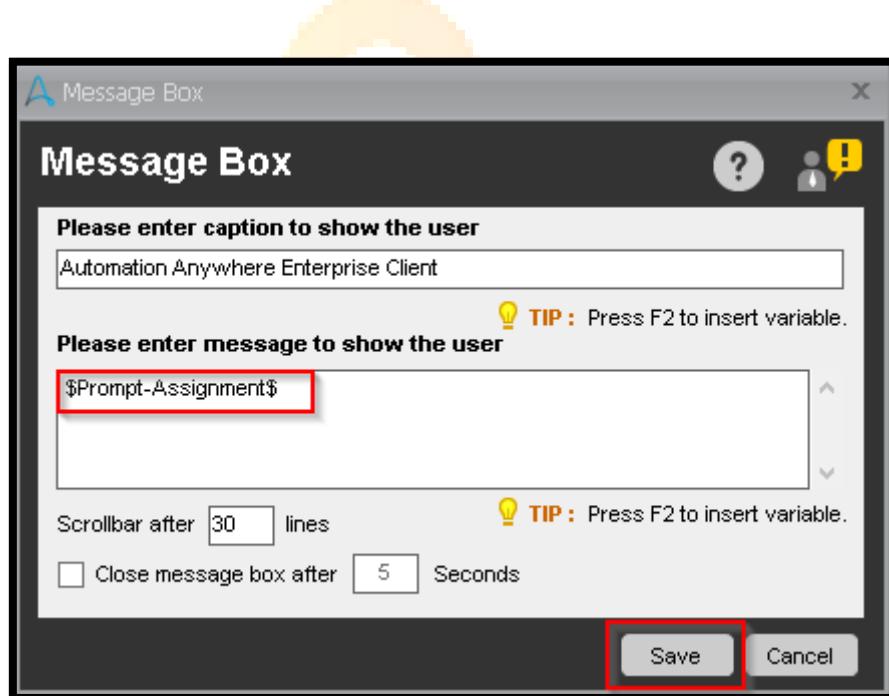
- This shows the following screen as below :-



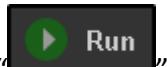
- To display the value of the NASDAQ Index value that has been captured, add a Message Box window from Commands as shown below.



- Selecting the message box opens the following dialog.



- Add a Message as above, Press F2 to add the variable in which the value has been stored (Prompt-Assignment- for example).

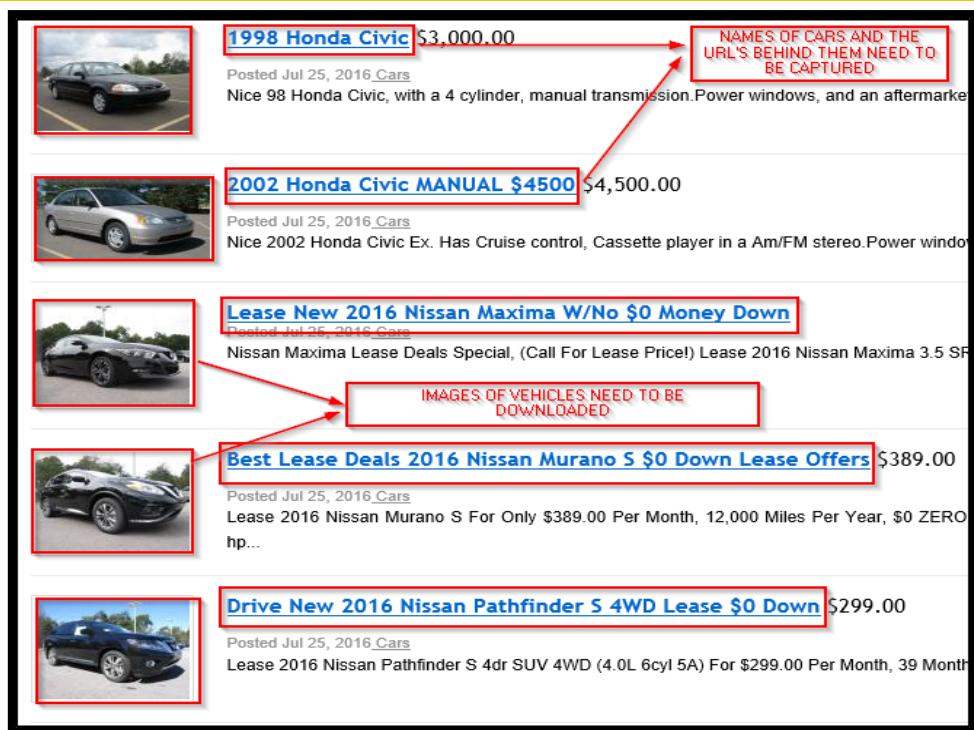
- Next click on “  Save ” and then click on “  Run ” .
- The task will execute and display the Message box.

### Example 3 (Extract Data - Pattern)

- Start Automation Anywhere Client, prior to that start your Automation Anywhere Web Control Room so that client can be registered successfully and connects with it.

- Click on “ ->” 

- Type the URL as <http://www.classifiedsgiant.com/273-cars/listings.html> where the recording is to be done and click on start.
- Note : The following information (In Red Box) needs to be captured , for the current page and by navigating forward using the Next button.



**1998 Honda Civic \$3,000.00**  
 Posted Jul 25, 2016 Cars  
 Nice 98 Honda Civic, with a 4 cylinder, manual transmission. Power windows, and an aftermark...

**2002 Honda Civic MANUAL \$4500 \$4,500.00**  
 Posted Jul 25, 2016 Cars  
 Nice 2002 Honda Civic Ex. Has Cruise control, Cassette player in a Am/FM stereo. Power window...

**Lease New 2016 Nissan Maxima W/No \$0 Money Down**  
 Posted Jul 25, 2016 Cars  
 Nissan Maxima Lease Deals Special, (Call For Lease Price!) Lease 2016 Nissan Maxima 3.5 S...

**Best Lease Deals 2016 Nissan Murano S \$0 Down Lease Offers** \$389.00  
 Posted Jul 25, 2016 Cars  
 Lease 2016 Nissan Murano S For Only \$389.00 Per Month, 12,000 Miles Per Year, \$0 ZERO...

**Drive New 2016 Nissan Pathfinder S 4WD Lease \$0 Down** \$299.00  
 Posted Jul 25, 2016 Cars  
 Lease 2016 Nissan Pathfinder S 4dr SUV 4WD (4.0L 6cyl 5A) For \$299.00 Per Month, 39 Month...

- The following warning is thrown, click on “OK” to proceed as below.



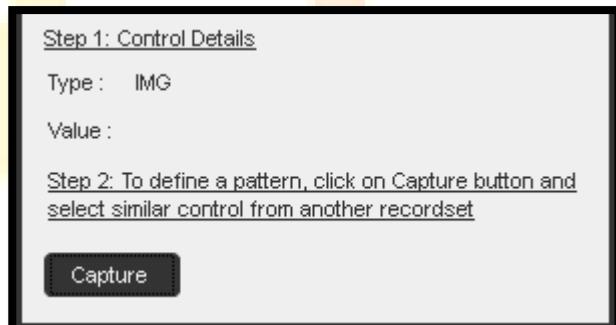
- The following Web Recorder bar appears at the bottom right corner of the screen.



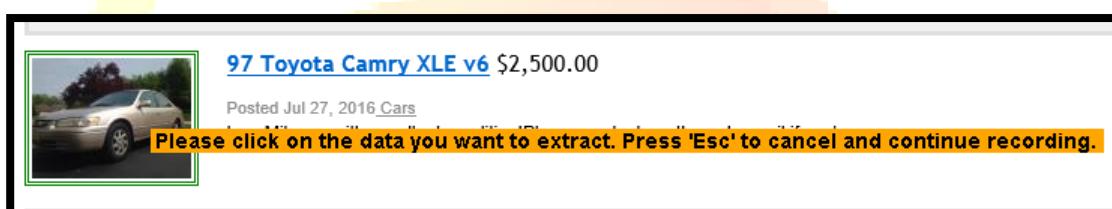
- For the sake of this demo we will try to extract information from the web page using “Extract Data” option shown in the recorder, click on “Extract Data” option from the “Web Recorder” dialog shown above.
- Choose “Pattern Based Data” and click on “Next” as shown below.



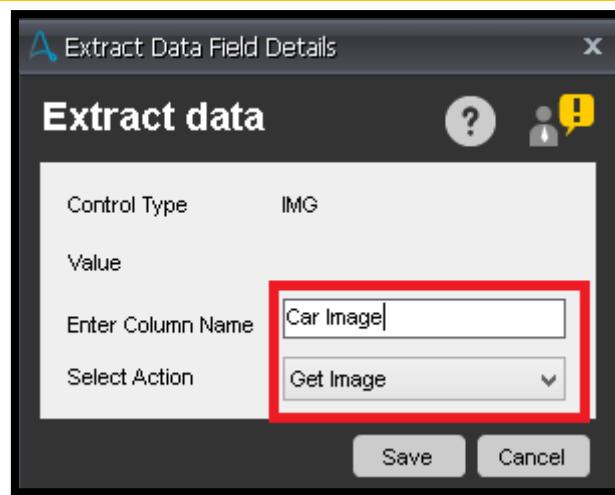
- Click as shown below.



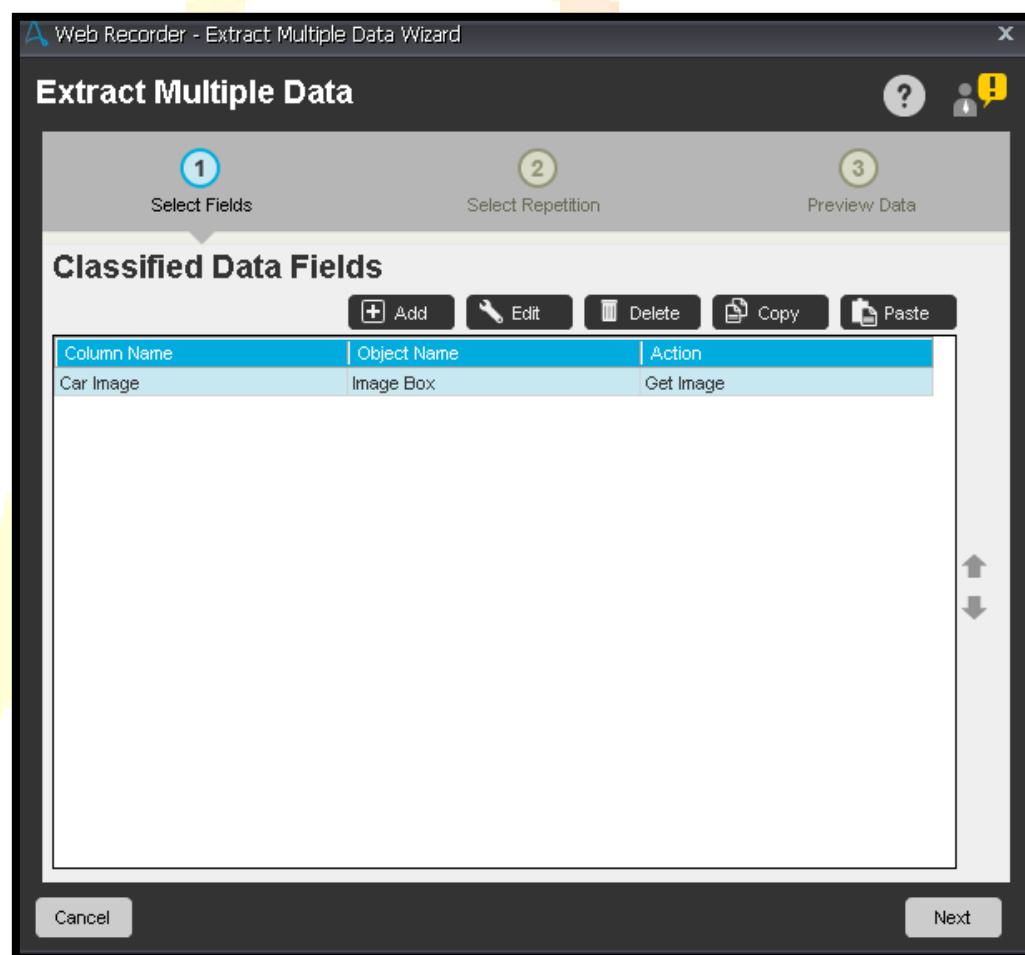
- Again click as shown below , on the next image following the above one.



- The following dialog opens, enter column name = “Car Image” and click on Save as below.



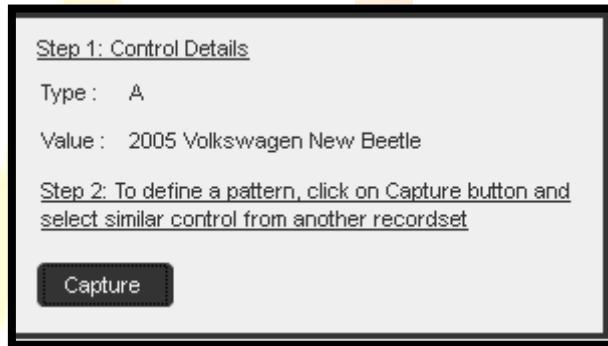
- The following dialog opens.



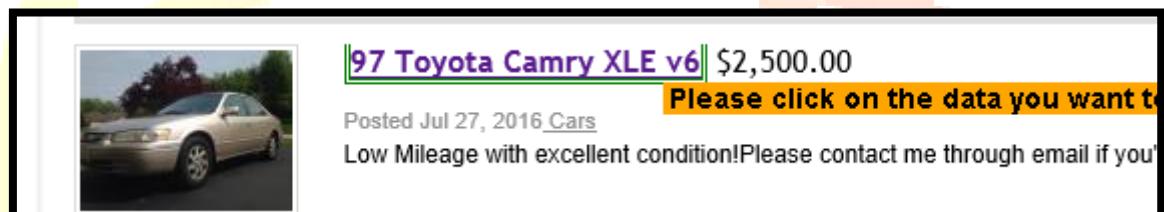
- In the above dialog, click on Add, next choose the first car name from the screen as shown below.



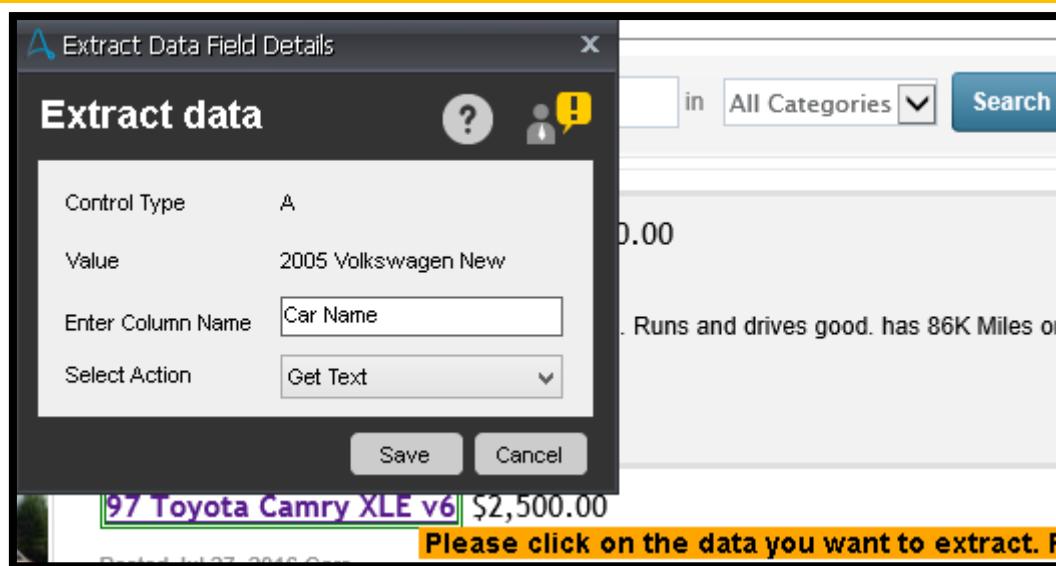
- This pops up the dialog below, click on capture .



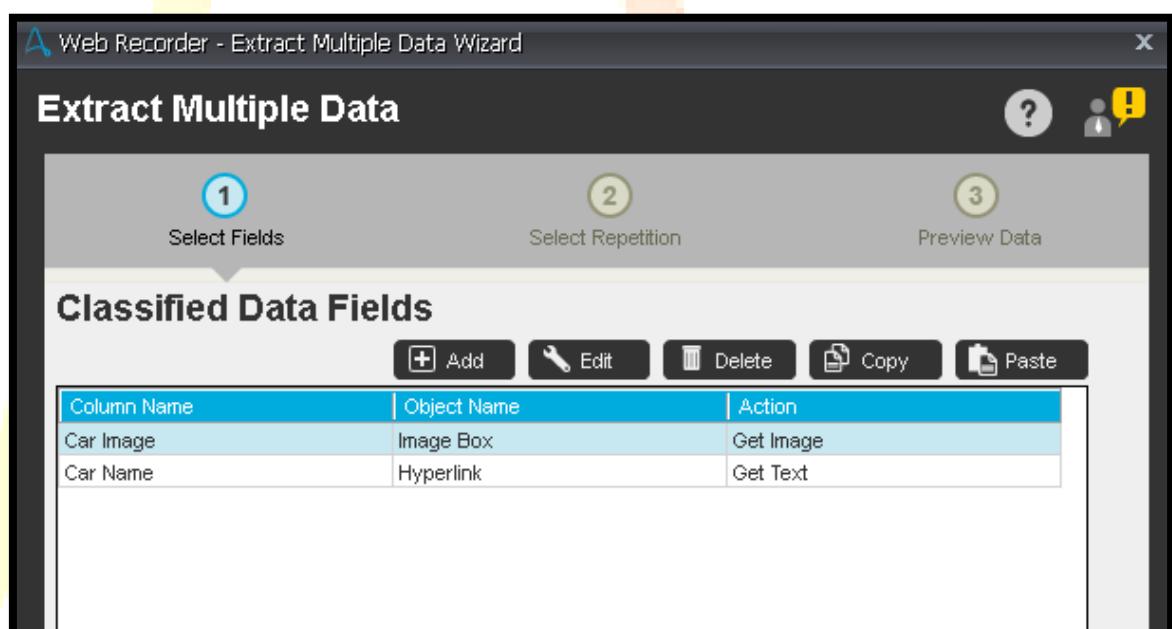
- This opens the following screen, click on the next car name immediately following the previous one.



- This opens the “Extract data” dialog below, give the column name as “Car Name” and then click on the “Save” button



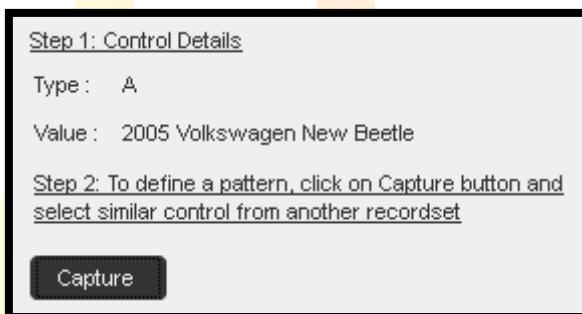
- This opens the following “Extract Multiple Data” dialog, click on the “Add” button to add another field.



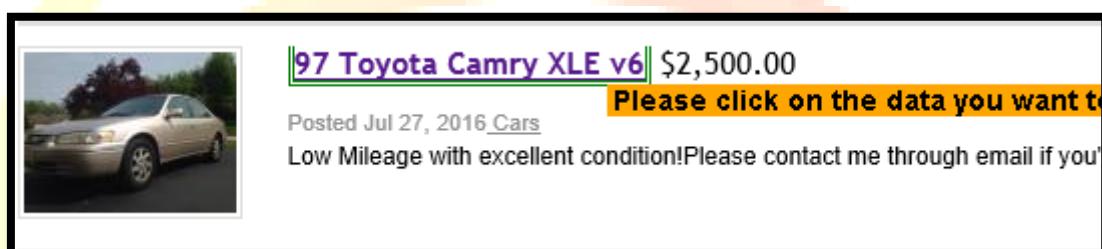
- This opens the following dialog, click on the first car on this page .



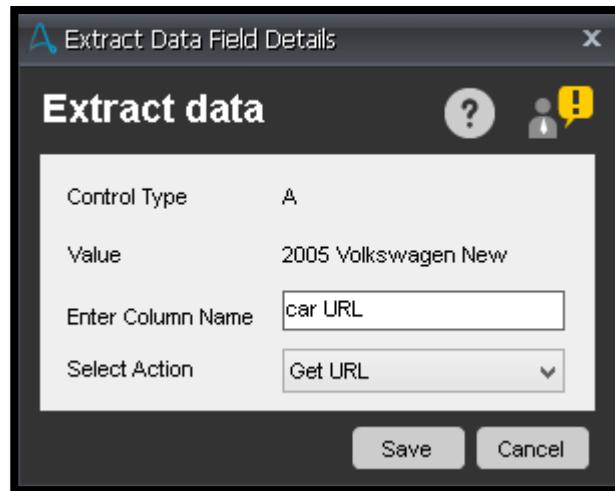
- This opens the following dialog.



- We click on the capture button as above and then click on the next car name following it as seen below :



- Next, give the column name as Car URL as can be seen below.



- List of column names that were created can be seen as below.

**Web Recorder - Extract Multiple Data Wizard**

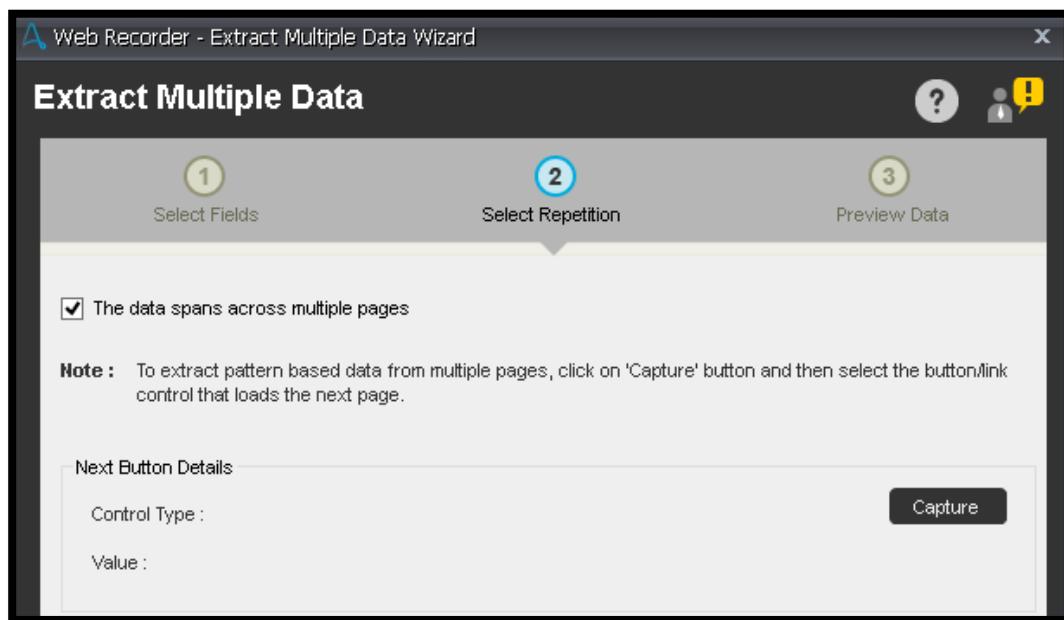
**Extract Multiple Data**

1 Select Fields      2 Select Repetition      3 Preview Data

**Classified Data Fields**

Column Name	Object Name	Action
Car Image	Image Box	Get Image
Car Name	Hyperlink	Get Text
car URL	Hyperlink	Get URL

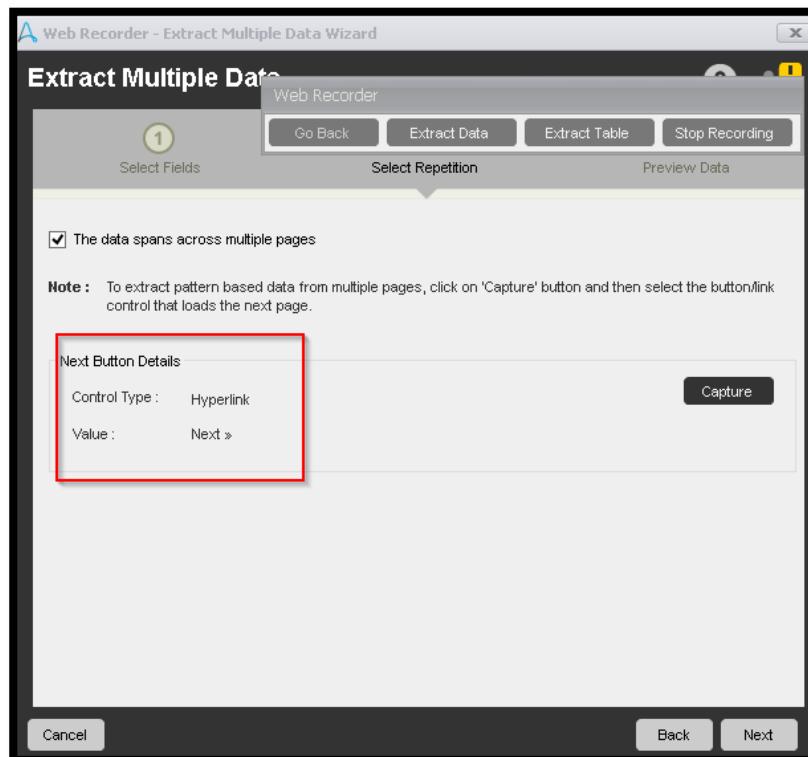
- Next, check Mark “data spans across multiple pages” and click on the capture button as below.



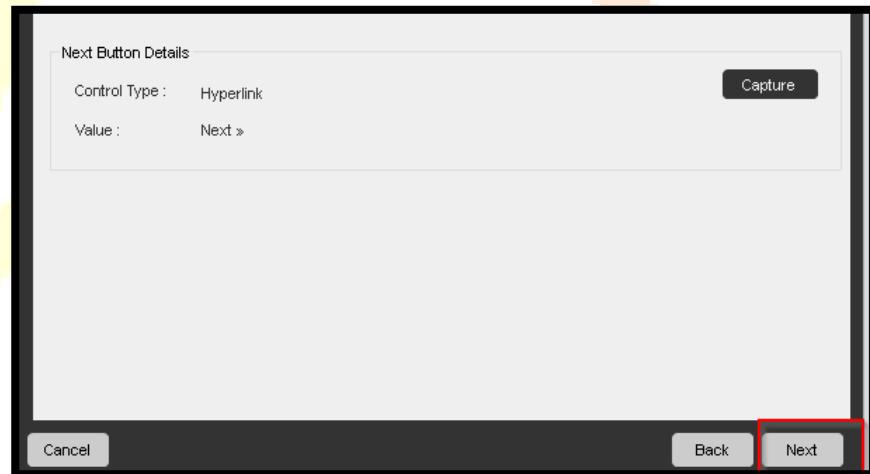
- Subsequently, scroll down in the web page and choose the “Next Button” as seen below.



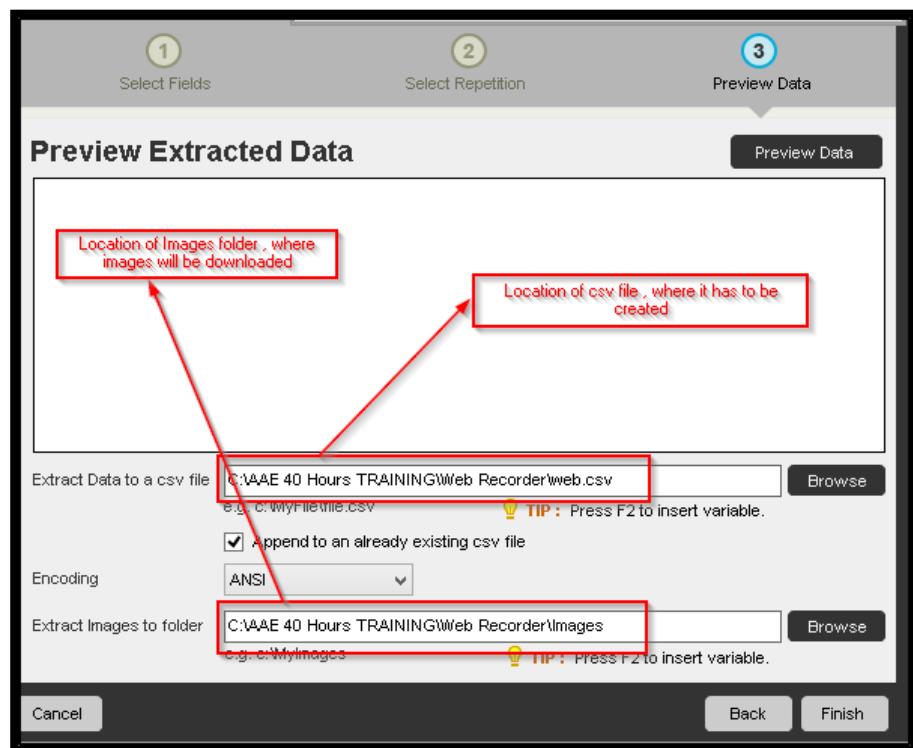
- The “Next” button is successfully captured in the Web Recorder dialog.



- Click on “Next” button in the web recorder dialog as seen below.



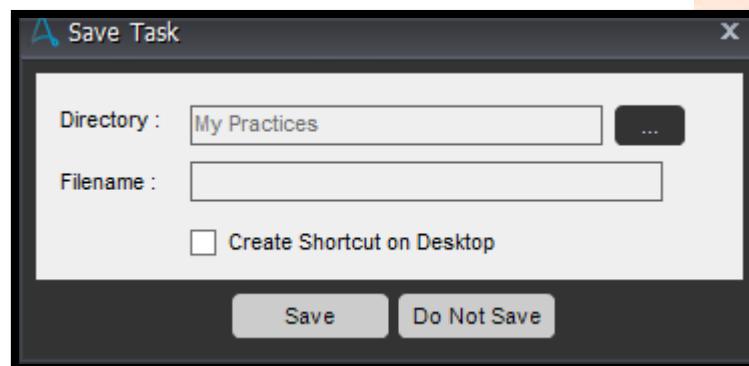
- The next dialog can be seen as below, where the CSV file in which the data is to be stored is specified along with the folder in which the images are to be downloaded is mentioned.



- Click on “Finish” in the above dialog and then choose “Stop Recording” from the web recording Toolbar as below.



- Save the recording as below, by giving it a name.



- Run the recording, see the Information getting populated in the CSV, and the Images getting downloaded in the Images folder that was specified.

A	B	C	D	E	F	G	H	I	J	K	L	M
1	Car Image	Car Name	Car URL									
2	Successfully downloaded	2002 Dodge Dakota Pickup - \$1305	<a href="http://www.classifiedsgiant.com/32795566-2002-dodge-dakota-pickup-1305/details.html">http://www.classifiedsgiant.com/32795566-2002-dodge-dakota-pickup-1305/details.html</a>									
3	Successfully downloaded	Leamington Spa Van Hire	<a href="http://www.classifiedsgiant.com/3289717-leamington-spa-van-hire/details.html">http://www.classifiedsgiant.com/3289717-leamington-spa-van-hire/details.html</a>									
4	Successfully downloaded	2002 Dodge Stratus/Repairable/Barter	<a href="http://www.classifiedsgiant.com/32899636-2002-dodge-stratus--repairable--barter/details.html">http://www.classifiedsgiant.com/32899636-2002-dodge-stratus--repairable--barter/details.html</a>									
5	Successfully downloaded	2007 Toyota Camry Good running condition	<a href="http://www.classifiedsgiant.com/3289624-2007-toyota-camry-good-running-condition/details.html">http://www.classifiedsgiant.com/3289624-2007-toyota-camry-good-running-condition/details.html</a>									
6	Successfully downloaded	2007 Toyota Camry Good running condition	<a href="http://www.classifiedsgiant.com/32899620-2007-toyota-camry-good-running-condition/details.html">http://www.classifiedsgiant.com/32899620-2007-toyota-camry-good-running-condition/details.html</a>									
7	Successfully downloaded	2007 Toyota Camry Super clean outside and ins	<a href="http://www.classifiedsgiant.com/32899597-2007-toyota-camry-super-clean-outside-and-inside/details.html">http://www.classifiedsgiant.com/32899597-2007-toyota-camry-super-clean-outside-and-inside/details.html</a>									
8	Successfully downloaded	Lease 2017 Chrysler Pacifica LX \$0 Down	<a href="http://www.classifiedsgiant.com/32899537-lease-2017-chrysler-pacifica-lx-0-down/details.html">http://www.classifiedsgiant.com/32899537-lease-2017-chrysler-pacifica-lx-0-down/details.html</a>									
9	Successfully downloaded	Lease 2016 Dodge Grand Caravan SE \$0 Down	<a href="http://www.classifiedsgiant.com/32899536-lease-2016-dodge-grand-caravan-se-0-down/details.html">http://www.classifiedsgiant.com/32899536-lease-2016-dodge-grand-caravan-se-0-down/details.html</a>									

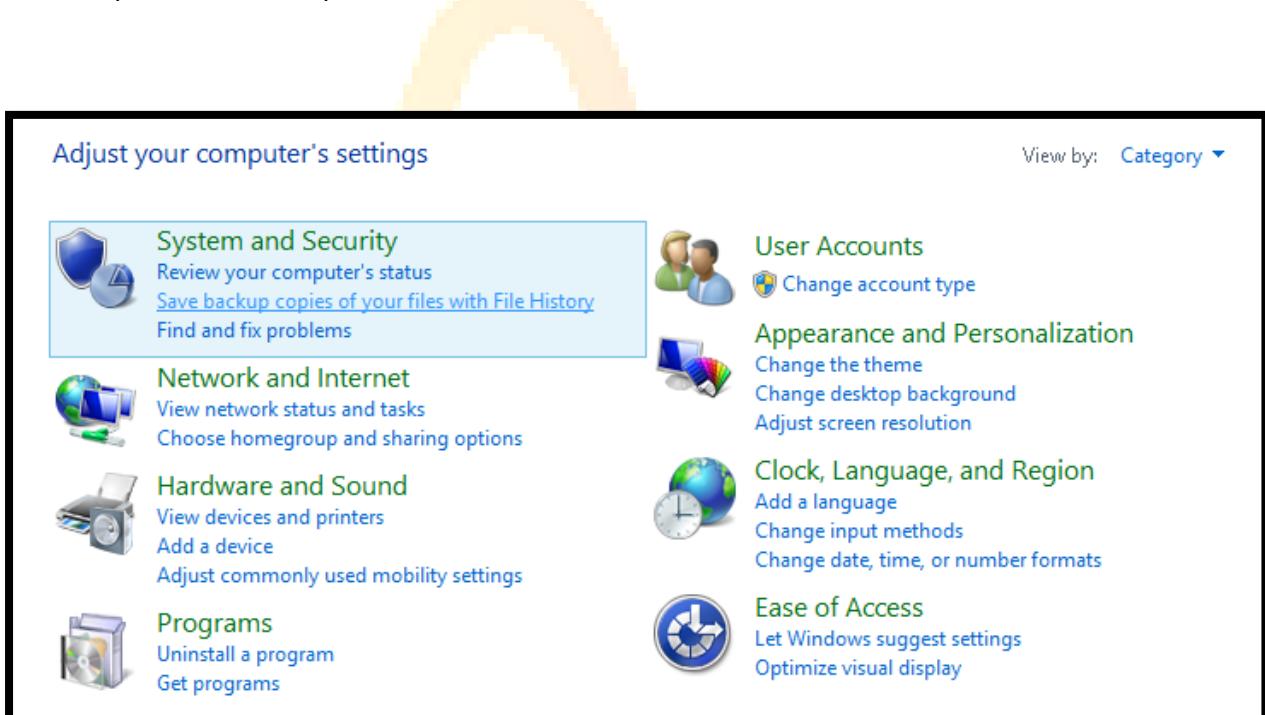
### 3) Object cloning

#### Objective

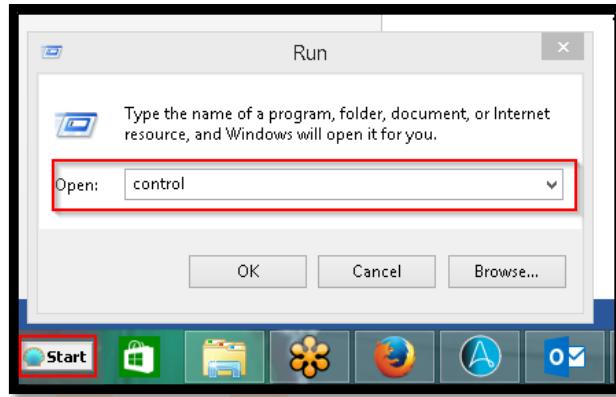
- Step by step demonstration of Object Cloning and its working mode features like Image mode, Co-ordinates mode and Object mode.

#### Example: Object Cloning a Windows Application

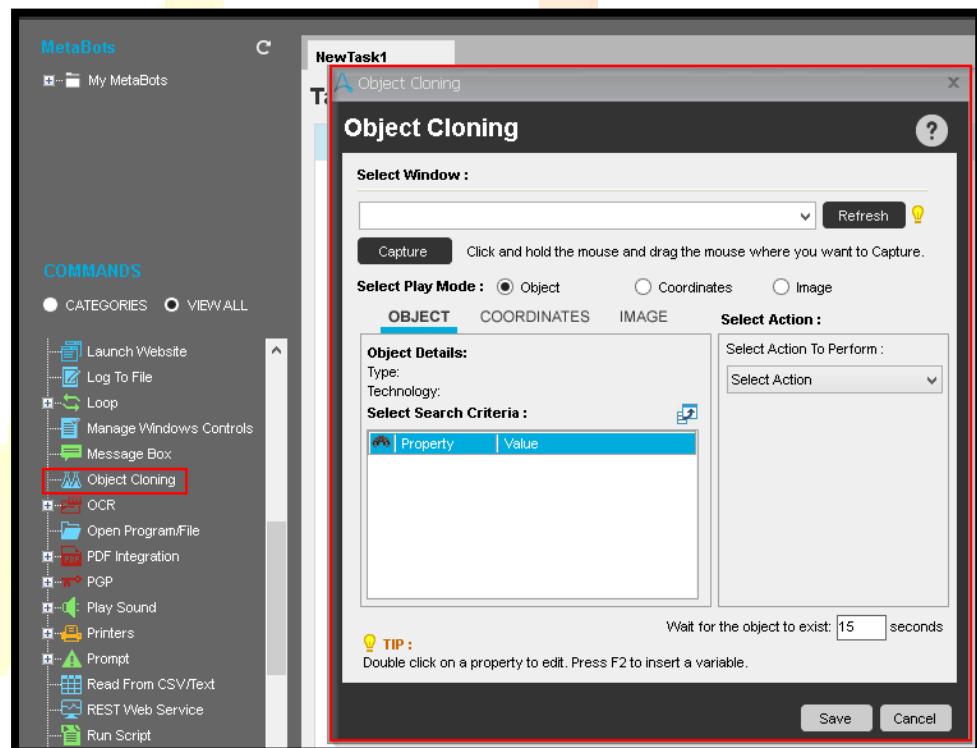
- Open the control panel as below.



- Create a simple recording to reach the control panel window or tweak "start->run "->" Control" to land there.



- Get into AA Development client and open the Task Editor.
- Now Drag an “Object Cloning” command to the right as below.



- Choose “Control Panel” in window name as below.



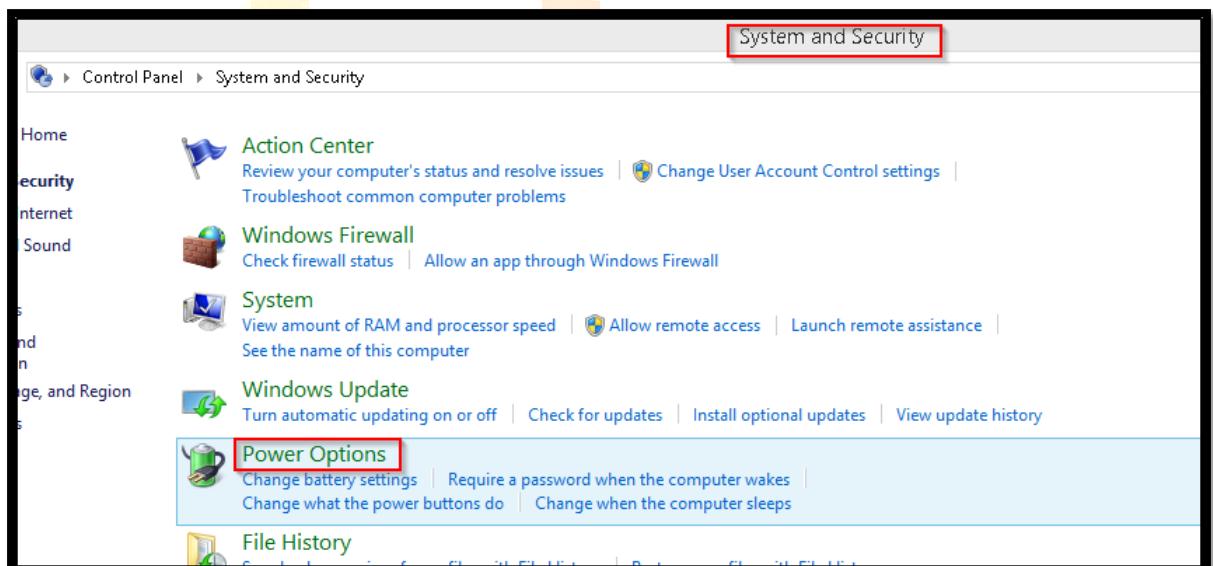
- Then on this same “Object Cloning” window, click on the “Capture” button, keep it pressed and bring it over **System and Security** and click on it.
- Now edit the recorded step, notice that how recordings have been done under all three modes “Object”, “Co-ordinates” and “Image”. Also click on the  next to “Select Search Criteria” in Object Mode, to get a better view of the captured object properties.

Property	Value
Name	System and Security
Type	Link
Path	4 1 4 2 4 1 4 1 1 5
Index	5
Left	400
Top	138
Width	145
Height	21
Description	System and SecurityView and change system an...

- Do click on the “Image” mode to ascertain that the image is clear and recognizable.
- In “Select Action to Perform”, choose “click” as below.

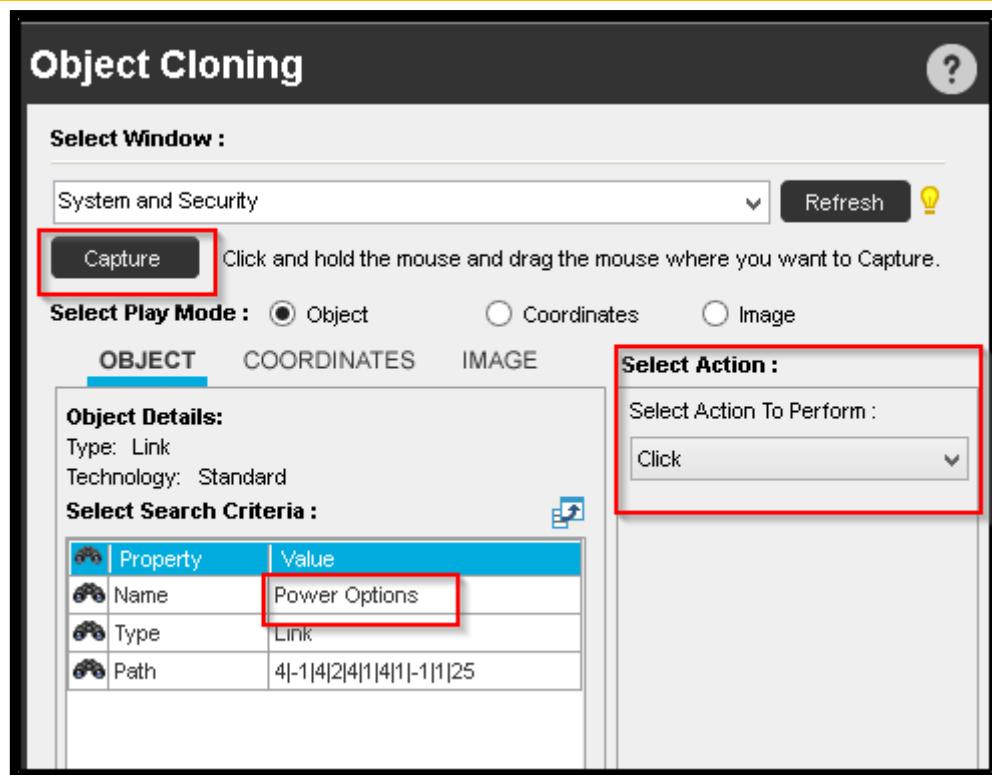


- Now go into the “Control Panel” window by yourself and click on **System and Security** so that we are now on the page as below

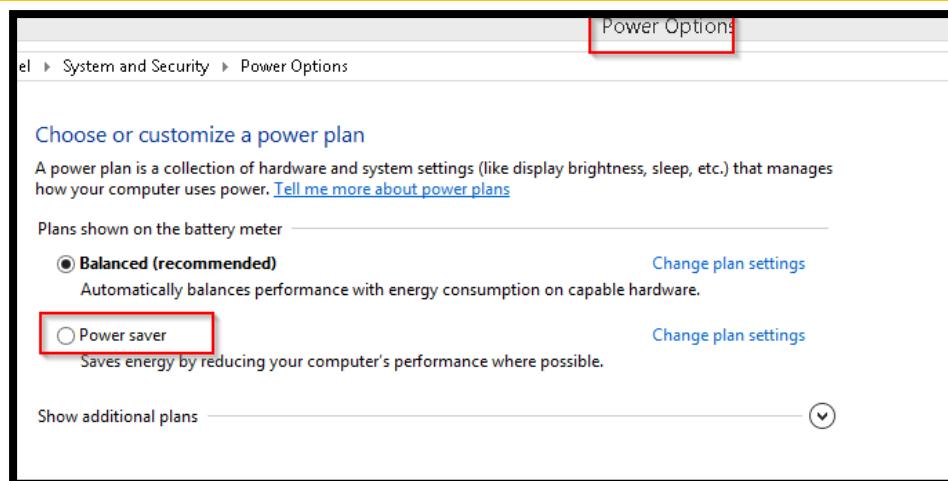


- Go back into the Task Editor in AA Dev client, drag an object cloning over again, choose “System and Security” as the window this time, then click on “Capture” and keep it pressed till you go and select “Power Options” in the “System and Security” window.



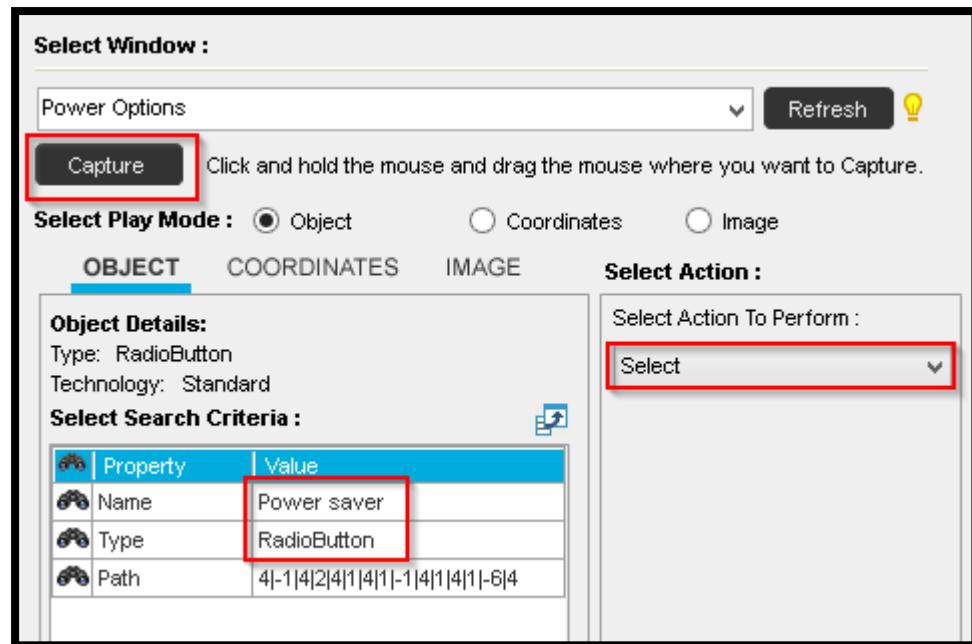


- Do click on the “Image” mode to ascertain that the image is clear and recognizable. Choose action as “Click” as shown above.
- Now manually go back to the “System and Security” window, click on “Power Options” so that we are now on this last page of our recording as below.

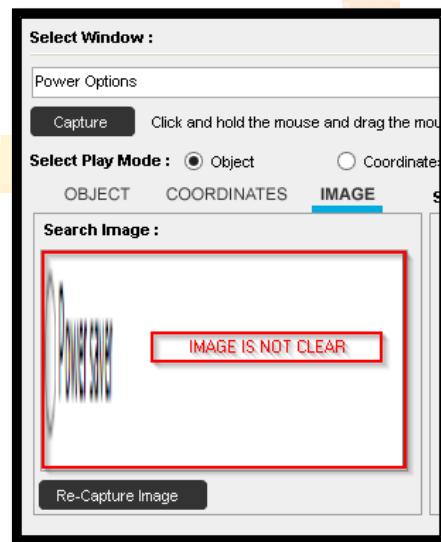


- Go back into the Task Editor in AA Dev client, drag an object cloning over again, choose “Power Options” as the window this time, then click on “Capture” and keep it pressed till you go and select “Power Saver” Radio Button in the “Power Options” window.



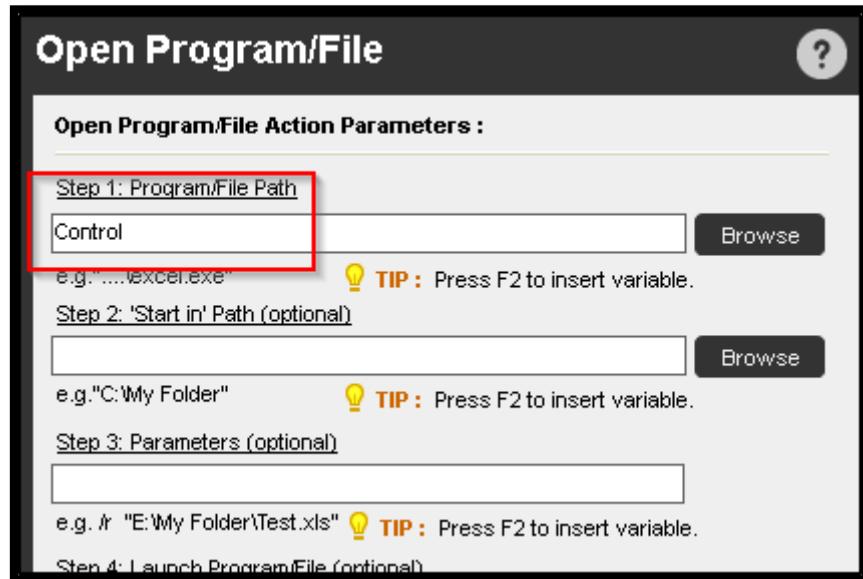


- Do click on the “Image” mode to ascertain that the image is clear and recognizable. Choose action as “Select” as shown above. If the Image is not clear as below for example.

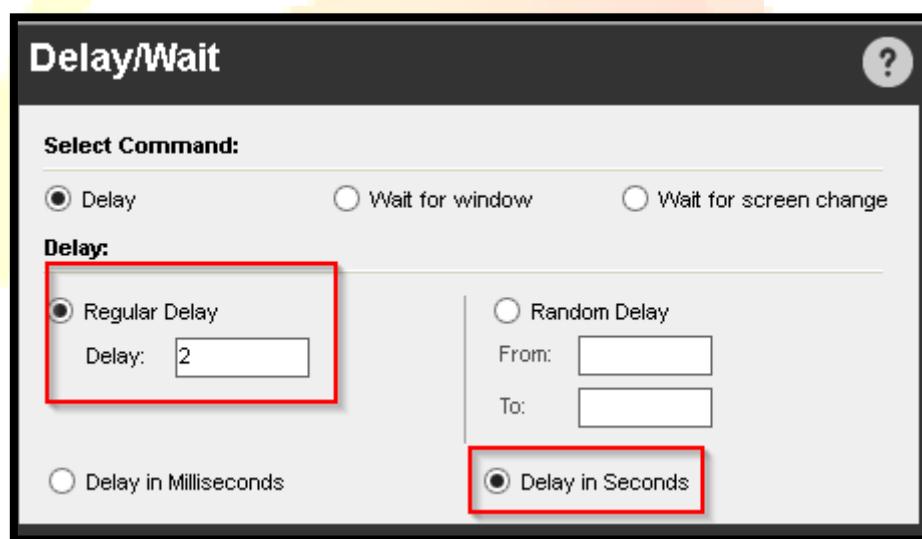


- Then click on the “Re-Capture Image” button and capture the Image again.

- Drag an “Open Program/File” command to the top right of the script and make following changes as below



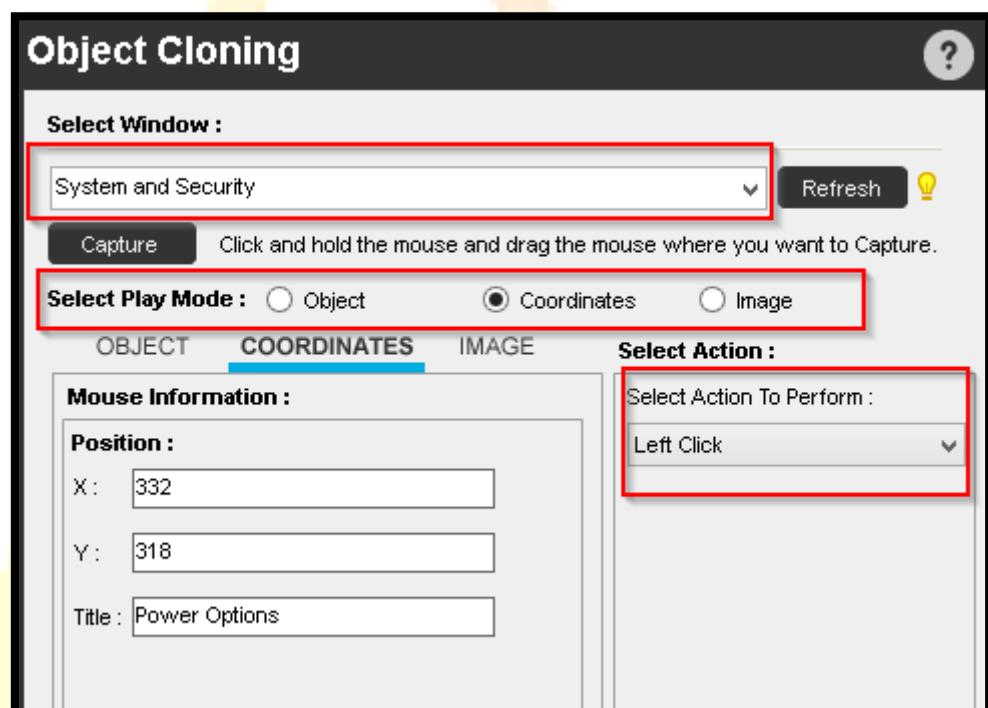
- This is to launch the Control Panel window.
- Next Drag a “Delay command” and keep a delay of 2 seconds as below.



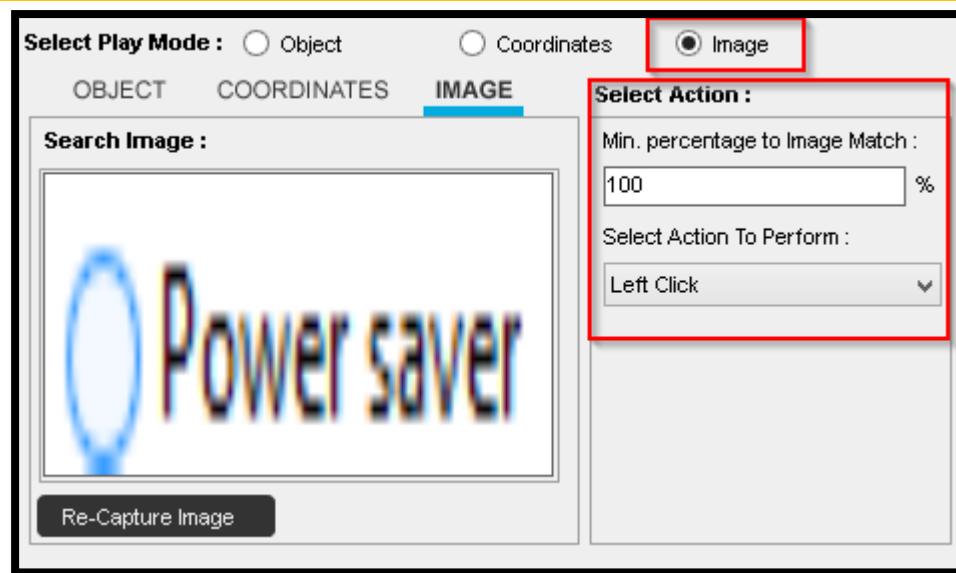
- Position as below



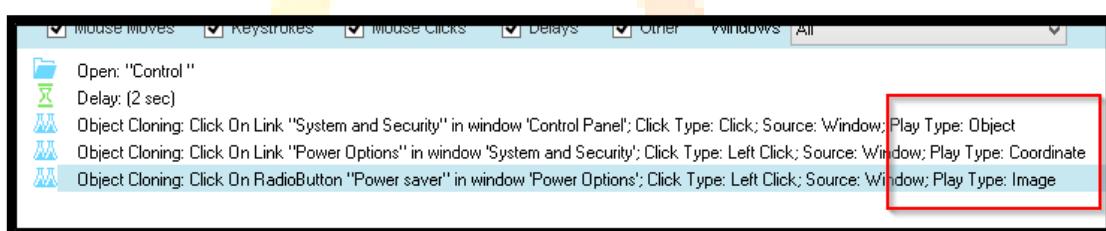
- Run the script. We should get success.
- Next we will edit the second “Object Cloning” command in our script and change play mode to Co-ordinate as below



- Next we will open the third Object Cloning command in the script and change its play mode to Image as below



- Our Script should look like as below



- As before the script can get all the jobs done successfully.

## 4) Excel Commands

### Objective

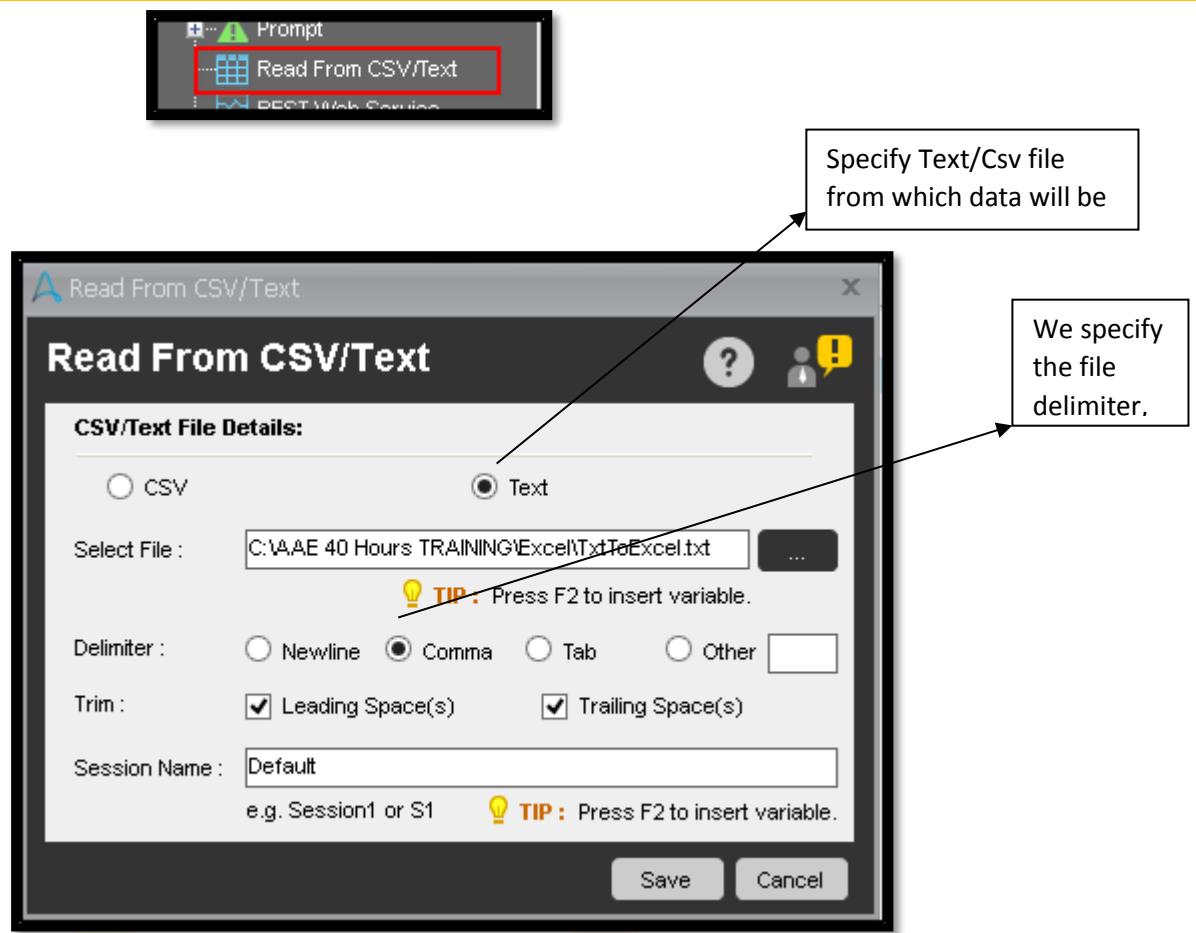
- Step by step demonstration of Excel commands for example reading and writing data to an excel sheet along with various other features like executing a macro and combining it with various other commands for example Database Command.

### Example 1: Reading Data from a CSV/TXT and writing into an Excel spreadsheet

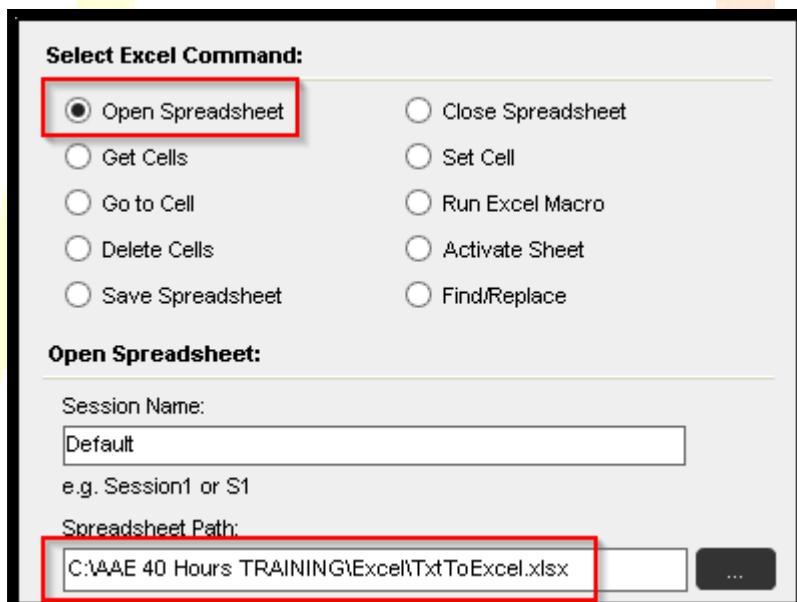
- We create a TXT file with the following data as seen below

Name, Age
ABC, 10
Def, 20
Ghi, 30
Jkl, 40
Mno, 50

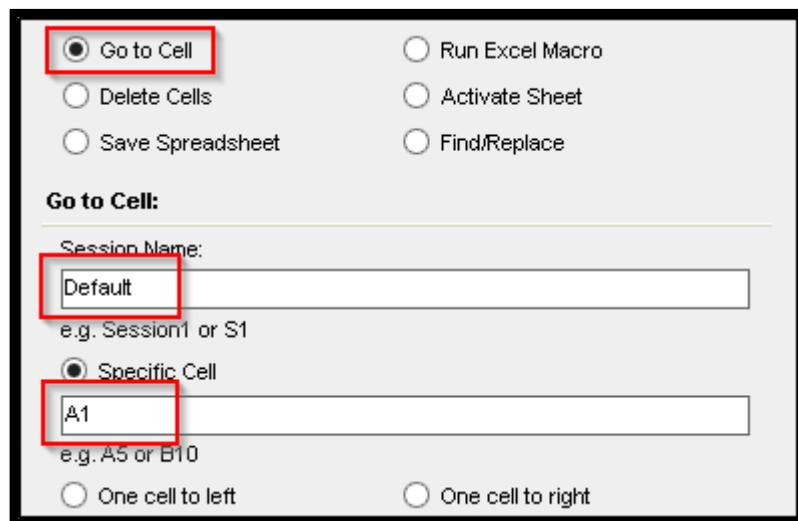
- We can also create a blank .xlsx file and save it in advance.
- Open the Automation Anywhere Client as below: -
- Next we need to open the Task Editor.
- We click on “New -> Task Editor” to open the Task Editor.
- We choose “Read from CSV/Text” as below from the commands in the Task Editor as below.



- We then choose “Excel – Open Spreadsheet “as below.

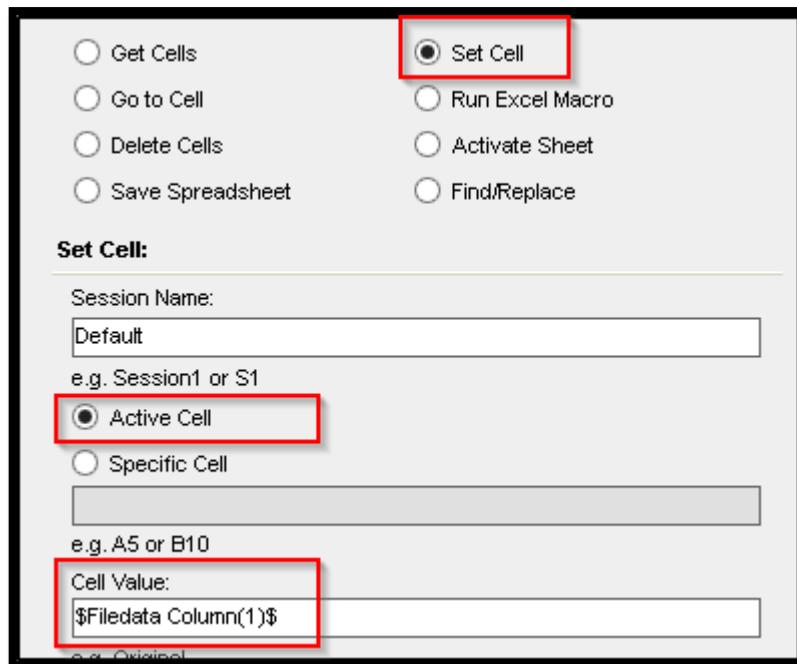


- As seen above we are specifying our excel file name where we have to write in after reading from the csv file , this is the same blank excel file that we created earlier .
- Next we want to position to cell “A1“ in excel , so that the write operation in our excel sheet takes from that point , we choose “Excel – Goto Cell” command for the same, as below

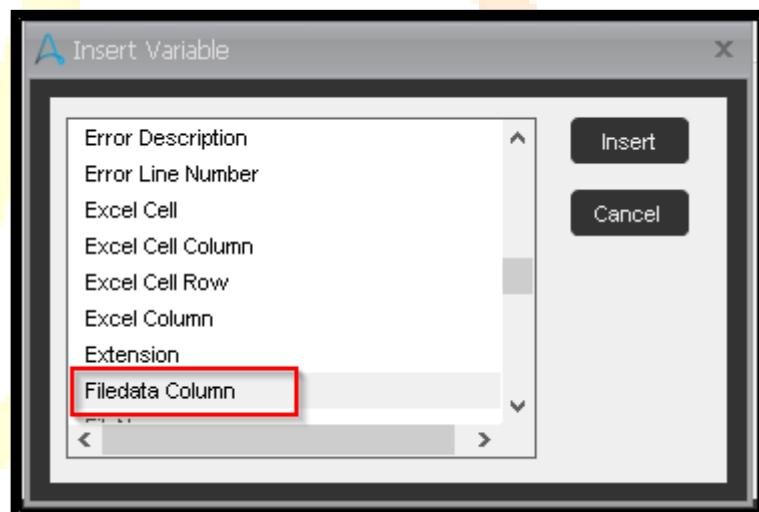


- Next we get inside the loop and start writing into the spreadsheet using “Excel -> Set Cell” as below.

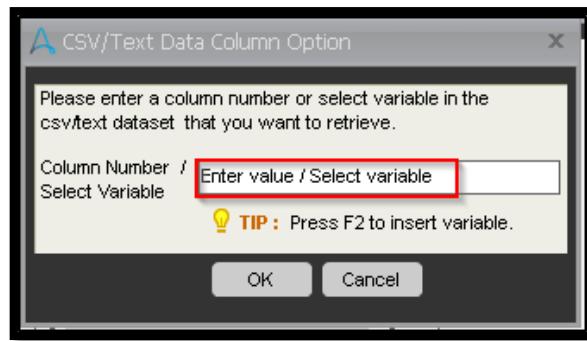




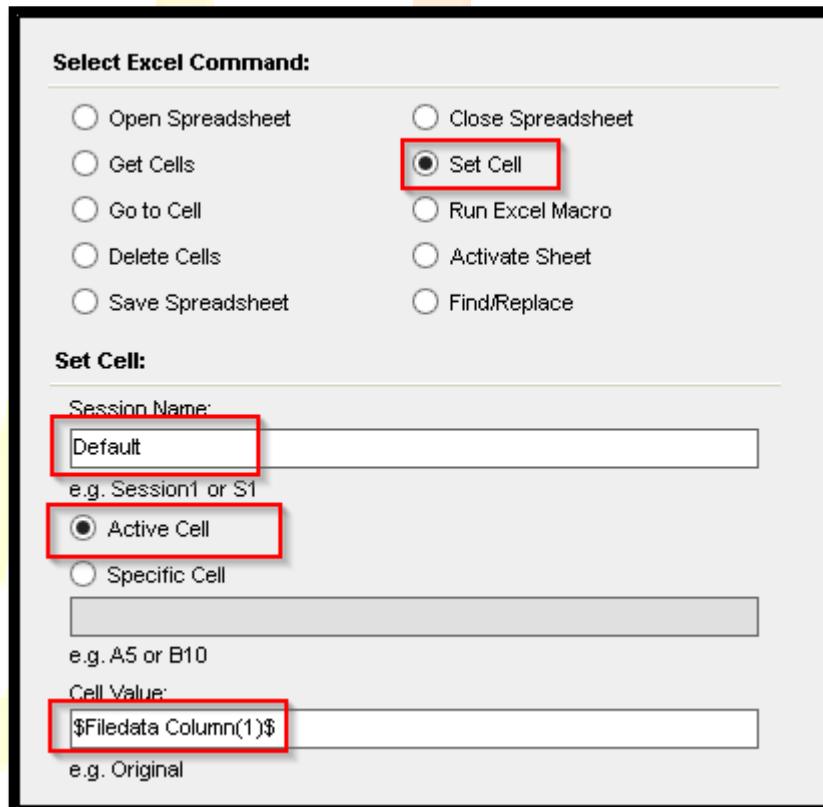
- Pressing F2 will popup the following dialog as below :



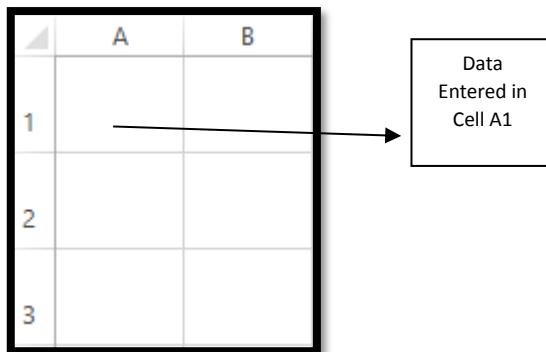
- We choose “Filedata Column” variable as seen above and click on insert which opens another dialog as below , where we have to insert the column no as below



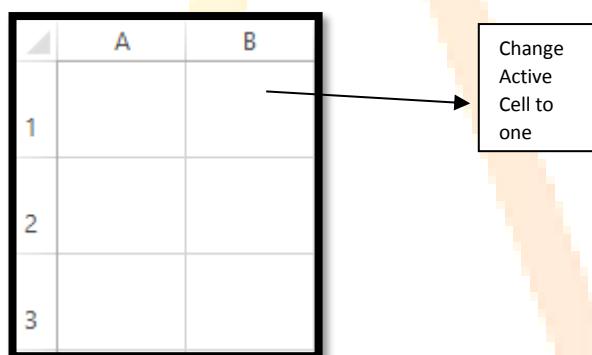
- For the first column , we will enter value as “1” , so that the final dialog looks like below



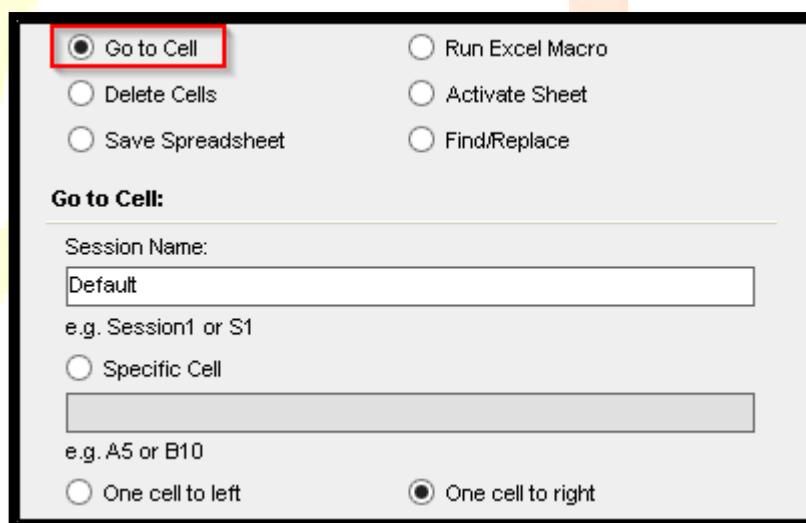
- This will enter data as below , in our excel file



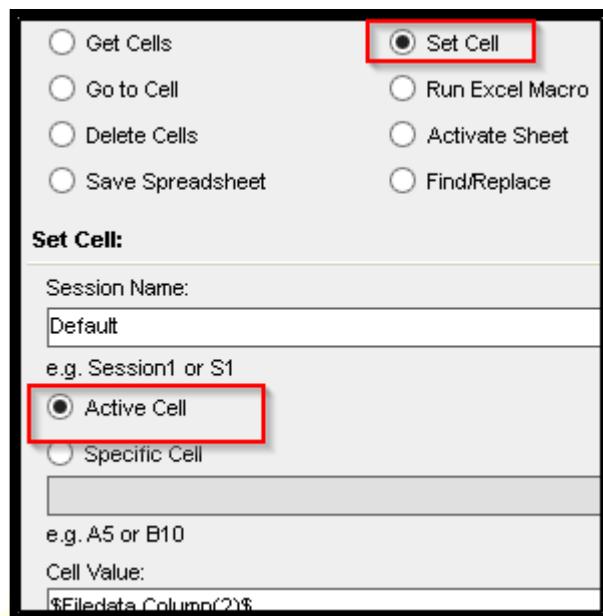
- Next , we want data to be entered in cell B1 , for that Active cell has to be made one right of the current cell as below



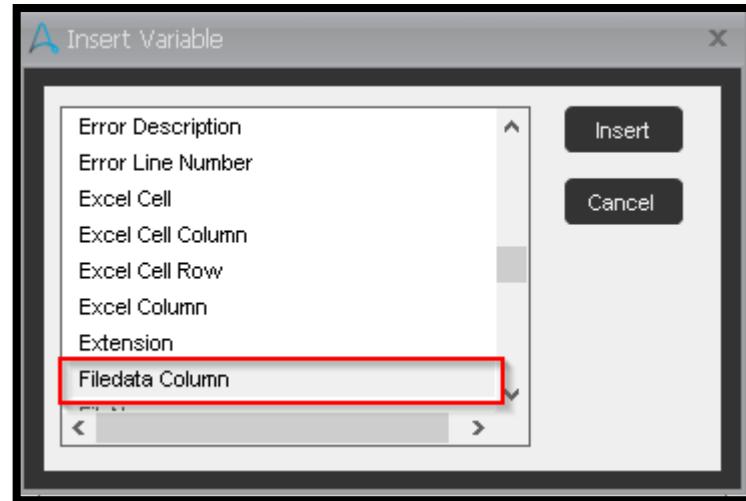
- For the above , we will use “Excel -> Goto Cell” as below



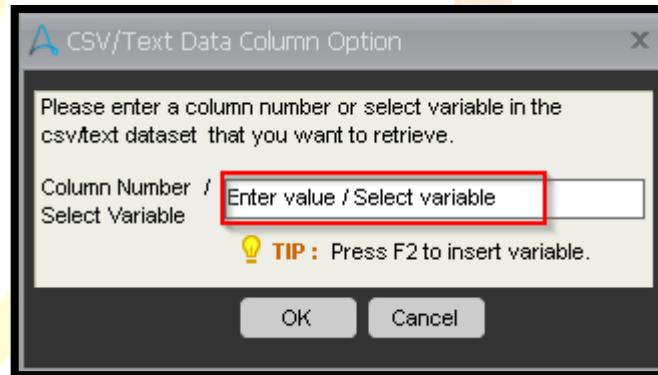
- The above command will position to the next column and we are good to assign value to this new cell as below using “Excel – Set Cell” as below



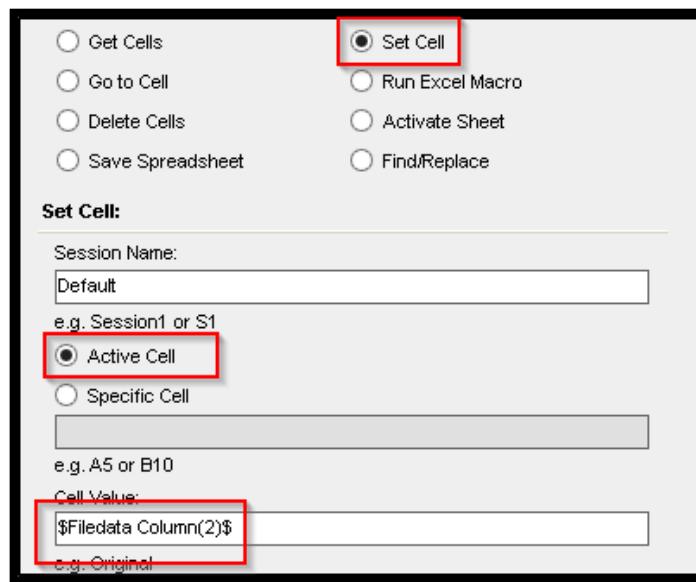
- Pressing F2 will popup the following dialog as below :



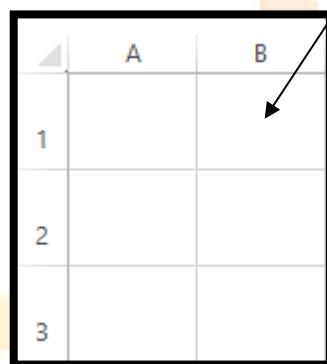
- We choose “Filedate Column” variable as seen above and click on insert which opens another dialog as below , where we have to insert the column no as below



- We will choose 2 this time, as we need to write the value of the second column, so the final dialog looks as below:

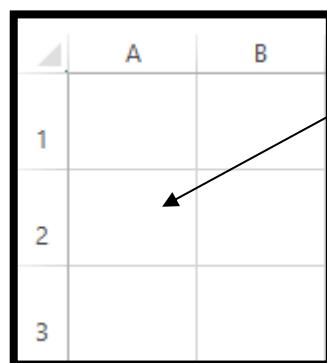


- Value is now entered in B1



	A	B
1		
2		
3		

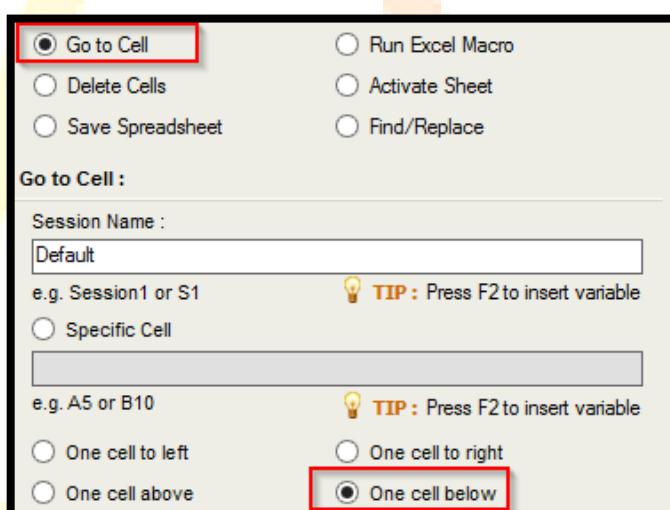
- So, the first row from the Text file is successfully written into the first row of our Excel sheet, now for the next write operation the active cell must be A2 as below



	A	B
1		
2		
3		

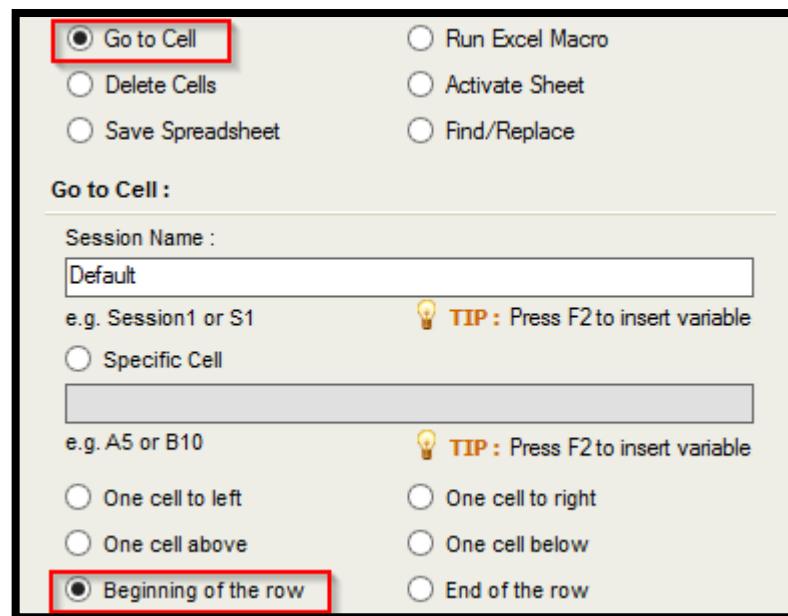
- This will be achieved using two “Excel – Goto Cell” commands , the first will move the current cell one down

	A	B
1		
2		
3		

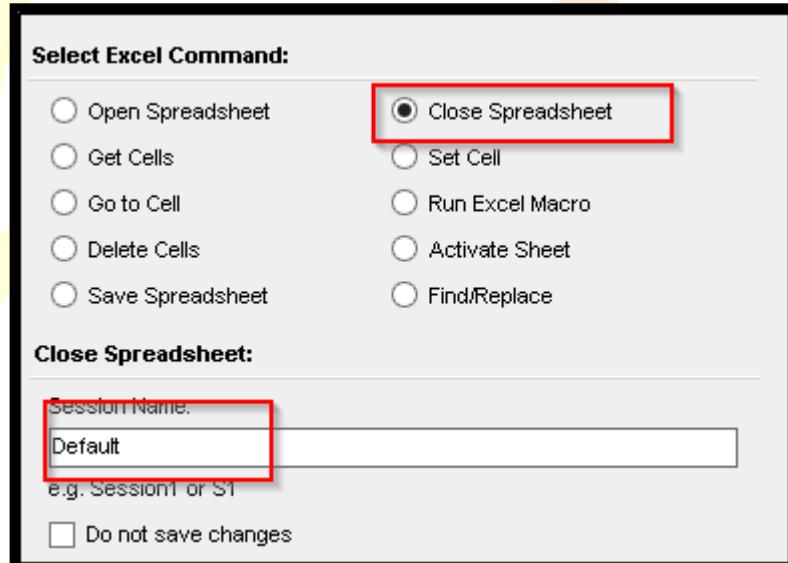


- The Next “Excel – Goto Cell” will move it to the beginning of the row , in this case to A2 as seen below

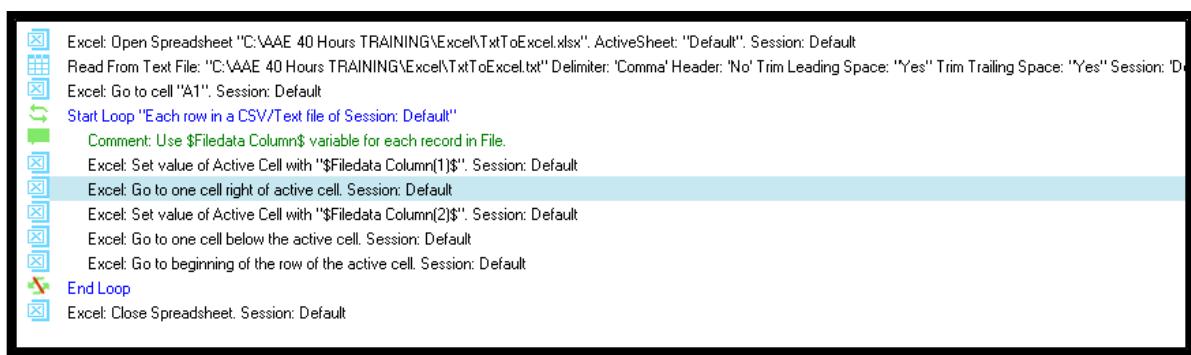
	A	B
1		
2		
3		



- This will take care of moving and writing in the spreadsheet.
- Last but not the least we need to close the spreadsheet using “Excel – Close Spreadsheet”



- Our Script Looks like as below



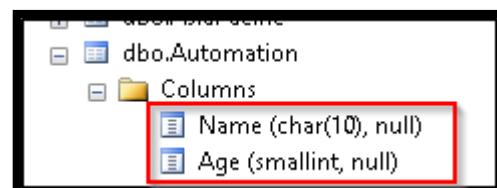
- We choose “Save Task” and then we choose “Run Task”
- The Task executes and our excel is populated with data from our CSV/TXT file .

### Example 2: Reading Data from an Excel File and Writing into a Database

- We first create an excel file and type some data into it as below

Name	Age
abc	10
def	20
ghi	30
jkl	40
mno	50

- Also, we create a blank database, create a table as below.

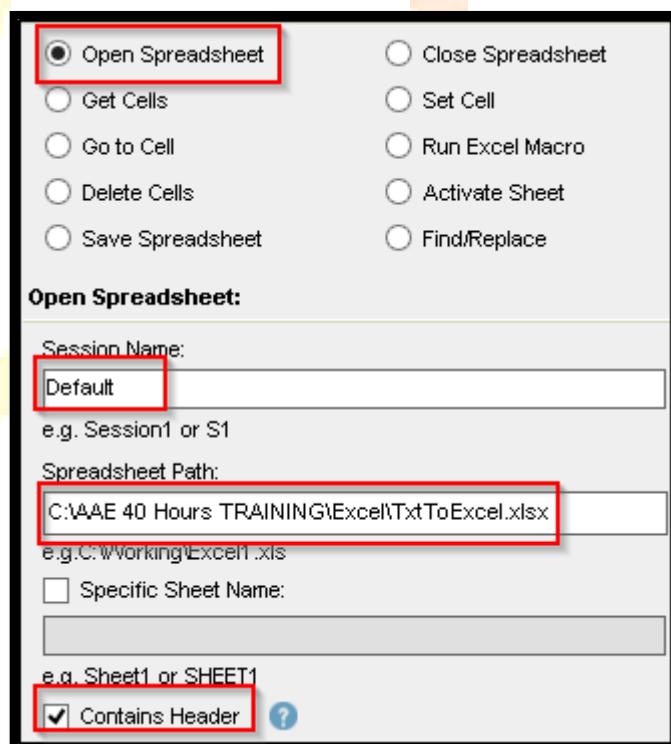


- This table has two fields “Name” and “Age” as seen above .

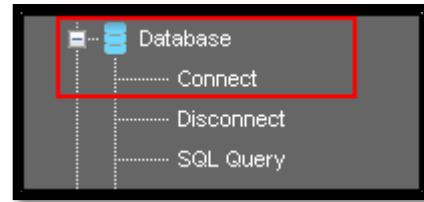
- We choose “Excel – Open Spreadsheet” command to open the spreadsheet and start reading from it as below.



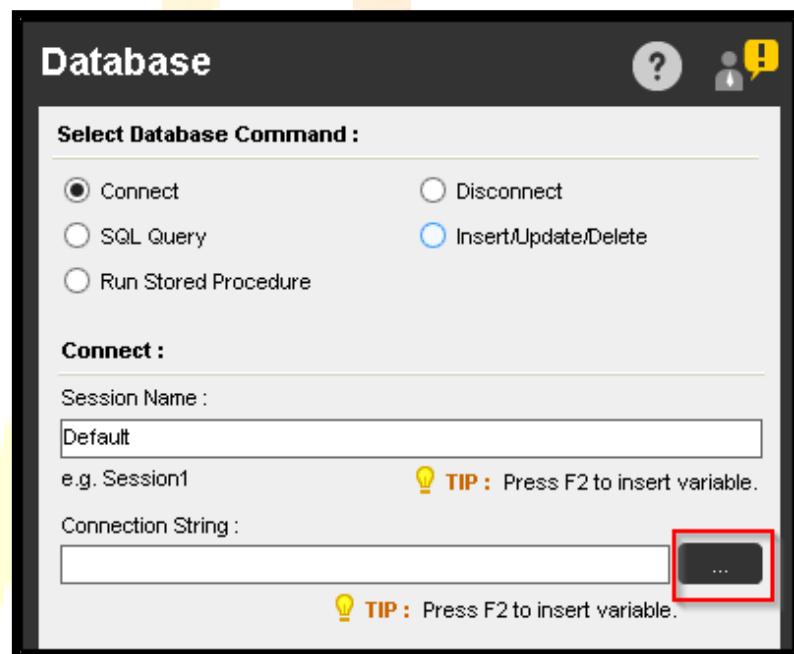
- The following dialog opens where we specify the path to our excel file which we have to read as below



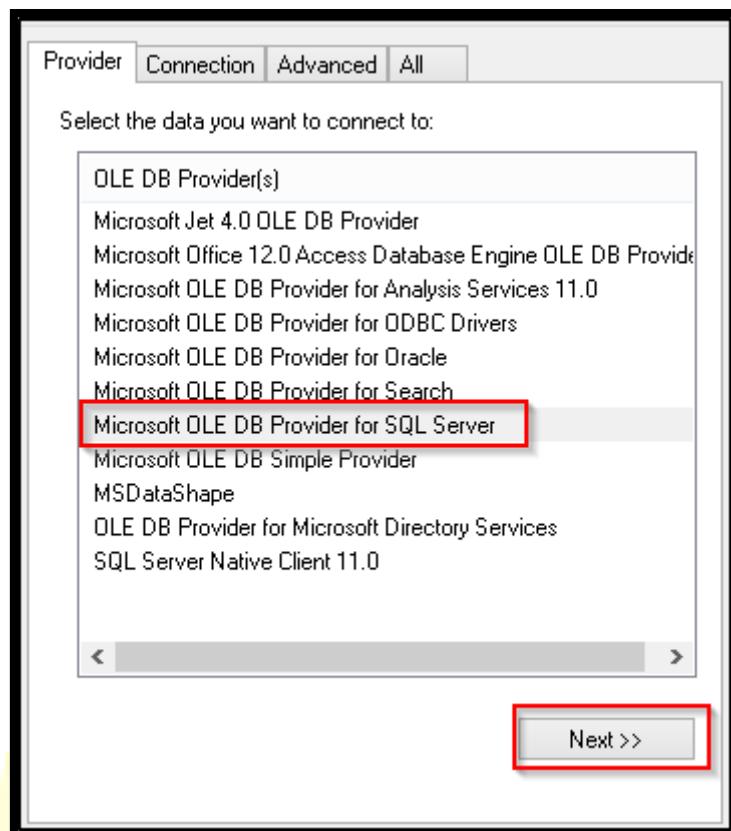
- Next we choose “Database – Connect” command as below :



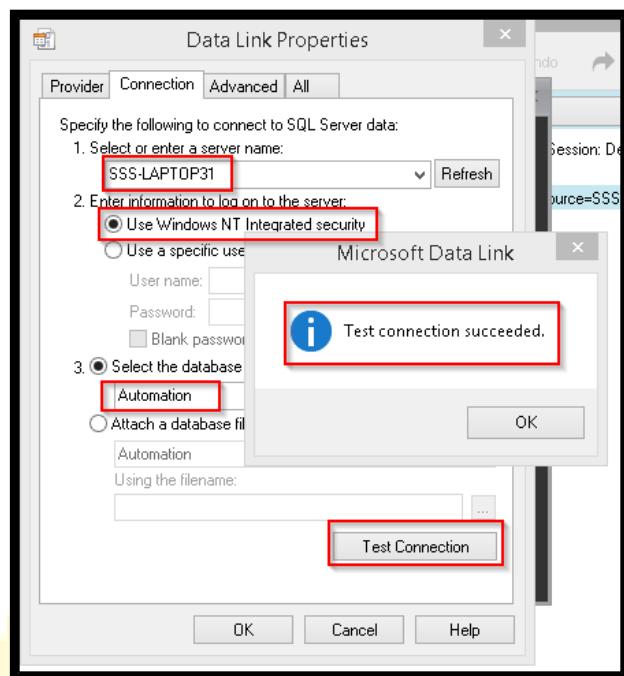
- The following dialog opens



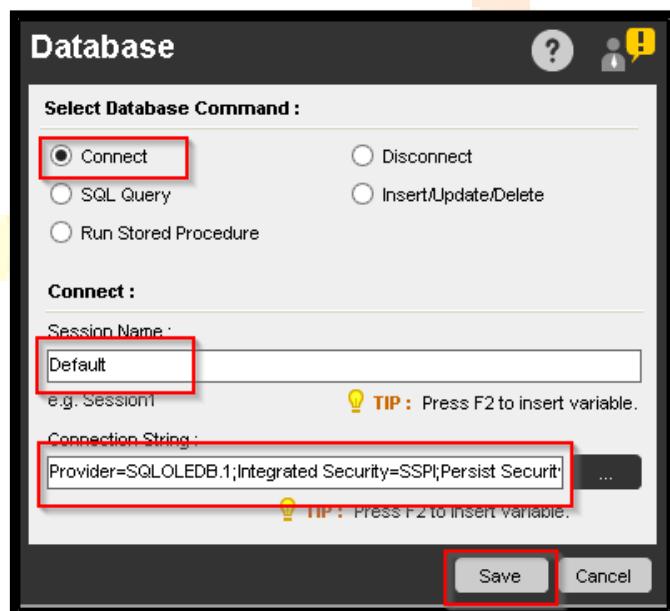
- This opens the following window as below :



- We give details of our server , and click on “Test Connection” , we should get “Test Connection Succeeded” as below

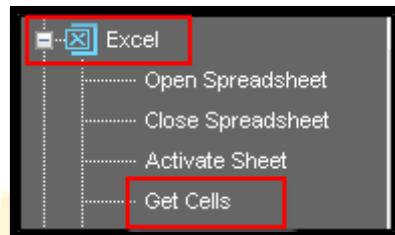


- Next , we click on “Save” as below

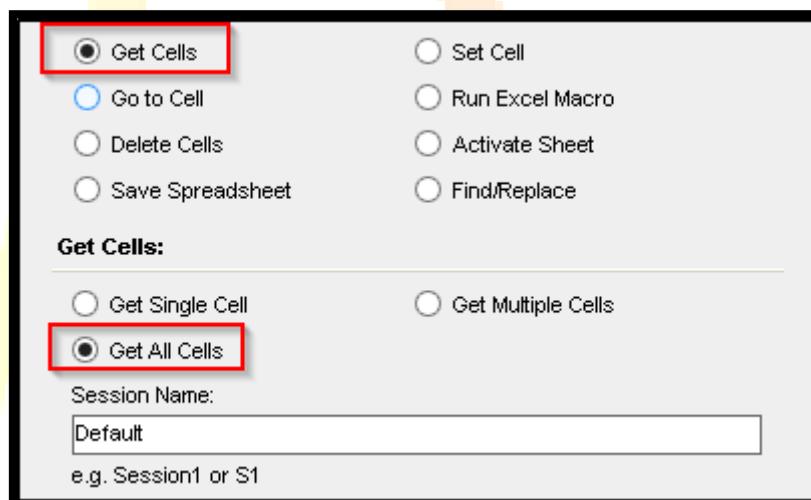


- The database connection is created .

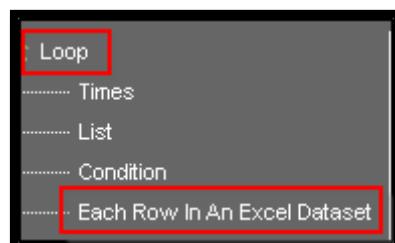
- Next , we want to read all the information from the excel sheet , for which we will use the “Excel – Get Cells” command as below :



- Then we will choose “Get All Cells” option as seen below :



- Next , we want to loop through the cache of records created in the above step , we will pick up “Loop – Each Row in a Excel dataset” as below



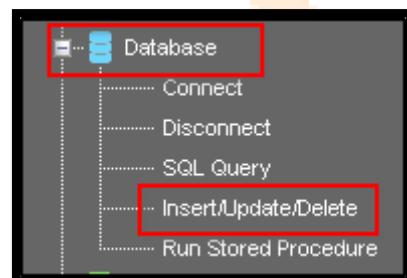
- The following Loop construct gets created and added in our Task Editor window as below :

```

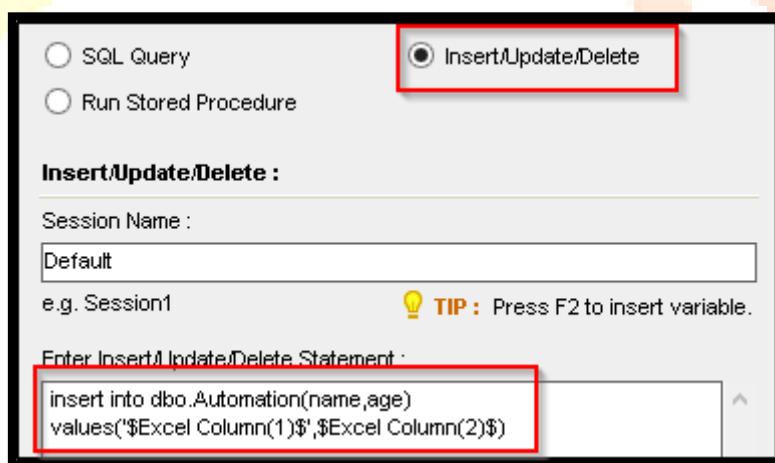
10  ↲ Start Loop "Each row in SQL Data Set Session: Default"
11  [green bar] Comment: Please enter your commands to loop. Use $Dataset Column$ variable for each record in Dataset.
12  ✎ End Loop

```

- Next , we want to bring in a “Database – Insert/Update/Delete” statement , to insert the values that we have read from the database within the above loop ,



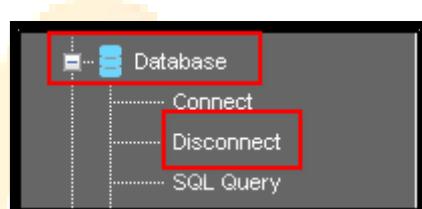
- Next , we modify our “Insert/Update/Delete” statement , to allow for insertion of records in the underlying table as below :



- Next, we call the “Excel – Close Spreadsheet” command , followed



- This, we follow with the “Database – Disconnect” statement as below



- Our script looks like as below :

```

1  Excel: Open Spreadsheet "C:\VAAE 40 Hours TRAINING\Excel\TxtToExcel.xlsx". ActiveSheet: "Default". Contains Header. Session: Default
2  Excel: Get All Cells Session: Default
3  Database: Connect to "Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=ernestyoung;Data Source=SSS-LAPTOP31" Session:'Default'
4  Start Loop "Each row in an Excel dataset of Session: Default"
5    Execute SQL Statement: "insert into dbo.ernestyoung(name,age)values('$Excel Column(1$', '$Excel Column(2$') Session: 'Default'
6  End Loop
7  Excel: Close Spreadsheet. Session: Default
8  Database: Disconnect from database Session:'Default'
```

- Save and run the script , it should read data from Excel sheet and write into the data table within our database.

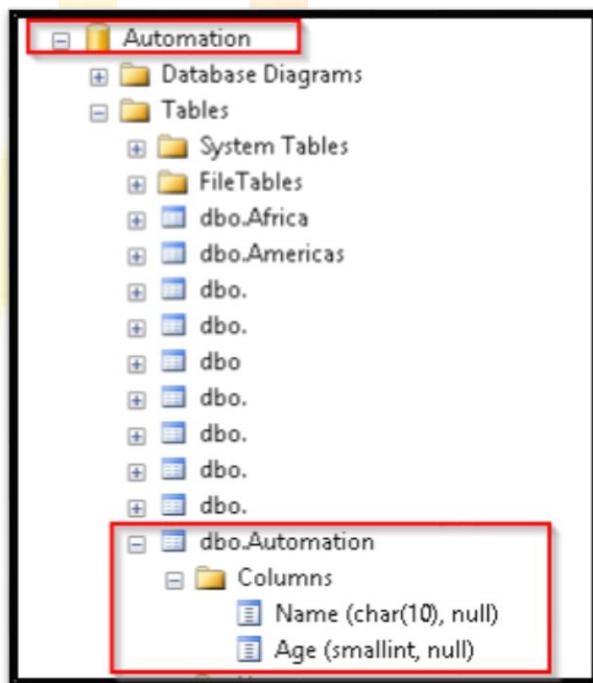
## 5) Database

### Objective

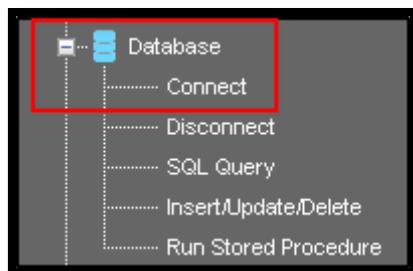
- Step by step demonstration of Database commands, for example inserting data into a database and read data from a database.

### Example 1: Writing into a Database

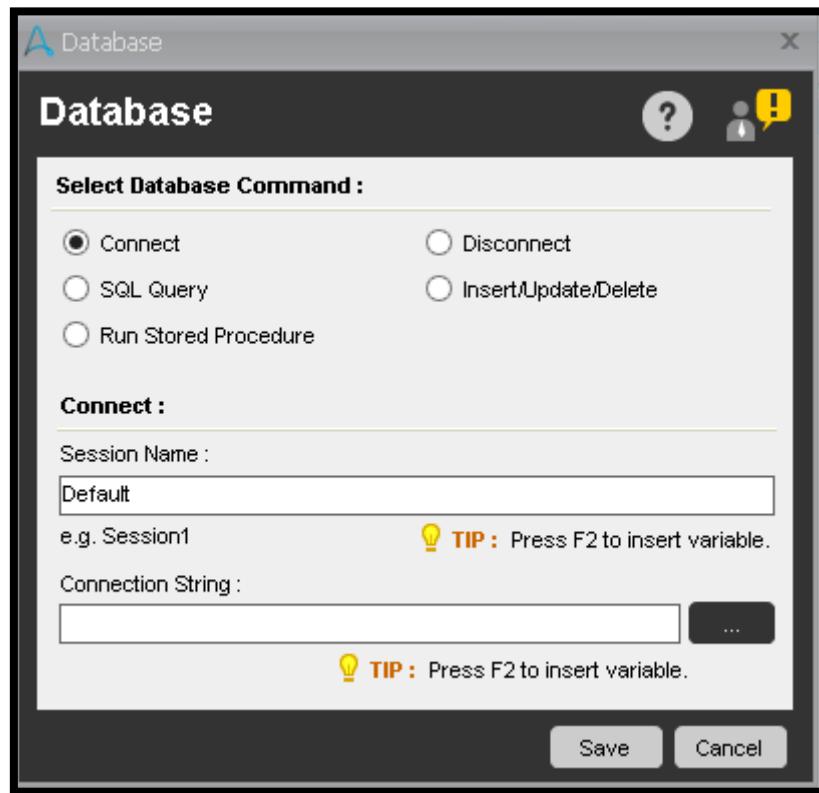
- We have already created a database, which has a table as below. Two columns Name, Age as seen below are created in the same.



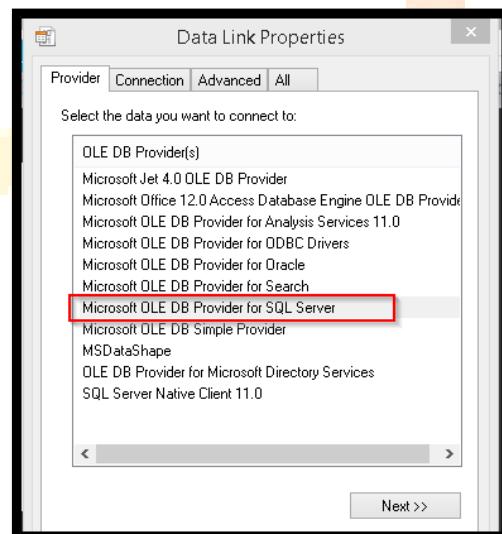
- Open the Automation Anywhere Client
- We need to open the Task Editor next.
- Once we have opened the Task Editor from New -> Task Editor
- Next choose Database -> Connect command.



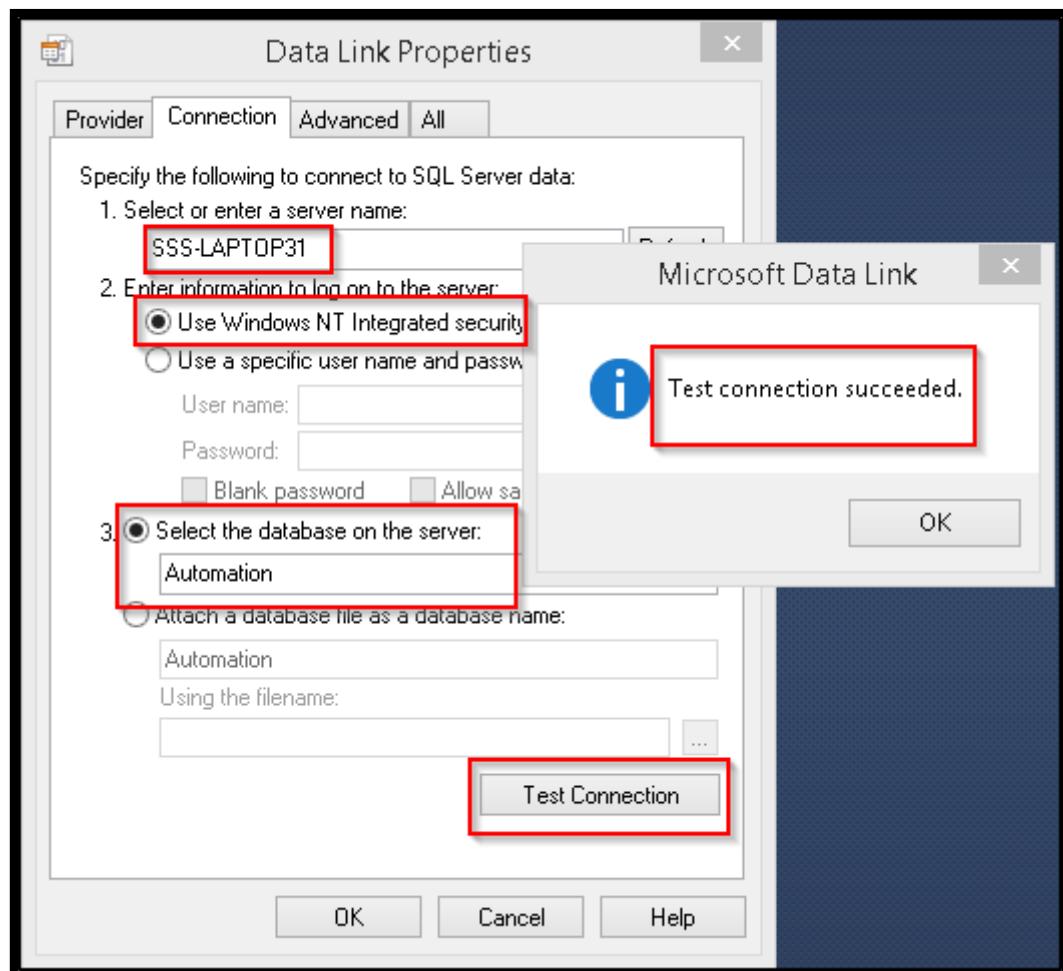
- The following dialog opens



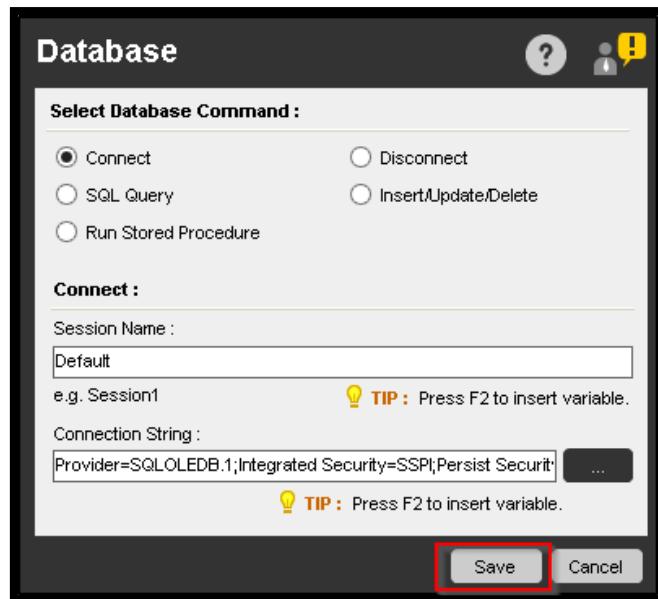
- The following dialog opens as below



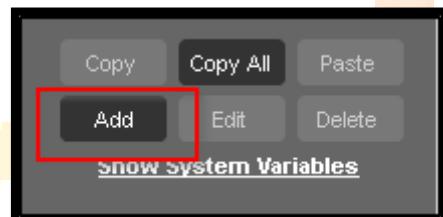
- We give complete path of database , and click on “Test Connection” , we should get “Test Connection Succeeded” as below



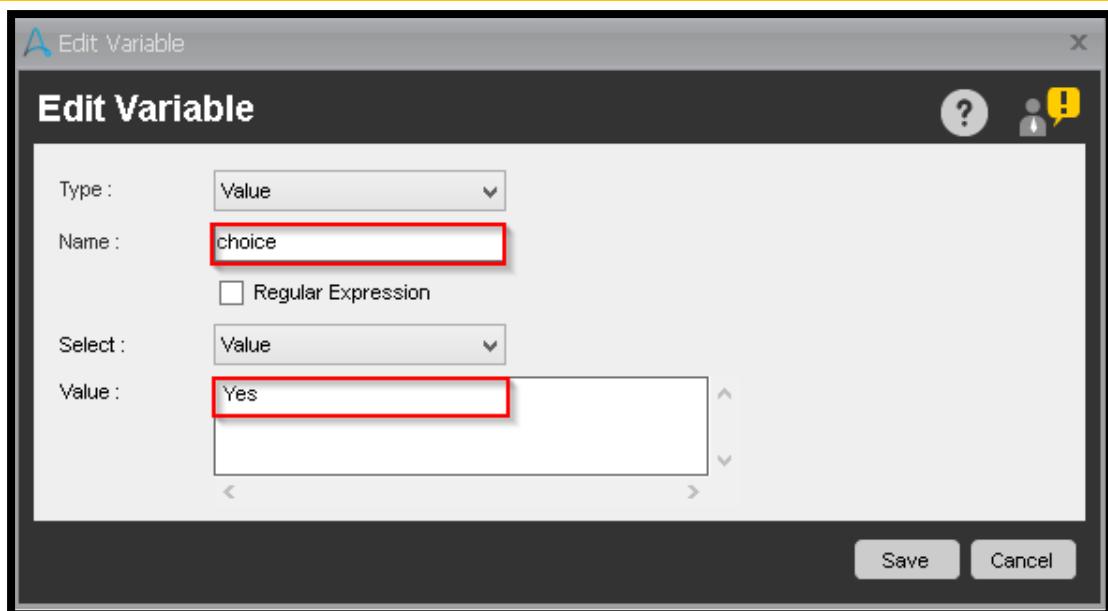
- Next , we click on “Save” as below



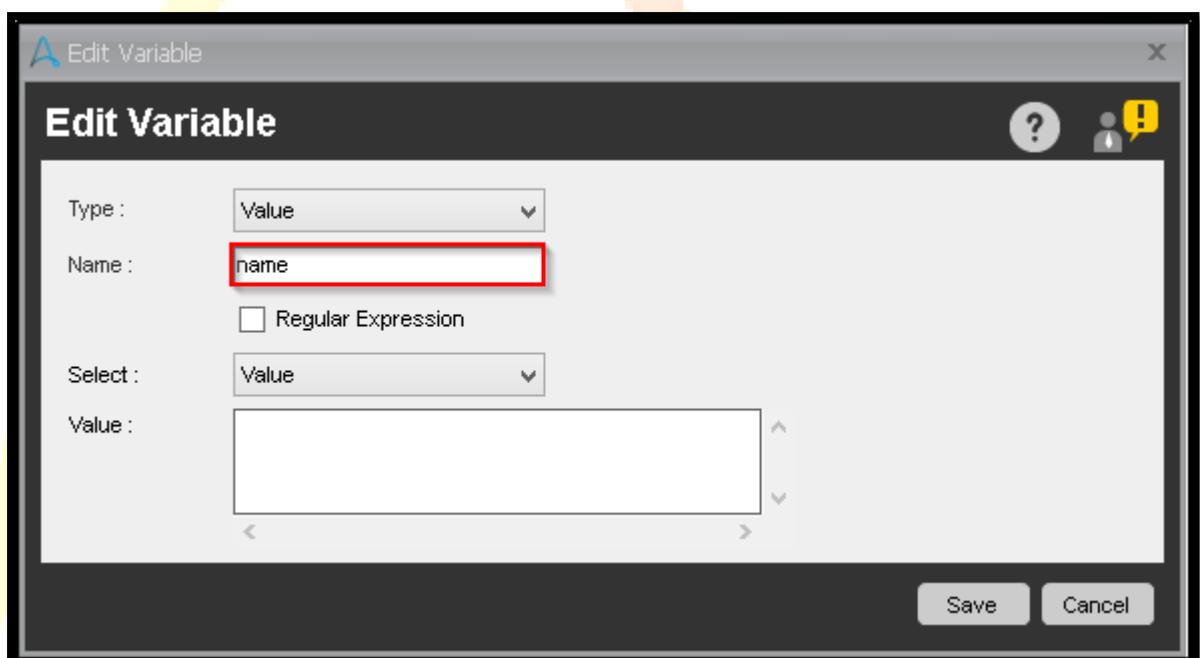
- Next, we create three variables, Name variable for accepting and inserting name in database, Age variable for accepting and inserting name in database, and finally choice variable to ask the user if he wants to iterate again and enter more data.
- We click on “**VARIABLE MANAGER**” at the right corner of the Task editor .
- Next we click on “Add” as below :

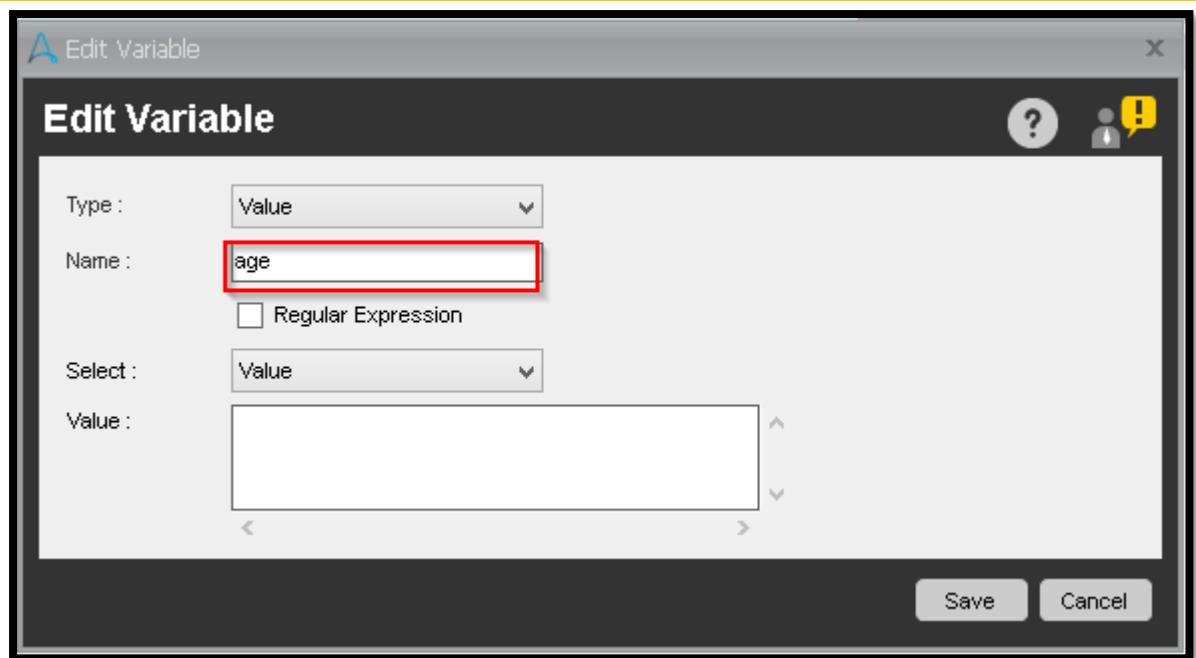


- We create variable “choice” , with value as “Yes” and click on “Save” as below

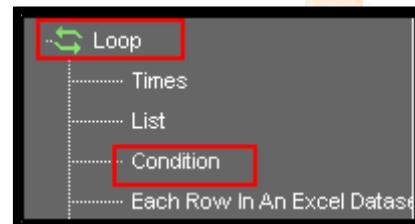


- Variables “Name” and “Age” are created next, no value is assigned to them whatsoever, as below:

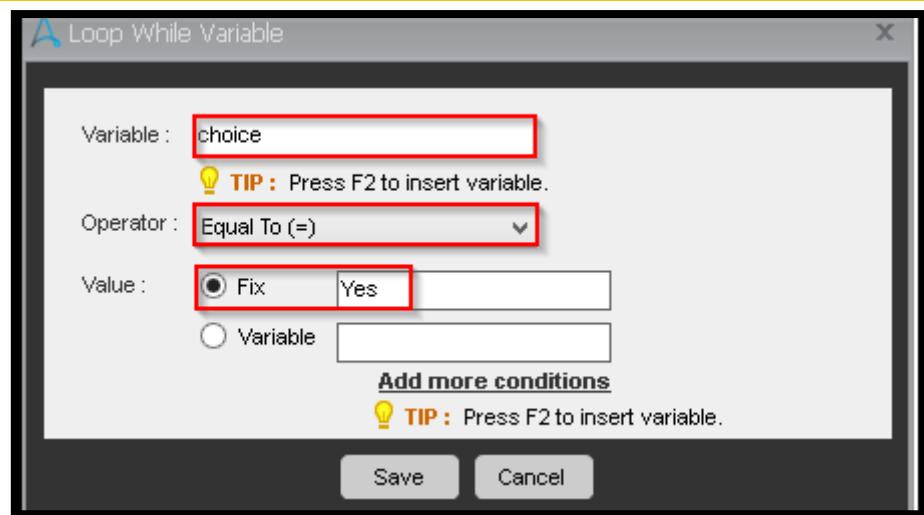




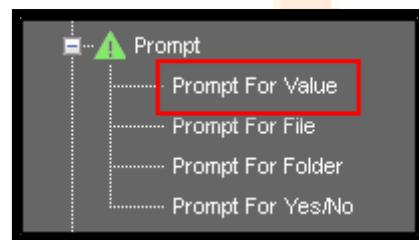
- Next we choose “Loop – Condition” as below

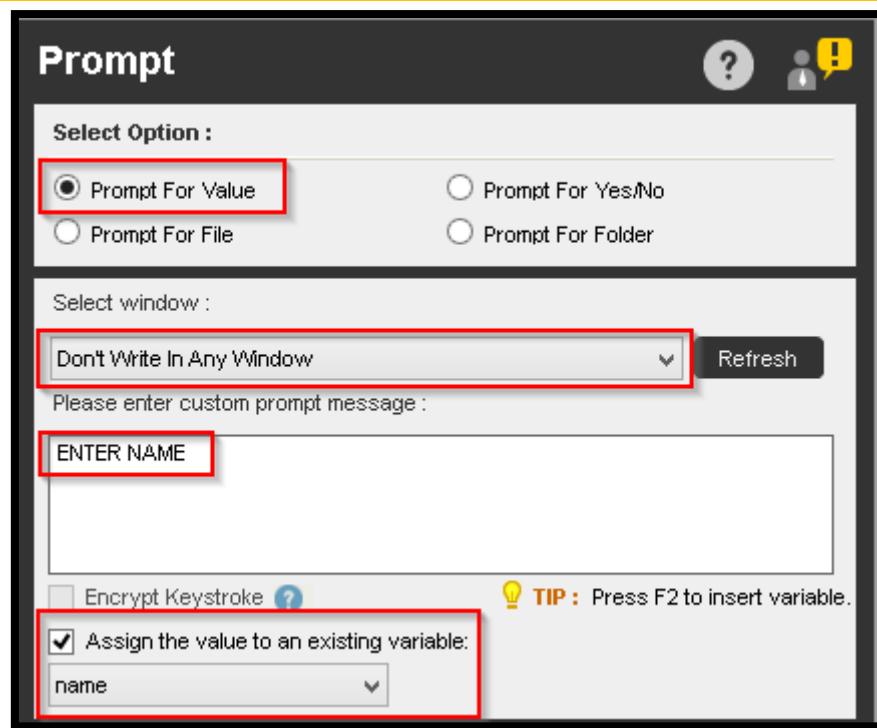


- We enter details as below

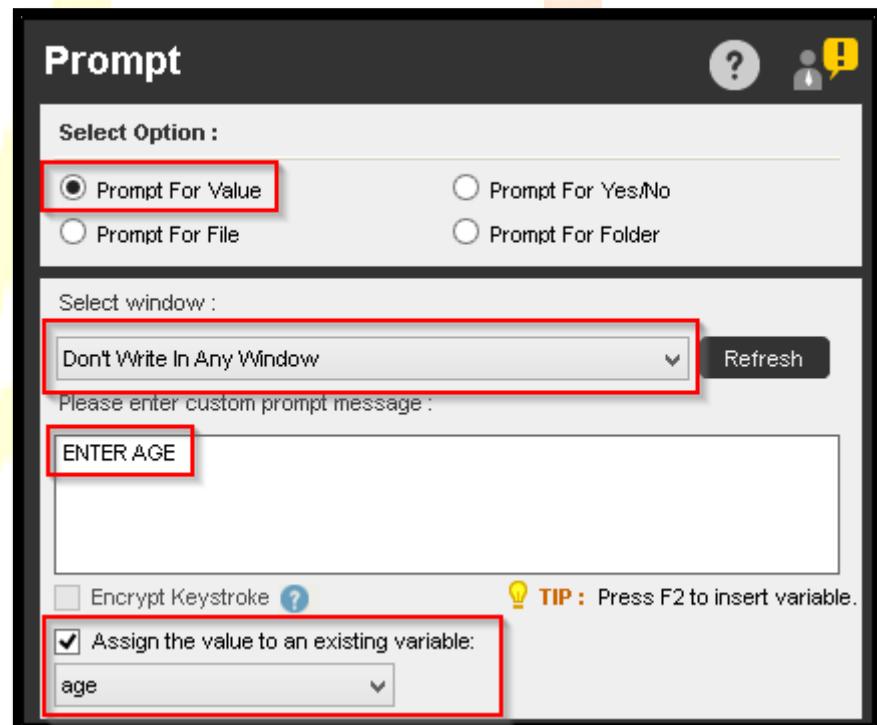


- Next, we choose “Prompt – Prompt for Value” command as below, so that we can prompt the user to enter name, as below

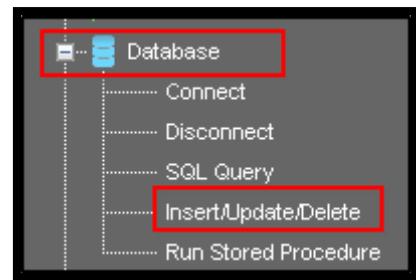




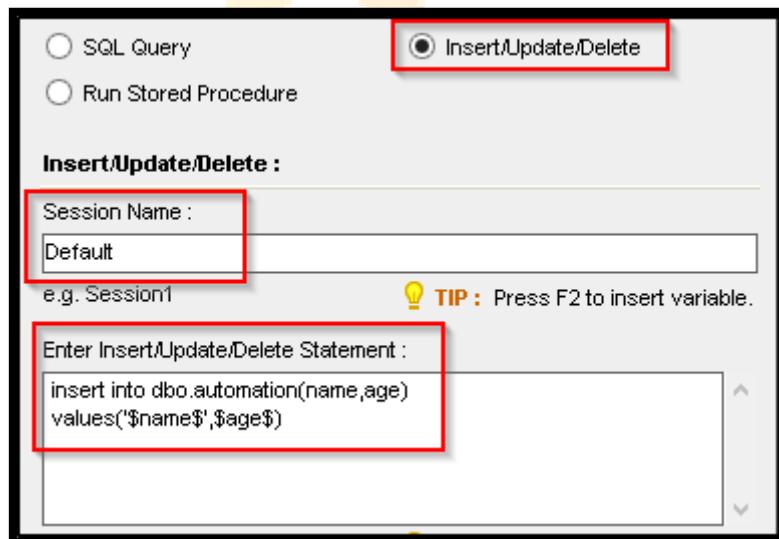
- Next , we prompt user to enter age as below :



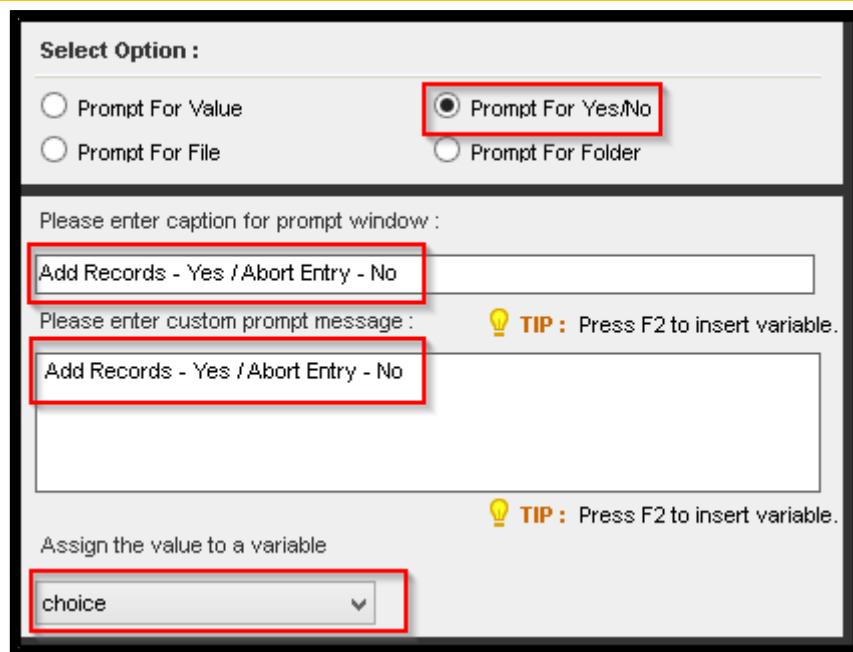
- We use “Database – Insert/Update/Delete” command to add data into the database as below



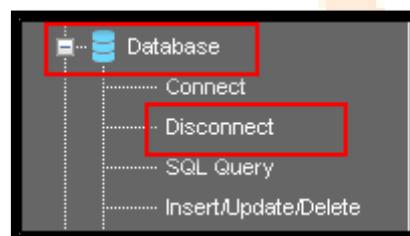
- The following dialog opens as below



- Next we again prompt the user to enter “choice” value “yes” or no”



- Next we choose “Database - Disconnect” command as below



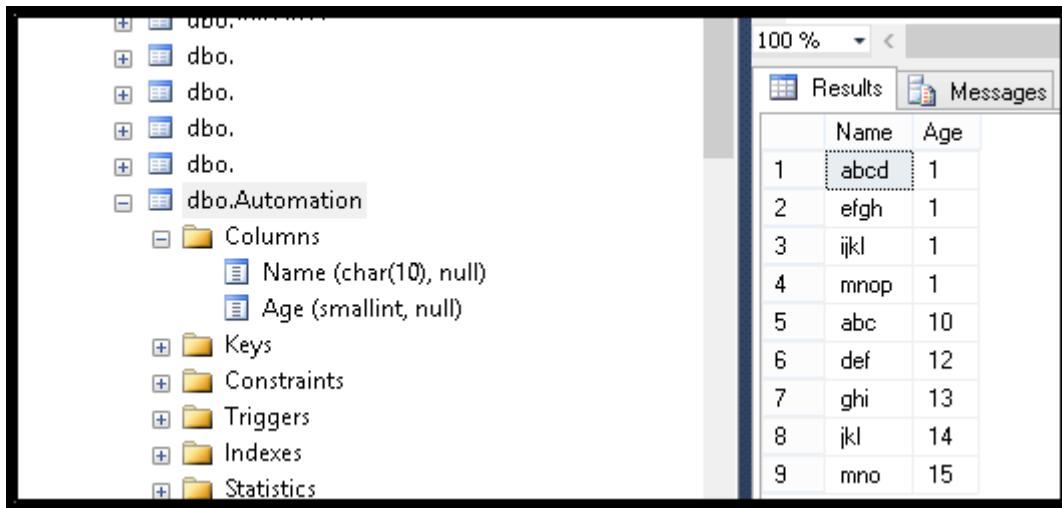
- Our script looks as below

```

Connect to "Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=ernestyoung;Data Source=SSS-LAPTOP31" Session='Default'
Loop While $choice$ Equal To (=) "Yes"
    Comment: Please enter your commands to loop.
    Prompt Message: "ENTER NAME" in "Don't Write In Any Window" Assign value to Variable: $name$
    Prompt Message: "ENTER AGE" in "Don't Write In Any Window" Assign value to Variable: $age$
    Execute SQL Statement: "Insert into dbo.ernestyoung(name,age)values('$name$', $age$)" Session: 'Default'
    Prompt For Yes/No: "Add Records - Yes / Abort Entry - No" for File Assign to variable "$choice$"
End Loop
Disconnect from database Session:'Default'

```

- Next we choose “Save Task” and then click on “Run Task”.
- Once all values have been accepted and entered in the database , we open the database to see output as below



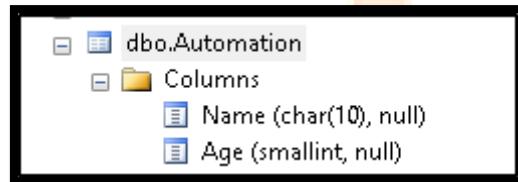
The screenshot shows the Object Explorer on the left with a tree view of a database named 'dbo.Automation'. Under 'Columns', there are two entries: 'Name (char(10), null)' and 'Age (smallint, null)'. To the right, the 'Results' tab is selected in the SSMS interface, displaying a table with 9 rows of data:

	Name	Age
1	abcd	1
2	efgh	1
3	ijkl	1
4	mnop	1
5	abc	10
6	def	12
7	ghi	13
8	JKL	14
9	mno	15

- We have successfully written into our database .

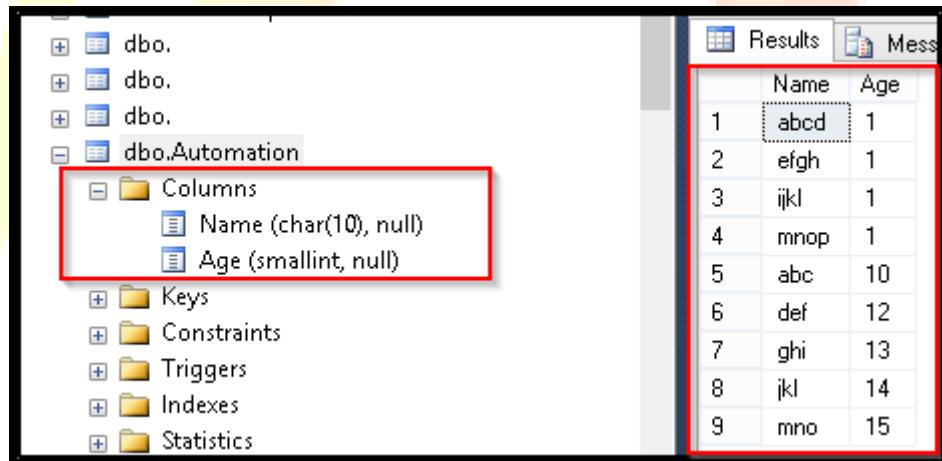
### Example 2: Reading from a Database into a CSV file

- We have already created a database, which has a table as below. Two columns Name, Age as seen below are created in the same.



The screenshot shows the Object Explorer with a callout box highlighting the 'Columns' node under 'dbo.Automation'. This node contains the two columns: 'Name (char(10), null)' and 'Age (smallint, null)'.

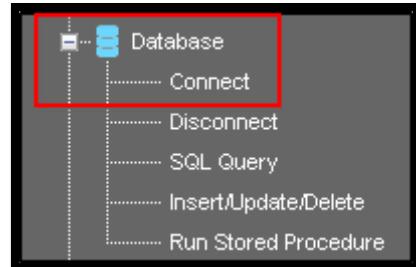
- Our database already has values as below, populated by previous Example



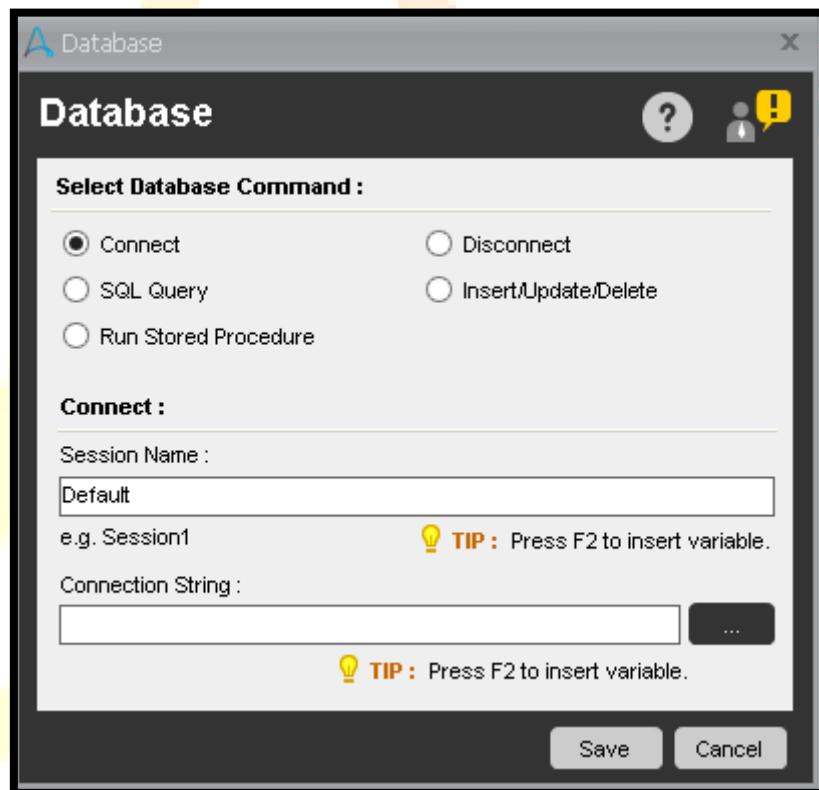
The screenshot shows the Object Explorer and Results grid. A red box highlights the 'Columns' node under 'dbo.Automation'. Another red box highlights the entire data grid in the 'Results' tab, which displays the same 9 rows of data as in the previous screenshot.

- Open the Automation Anywhere Client as below: -
- We choose “New – Task Editor”

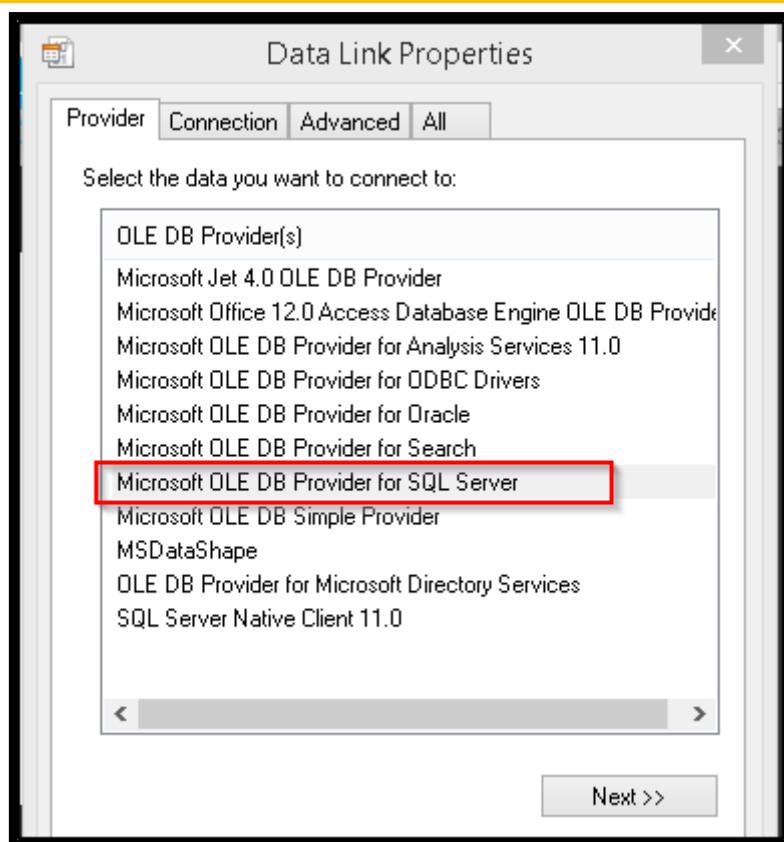
- Next choose Database -> Connect command.



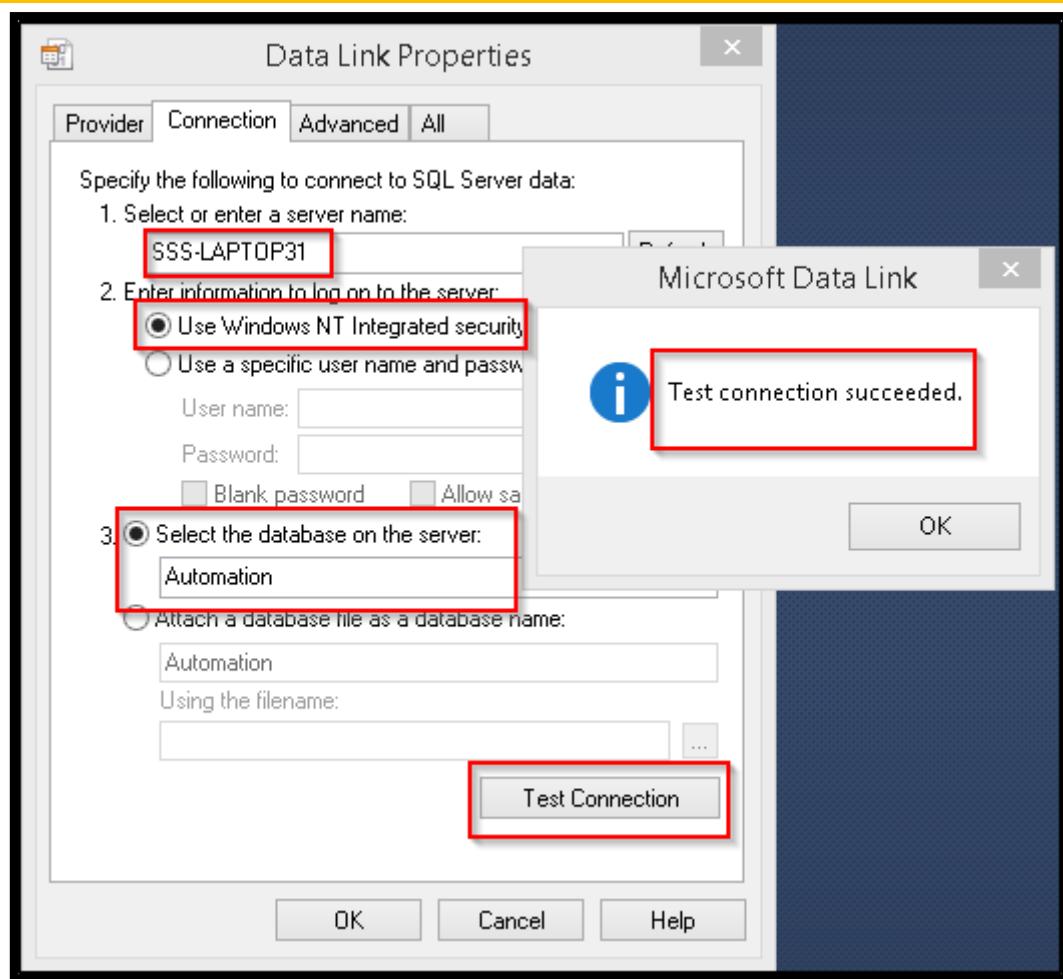
- The following dialog opens



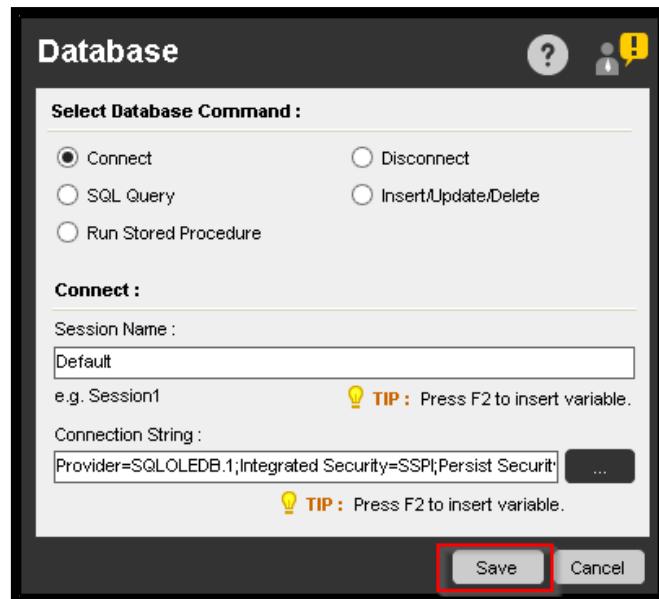
- The following dialog opens as below



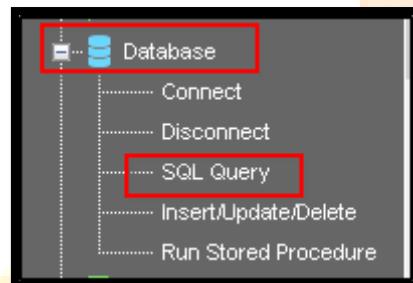
- We give complete path of database , and click on “Test Connection” , we should get “Test Connection Succeeded” as below



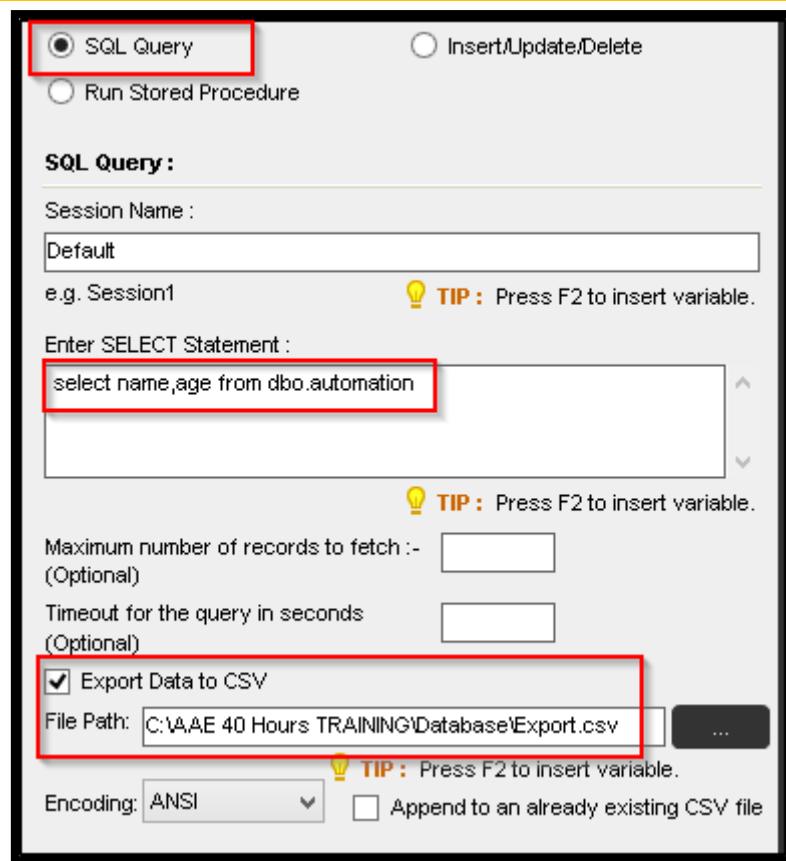
- Next , we click on “Save” as below



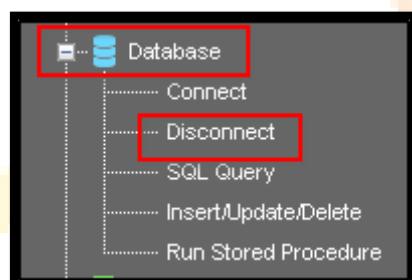
- Next we choose “Database – Sql Query” as below to read information from the database:



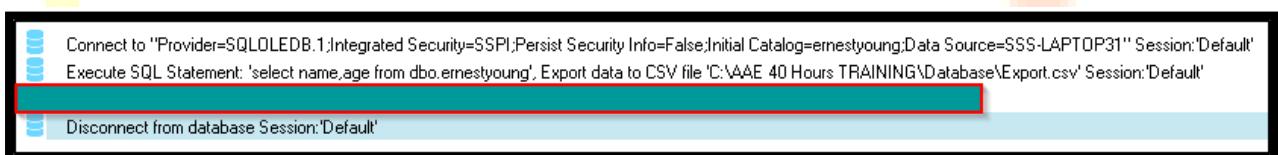
- We enter the “select statement” as below , also we choose “Export Data to CSV” and give “File Path” as below



- Next we choose “Database - Disconnect” command as below



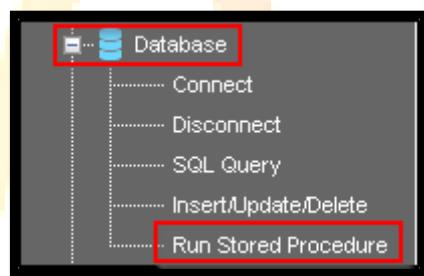
- Our script looks like as below



- Next we choose “Save Task” and then click on “Run Task”.
- We can see our csv file getting populated with contents of the database as below

1	abcd	10
2	efgh	20
3	abcd1	11
4	efgh1	21
5	ijkl	30
6	mnop	40

- Next , we want to execute a stored procedure and view it's results , for that we do the following :



- We give the details of our stored procedure as below

Run Stored Procedure

**Run Stored Procedure :**

Session Name :

e.g. Session1 💡 TIP : Press F2 to insert variable.

Procedure Name :

e.g ProcName(1,\$name\$) 💡 TIP : Press F2 to insert variable.

Add Parameter(s)

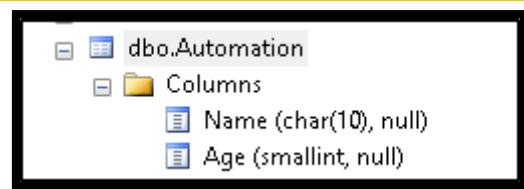
Export Data to CSV

File Path:

- We then proceed to save the script and then execute it , which will give us the results in a separate csv file .

### Example 3: Reading from a Database row by row

- We have already created a database, which has a table as below. Two columns Name, Age as seen below are created in the same.

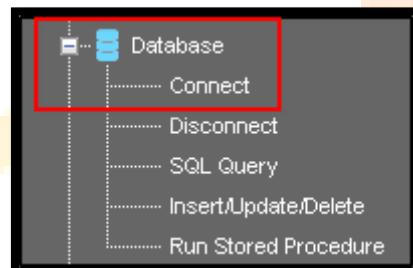


- Our database already has values as below, populated by previous Example

**Results**

	Name	Age
1	abcd	1
2	efgh	1
3	ijkl	1
4	mnop	1
5	abc	10
6	def	12
7	ghi	13
8	JKL	14
9	MNO	15

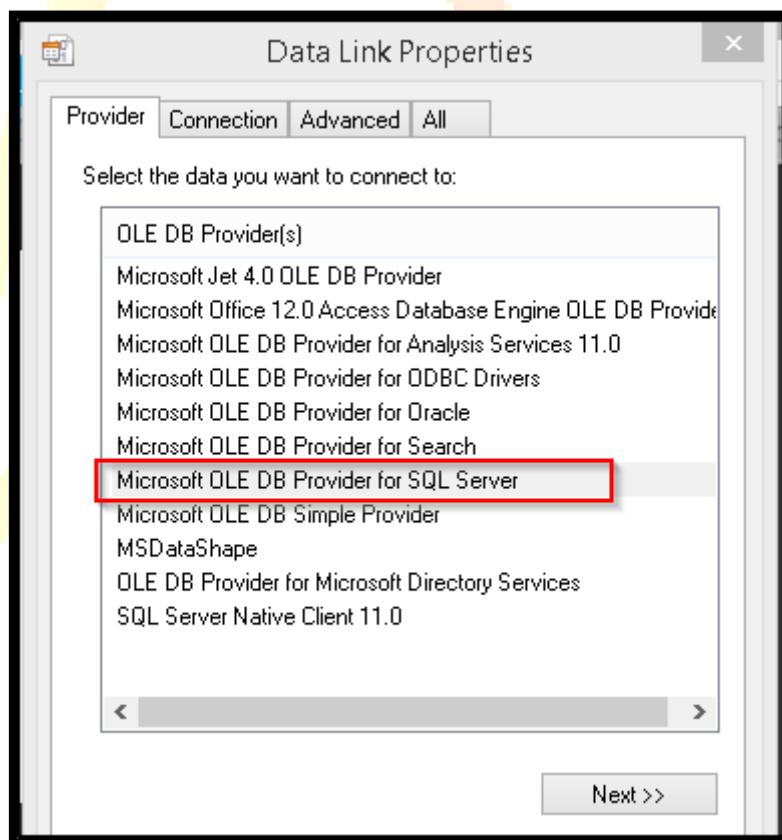
- Open the Automation Anywhere Client as below: -
- We choose “New – Task Editor”
- Next choose Database - > Connect command.



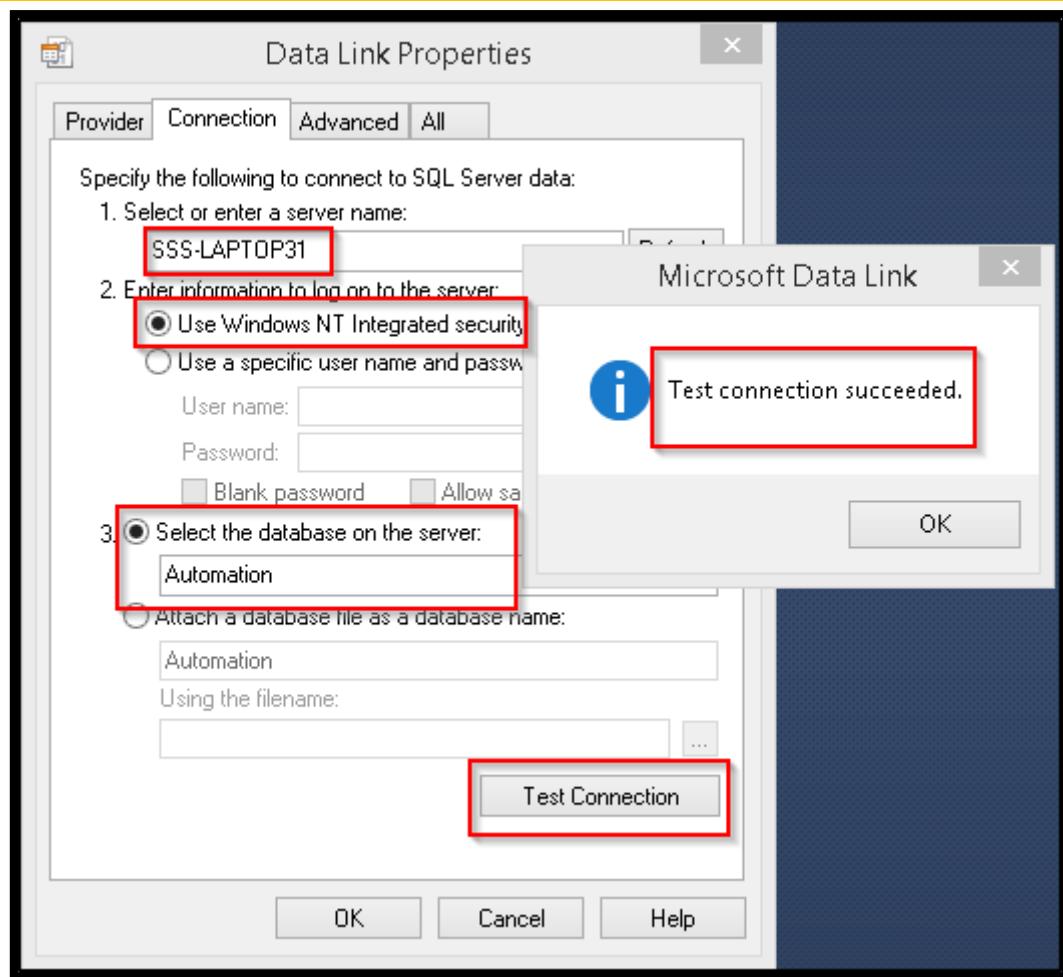
- The following dialog opens



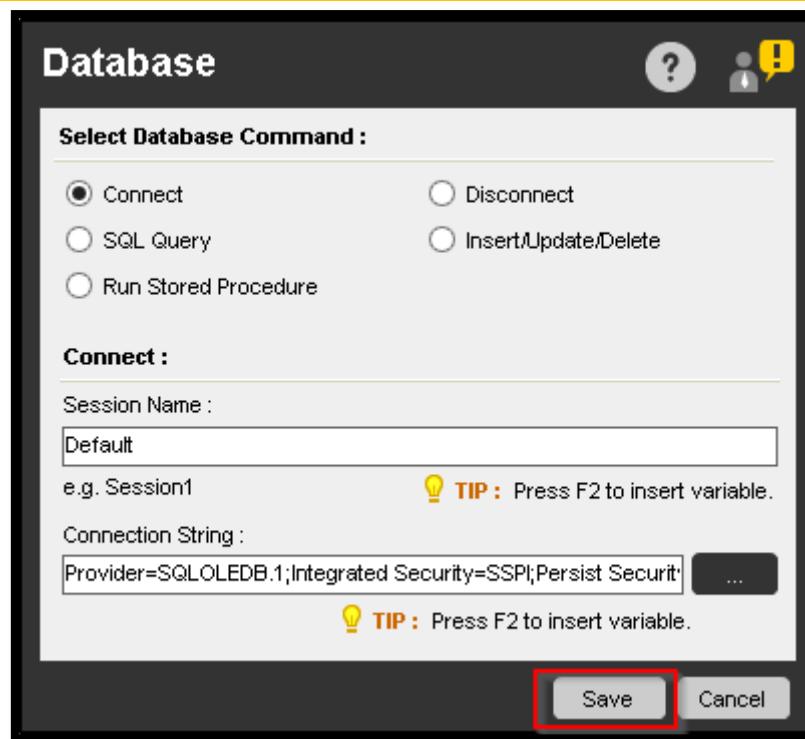
- The following dialog opens as below



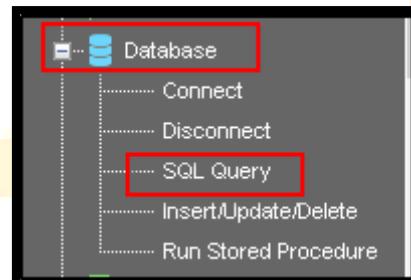
- We give complete path of database , and click on “Test Connection” , we should get “Test Connection Succeeded” as below



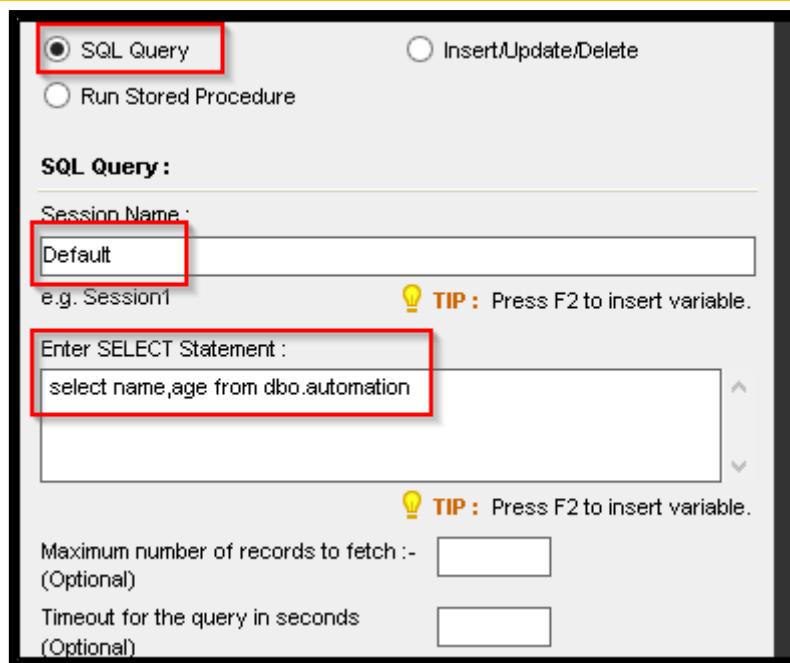
- Next , we click on “Save” as below



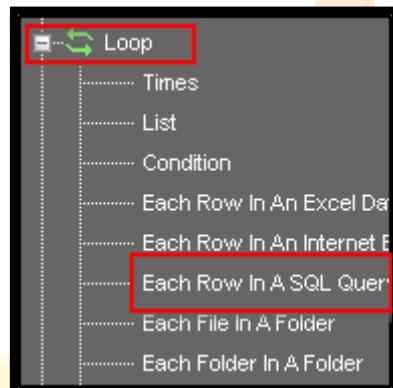
- Next we choose “Database – Sql Query” as below to read information from the database:



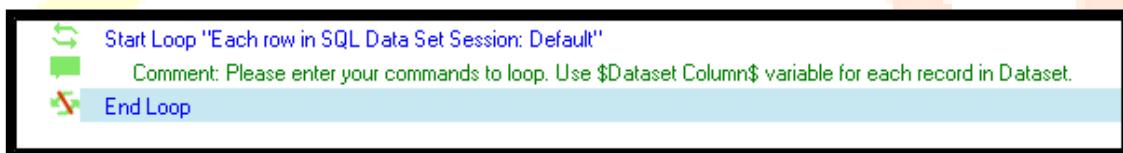
- We give in our query as below:



- Next we drag a “Loop – Each Row in a sql query dataset” as below



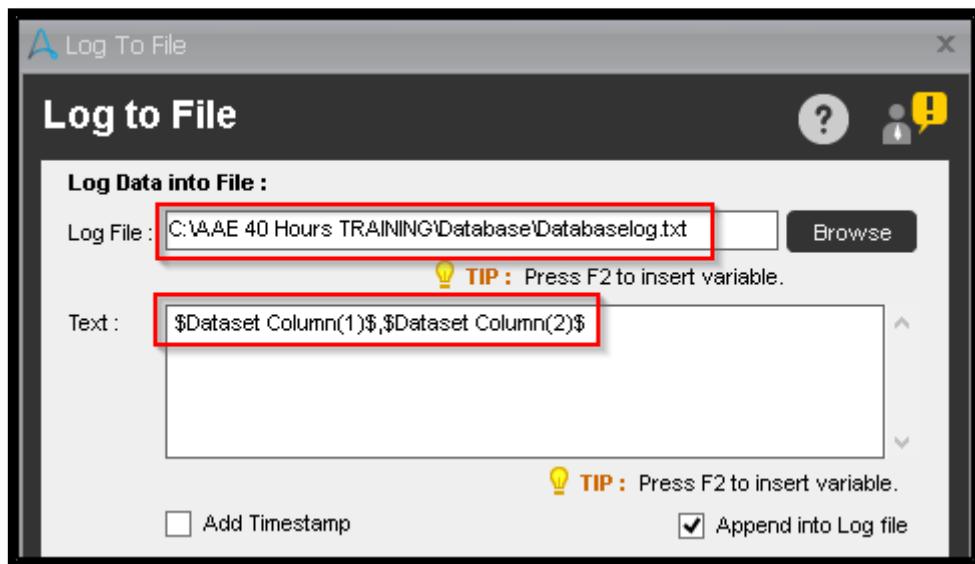
- This adds a “Start Loop – End Loop” statement to our code , which looks like below



- As we are iterating over the rows of the sql query set , we want to write out the data into a log file , for which we bring in the “Log to File” command as seen below :



- Our code looks like as below:



- A .txt file is created with the required data .
- This approach allow us to create a text file which will read information row by row , specific columns as per our requirement .

## 6) Files and Folders/Loop Each File in a Folder

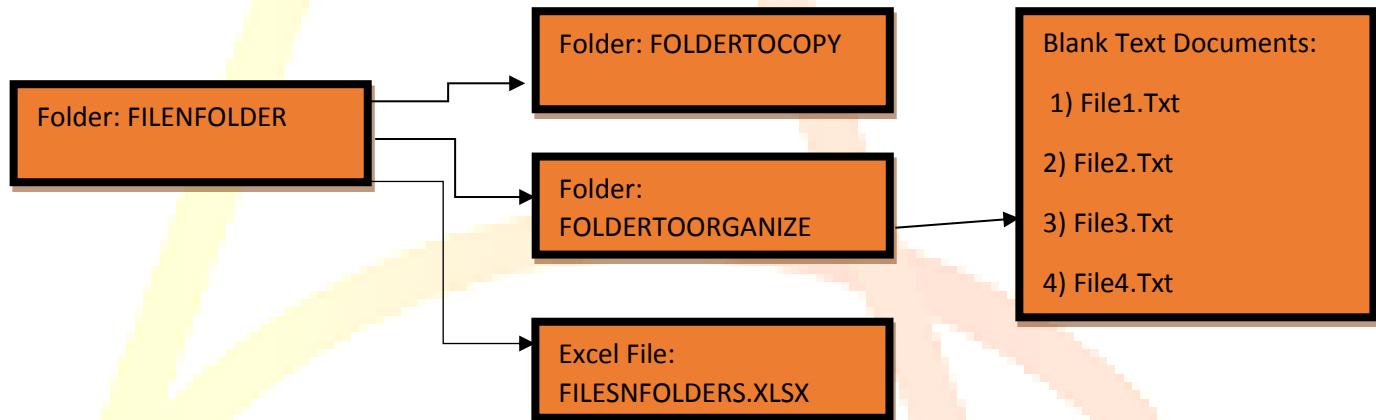
### Objective

- Step by step demonstration of Files and Folders command in Automation Anywhere Task Editor having the following features:

- 1) Copy Files
- 2) Rename Files
- 3) Delete Files
- 4) Loop Each File in a Folder
- 5) Prompt for Value/Folder
- 6) If/Else – Variable
- 7) If/Else – File Exists/File Does Not Exist
- 8) Revise Excel Commands

### Pre-Requisites:

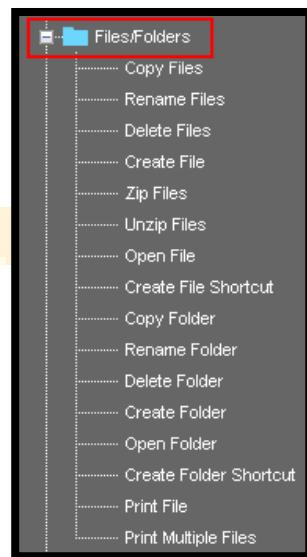
- In our Training Folder that we have made on the desktop, create the Following Folder and File Structure



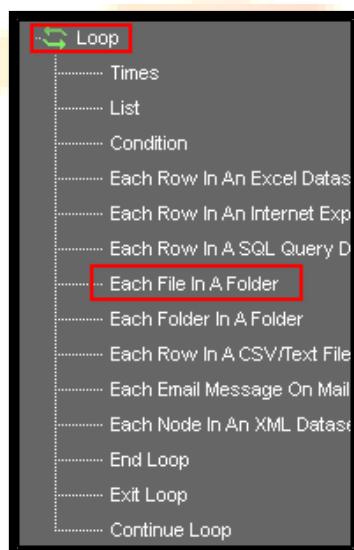
### Example: Copy/Rename/Delete Files in a Folder

- Start your Automation Anywhere Client, login using credentials provided to you
- Next we will open our Task Editor, by clicking on “”, this opens the following dialog where we can click on “Task Editor”

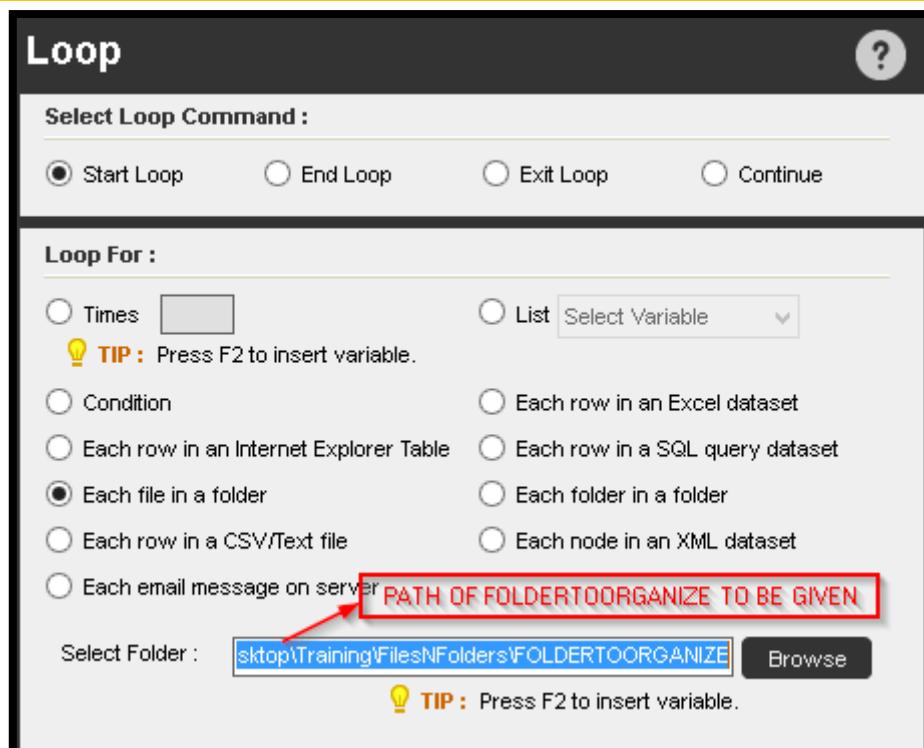
- This opens the “Task Editor” as below , where we need to work with the “Files And Folders” set of commands as seen below :



- We need to create a Task Script which will loop through every file in the concerned folder and will give us an option to Copy/Rename/Delete the file based on our choice. And also log the FileName And It's Extension in Excel along with the directory name.
- Since we need to pick up each and every file in a folder we will use the following command under Loop as below :



- We drag and drop the above command into our Task Editor, which opens the following dialog as below :



- The Loop gets created on the screen as below :

```

1  ↳ Start Loop "Each File In Folder E:\Ajay.Mehta\Desktop\Training\FilesNFolders\FOLDERTOORGANIZE"
2  ↳ Comment: Please enter your commands to loop. Use $filename$.extension$ variable for each file name in the Loop.
3  ✖ End Loop

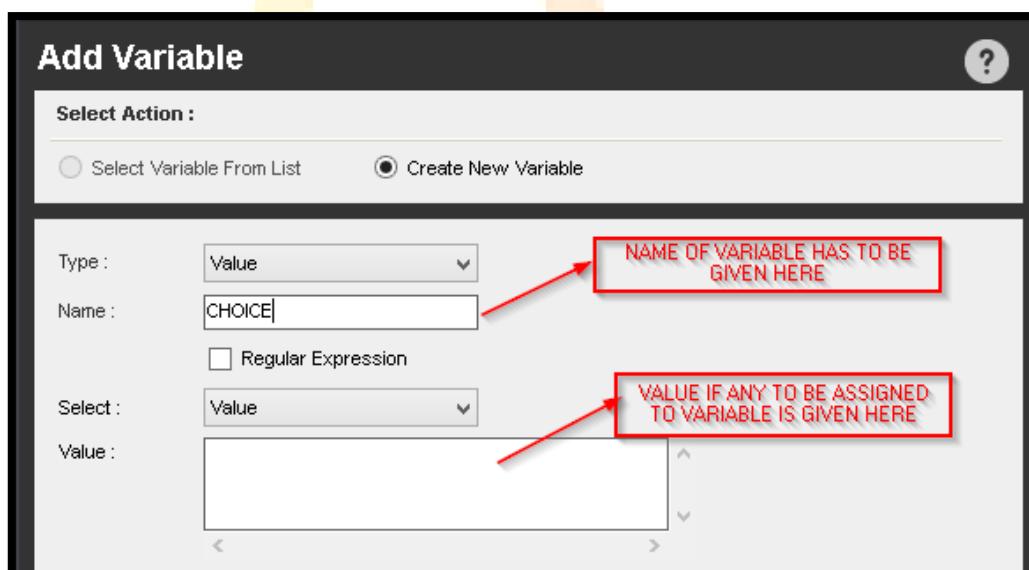
```

- The comment specifies that we need to use the \$filename\$.extension\$ variable , in order to get complete information about the file which has been currently picked up.
- In order to accept user inputs like “user choice – Copy or Rename or Delete” , folder to copy to , New filename , we need to create certain variables which we create as below
  - a)Choice – Copy/Rename/Delete , user choice to be captured here .
  - b)Copyfolder – Folder name to copy file to be captured here .
  - c)Renamefile – To get new name of file from the user .
  - d)Confirmdelete - To doublecheck with user if he wants to delete file

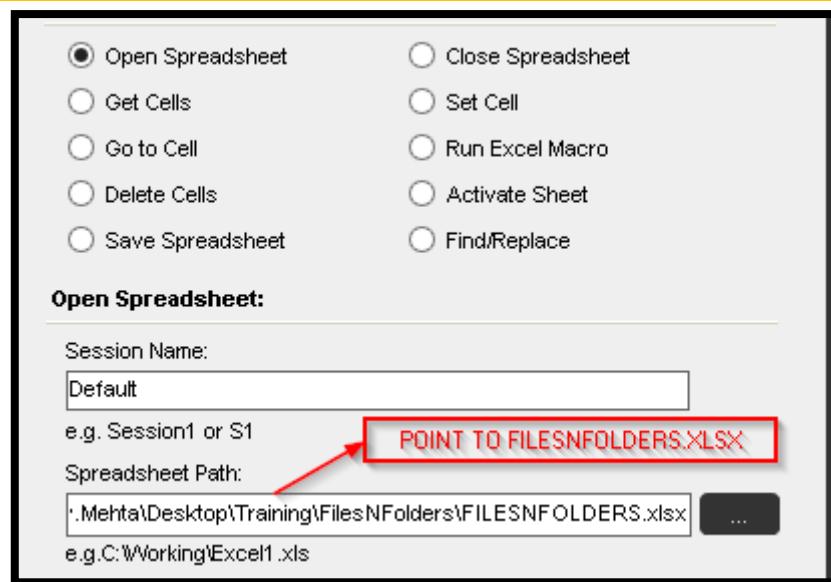
- In order to create variables , we click on “**VARIABLE MANAGER**” on the right of our Task Editor, this opens the Variable Manager on the right hand side of our screen.
- We click on “Add” inside “Variable Manager” as below :



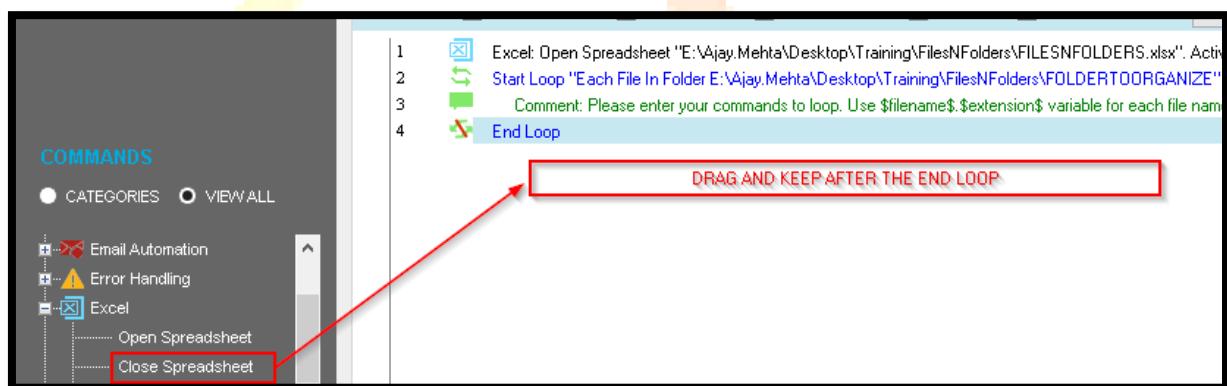
- This opens the following dialog as below :



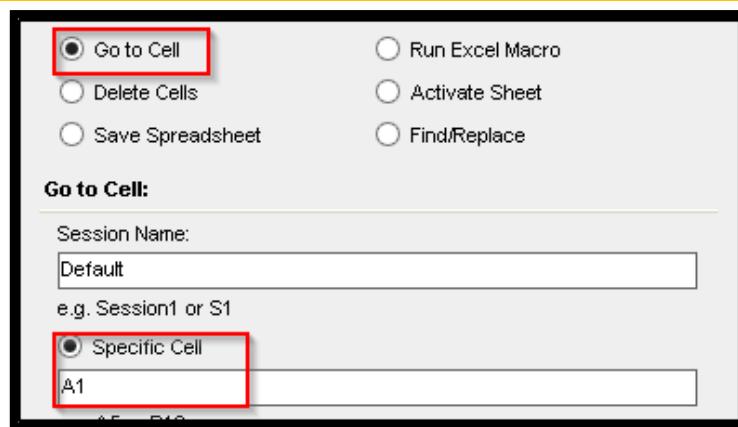
- Similar to creating the “Choice” Variable , we can create “Copyfolder” and “Renamefile” and “Confirmdelete” variable.
- Now we will put in the logic to log the “CurrentDirectory” Name and “FileName” and “Extension” of the file over which we are iterating into Excel.
- We Drag an “Excel – Open Spreadsheet” command above “Start Loop” and point to our Excel sheet made earlier as below



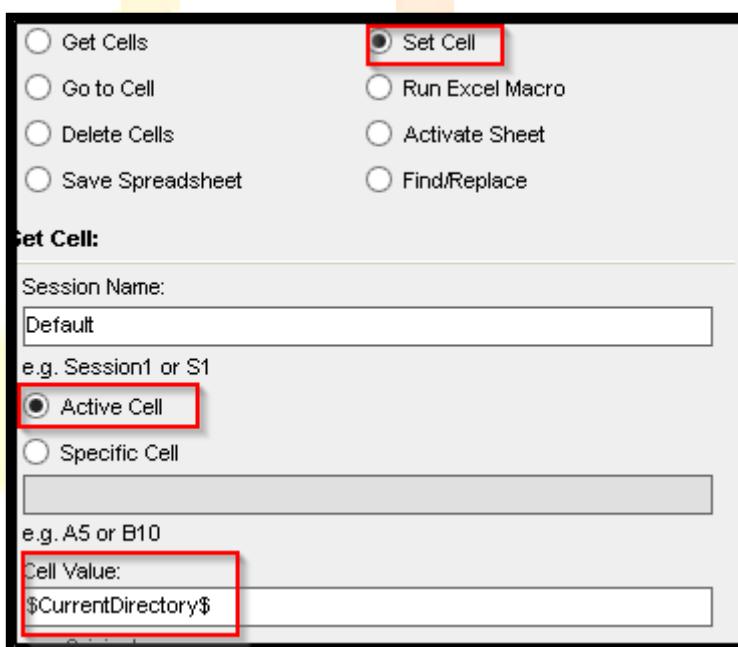
- Similarly drag an “Excel-Close Spreadsheet” and keep it after the “End Loop”



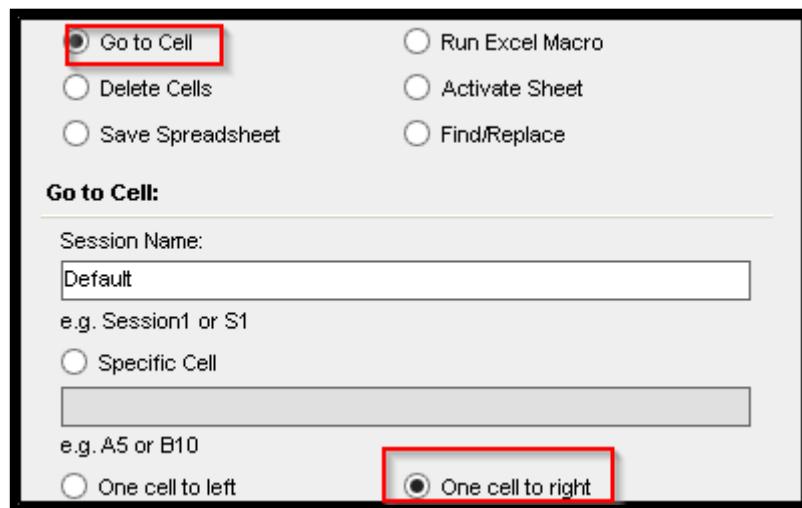
- Next Drag An “Excel – Goto Cell” between Line 1 And Line 2 and make “A1” as the active cell as below



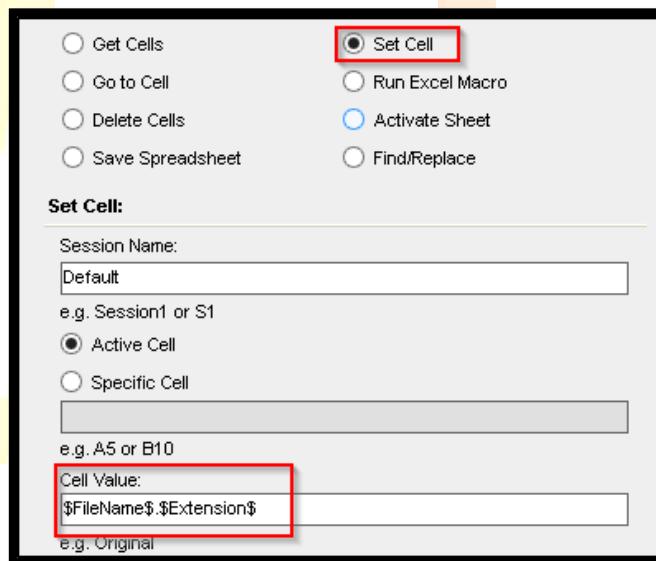
- Next Drag An “Excel – Set Cell” and Assign value as “\$CurrentDirectory\$” as below



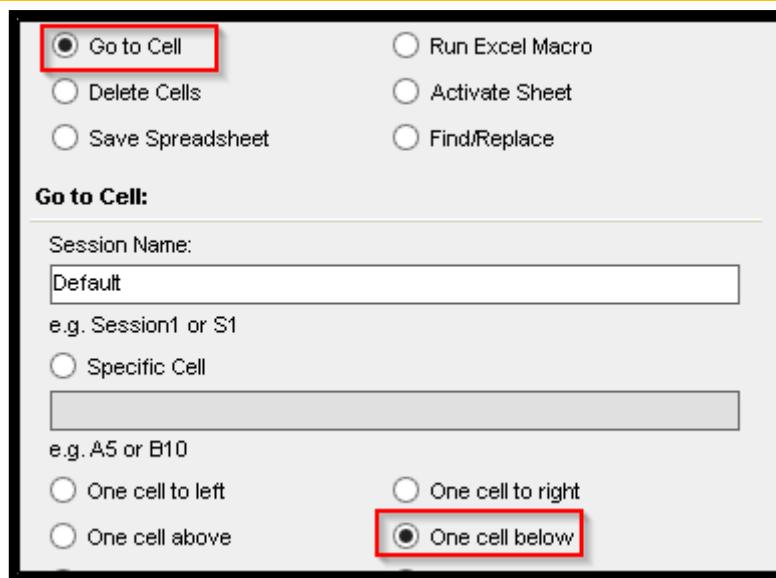
- Then Drag An “Excel – Goto Cell” and choose “One Cell to the right” button as below



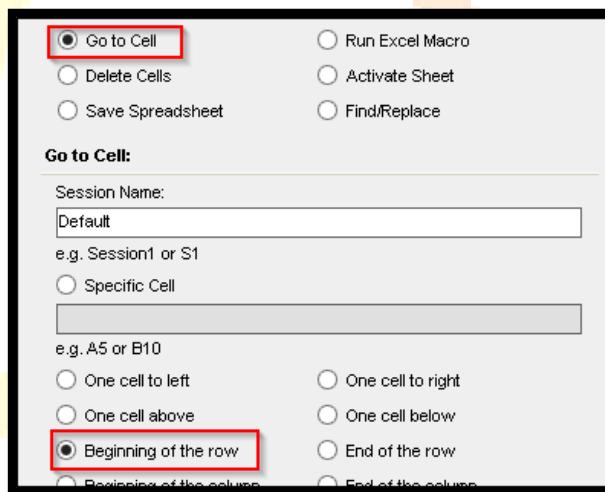
- Next Drag An “Excel – Set Cell” and Assign value as “\$FileName\$.Extension\$” as below



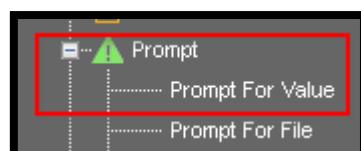
- Next We Will Drag An “Excel – Goto Cell” and choose “One Cell Below” button as below



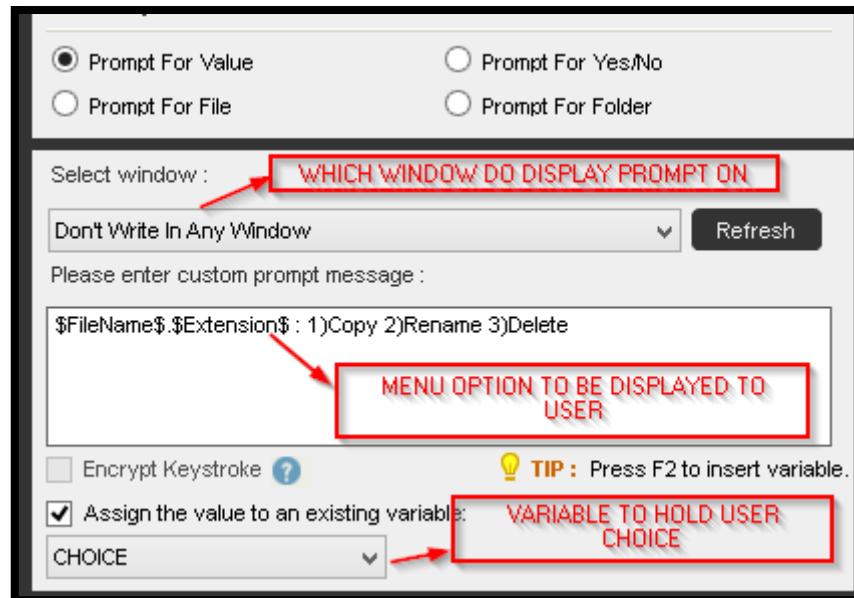
- Again Drag an “Excel – Goto Cell” and Choose “Beginning of the Row” button as below



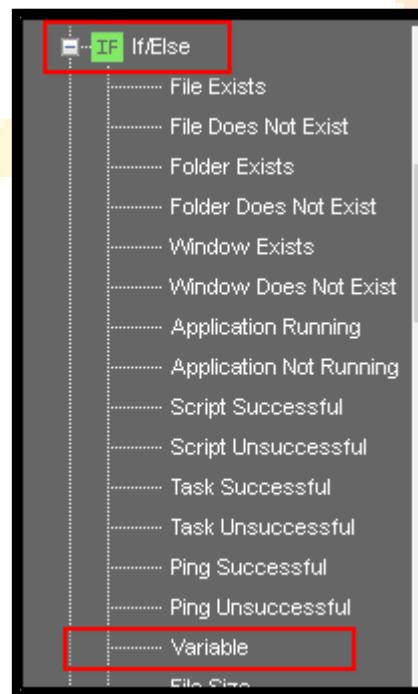
- Next , we need to prompt the user to enter his choice , for that we use “Prompt – for Value” command as below



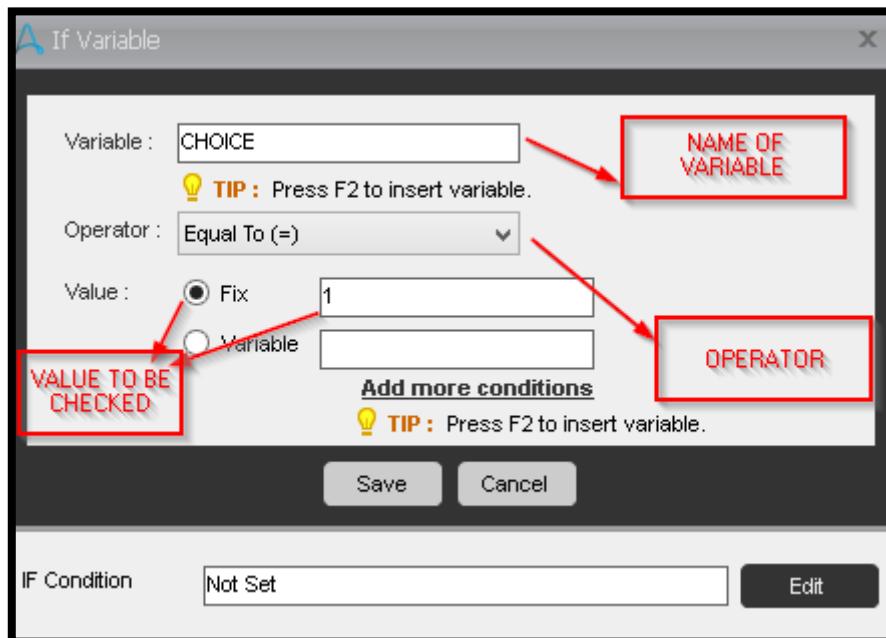
- We specify window to write on top , Specify custom message to prompt user and finally value is stored in variable “Choice”



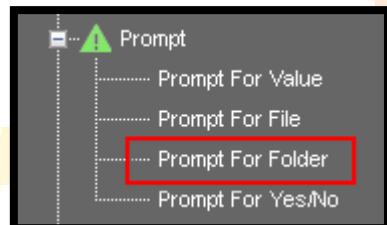
- Next based on user input , corresponding action has to be performed . So , we choose “If/Else – Variable ” command to check user input as below :



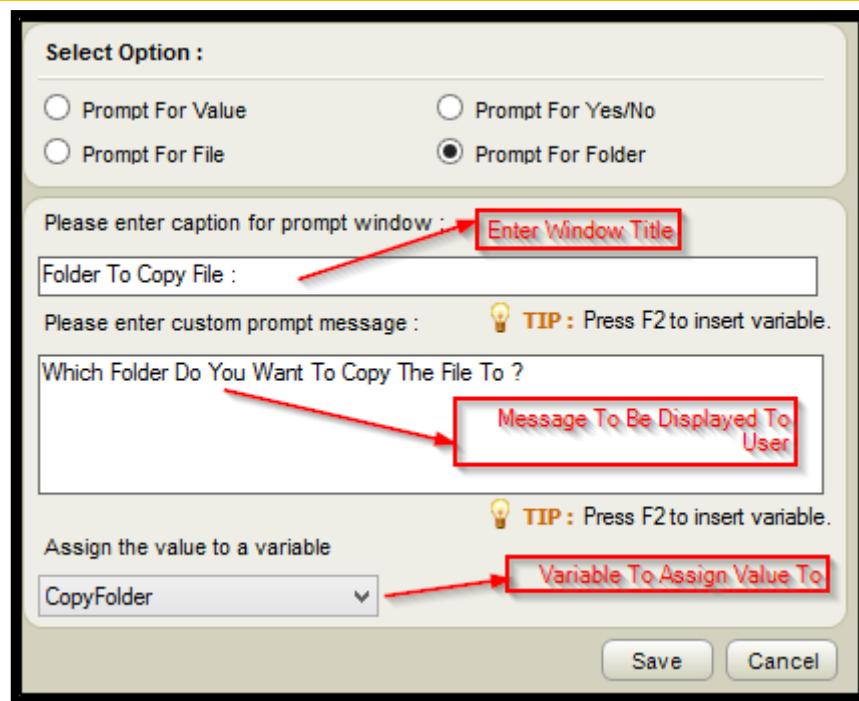
- We specify condition to be checked as below :



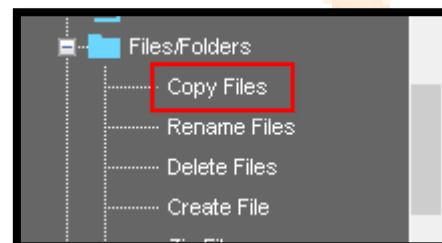
- If "Choice = 1" , then user has chosen to copy file , so next we need to ask him where he wants to copy the file to , using "Prompt – Prompt For Folder" as below



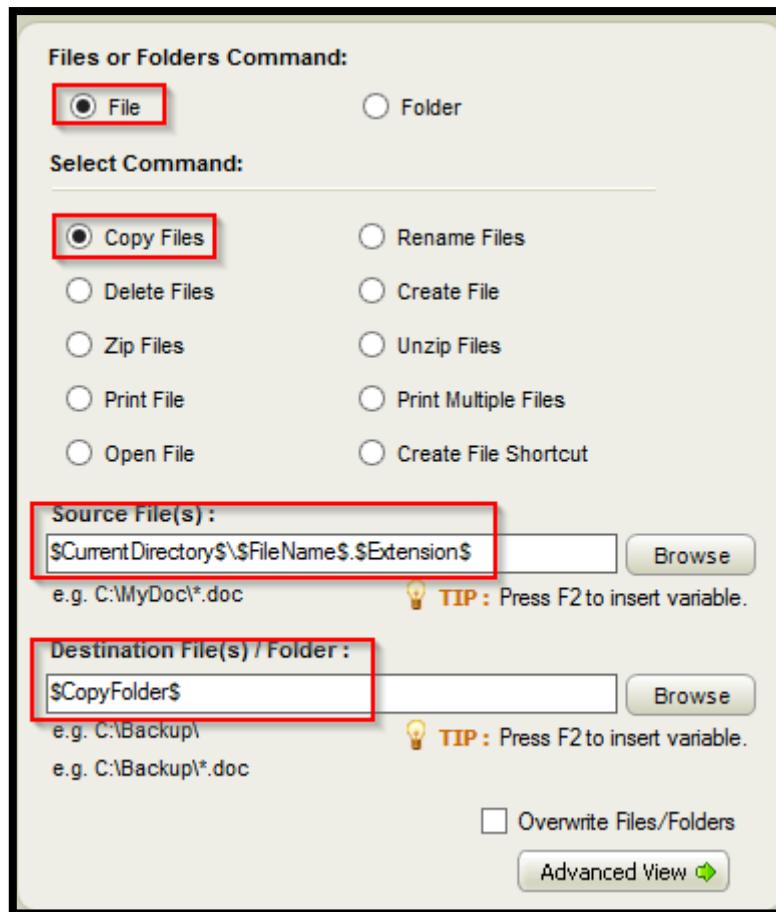
- We fill in the dialog that opens as below



- Next we use “Files/Folders – Copy Files” command as below



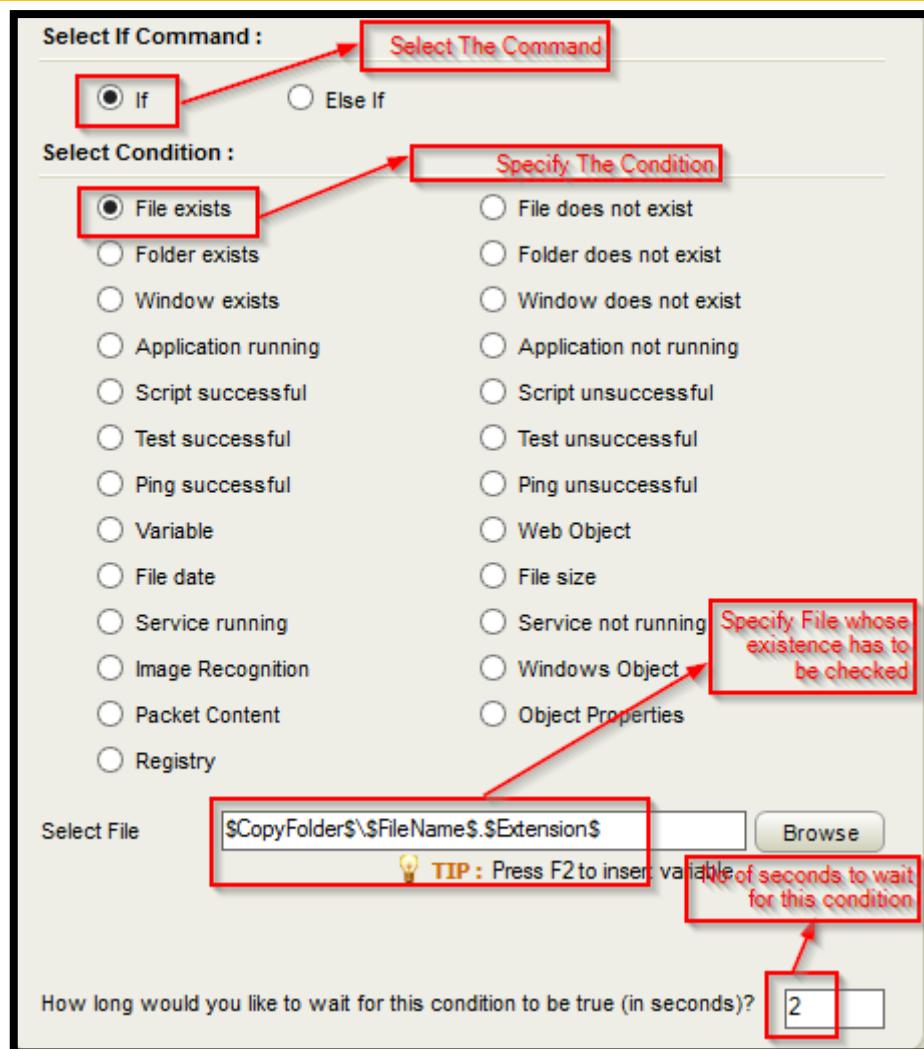
- We fill in the dialog as below :



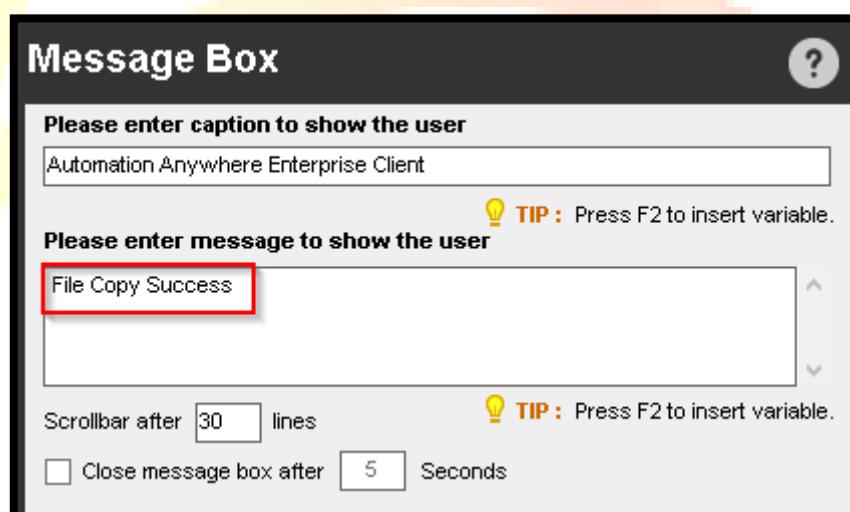
- Next , we check if the file has indeed been copied to the folder specified , for that we use “If/Else – File Exists” as below



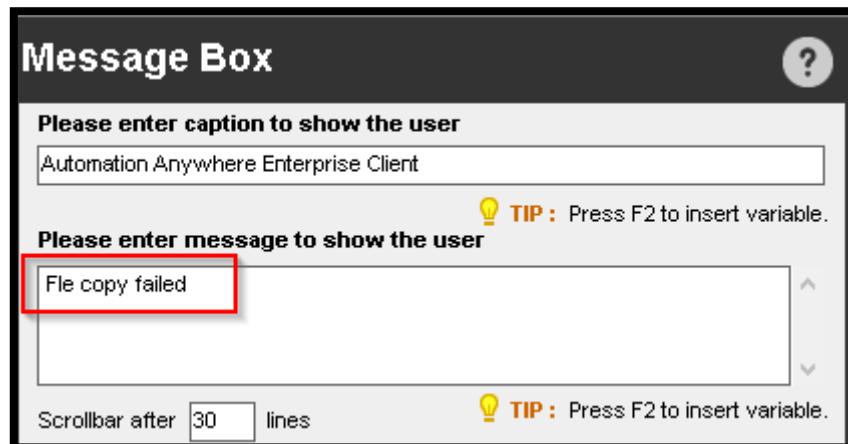
- The following dialog opens as below , where we can specify the condition as below :



- Next we display a message to the user , using “MessageBox” command



- Next we place an “” command , so as to handle the situation where the “If” condition may not have been satisfied .
- This is followed by a “MessageBox”(  ) command , to display the relevant message as below



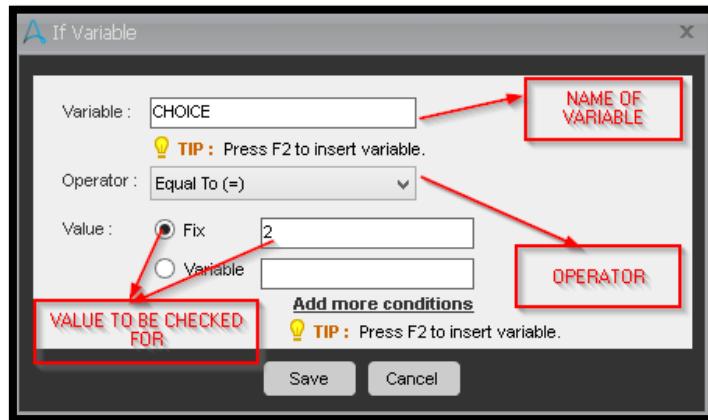
- Our script should look like as below :

```

1  Excel: Open Spreadsheet "E:\Ajay.Mehta\Desktop\Training\FilesNFolders\FILESNFOLDERS.xlsx". ActiveSheet: "Default". Session: Default
2  Excel: Go to cell "A1". Session: Default
3  Start Loop "Each File In Folder E:\Ajay.Mehta\Desktop\Training\FilesNFolders\FOLDERTOORGANIZE"
4    Comment: Please enter your commands to loop. Use $filename$.$extension$ variable for each file name in the Loop.
5    Excel: Set value of Active Cell with "$CurrentDirectory$". Session: Default
6    Excel: Go to one cell right of active cell. Session: Default
7    Excel: Set value of Active Cell with "$FileName$.$Extension$". Session: Default
8    Excel: Go to one cell below the active cell. Session: Default
9    Excel: Go to beginning of the row of the active cell. Session: Default
10   Prompt Message: "$FileName$.$Extension$ : 1)COPY 2)RENAME 3)DELETE" in "Don't Write In Any Window" Assign value to Variable: $CHOICE$
11   If $CHOICE$ Equal To (=) "1" Then
12     Comment: Please enter the conditional commands here.
13     Prompt For Folder: "Which folder to copy to " for Folder Assign to variable "$COPYFOLDER$"
14     Copy Files "$CurrentDirectory$\$FileName$.$Extension$" to "$COPYFOLDER$"
15     If File Exists ("$COPYFOLDER$\$FileName$.$Extension$") Then
16       Comment: Please enter the conditional commands here.
17       Message Box: "File Copy Success"
18     Else
19       Message Box: "File Copy Failed"
20     End If
21   End If
22 End Loop
23 Excel: Close Spreadsheet. Session: Default

```

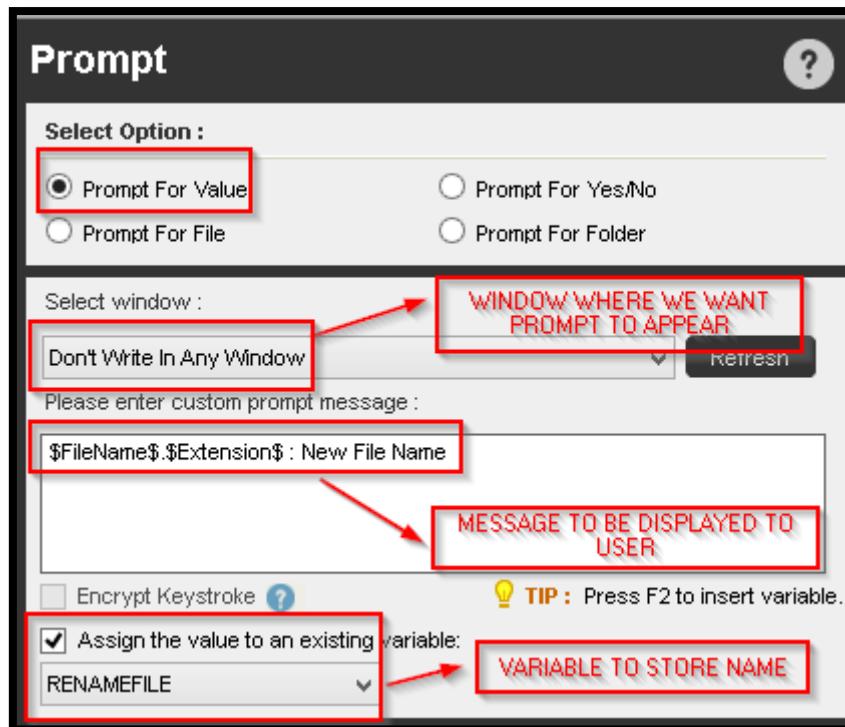
- Next , we again have to check for the value of the variable “choice” for which we again choose “If/Else – Variable” command , which opens the following dialog as below :



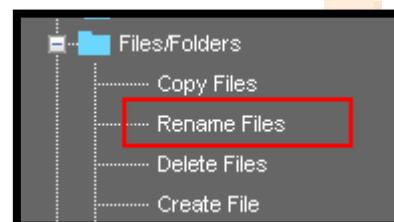
- Since “Choice = 2” is for rename , we need to ask the new name of the file here , for which we choose “Prompt – Prompt For Value” as below



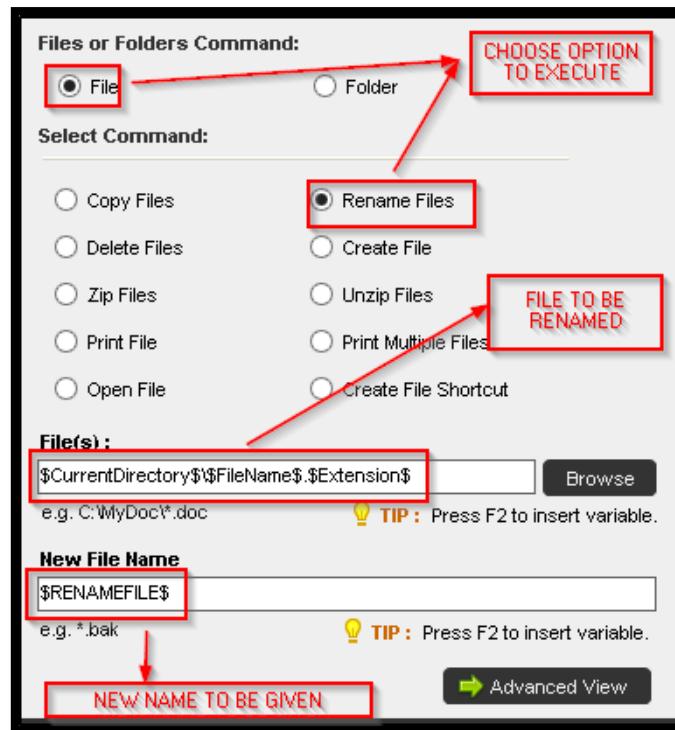
- This opens the following dialog , where we specify the following options as below



- Next we choose “Files/Folders – Rename Files” command as below :



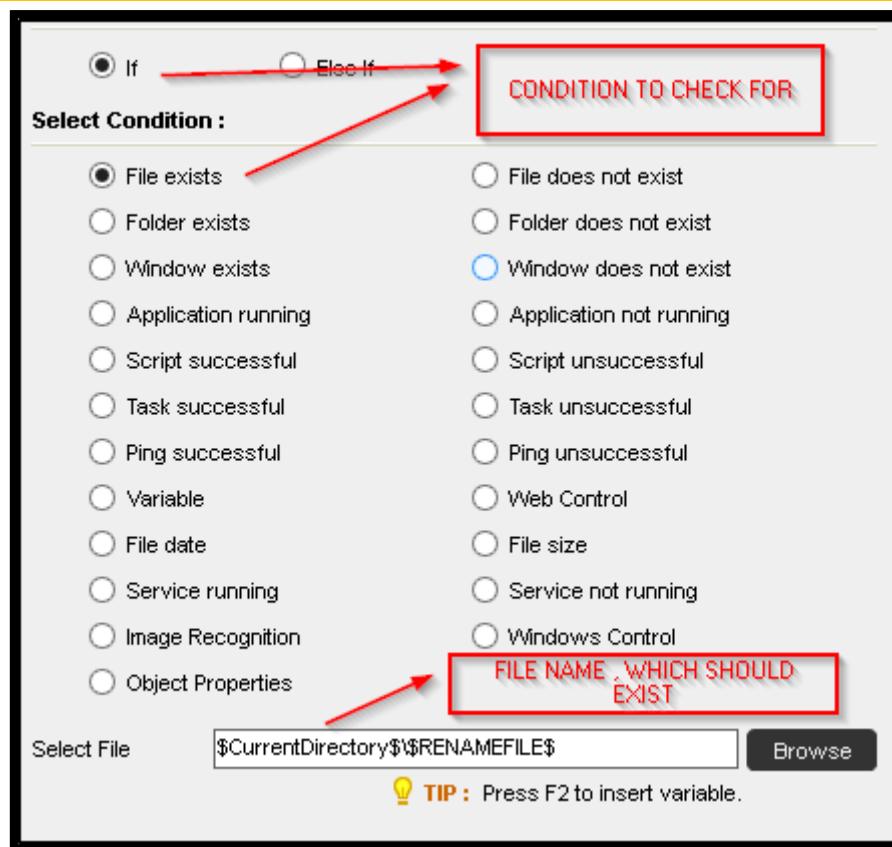
- The following dialog then opens as below



- Next we put in a check condition to see if the file did get copied or not as below :



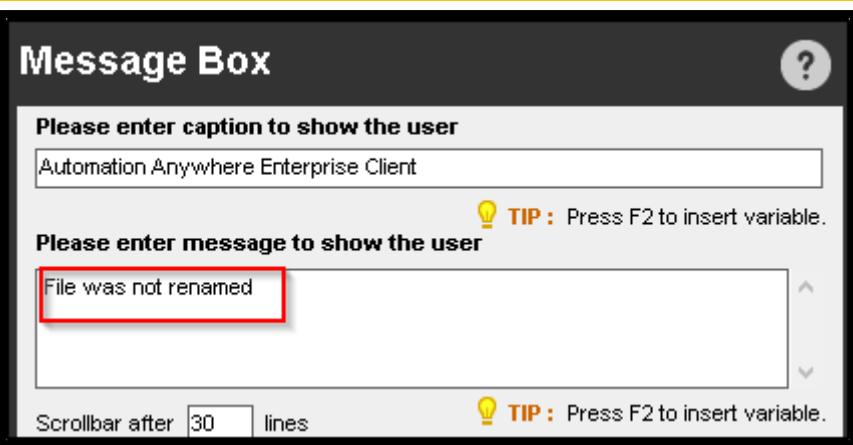
- The following dialog opens as below , where we specify our details



- Next, we specify a message box to display a message if the "If" condition is true as below :



- Next we put in a "Else" command , for the eventuality where the condition specified in the "If" part fails . A Message box is placed between this "Else" and it's corresponding "Endif" as below , which will contain the following Message to be displayed



- At this point our script looks as below :

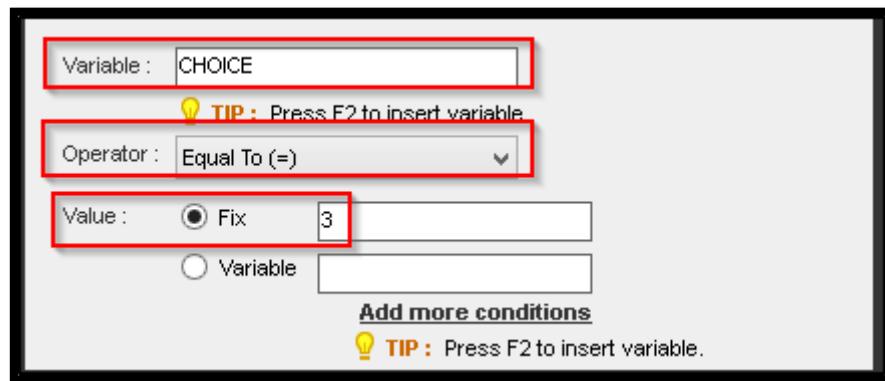
```

1  [Excel] Open Spreadsheet "E:\Ajay.Mehta\Desktop\Training\FilesNFolders\FILESNFOLDERS.xlsx". ActiveSheet: "Default". Session: Default
2  [Excel] Go to cell "A1". Session: Default
3  [Start Loop] "Each File In Folder E:\Ajay.Mehta\Desktop\Training\FilesNFolders\FOLDERTOORGANIZE"
4    [Comment: Please enter your commands to loop. Use $filename$.$extension$ variable for each file name in the Loop.]
5    [Excel] Set value of Active Cell with "$CurrentDirectory$". Session: Default
6    [Excel] Go to one cell right of active cell. Session: Default
7    [Excel] Set value of Active Cell with "$FileName$.$Extension$". Session: Default
8    [Excel] Go to one cell below the active cell. Session: Default
9    [Excel] Go to beginning of the row of the active cell. Session: Default
10   [Prompt Message: "$FileName$.$Extension$ : 1)COPY 2)RENAME 3)DELETE" in "Don't Write In Any Window" Assign value to Variable: $CHOICE$
11   [Comment: Please enter the conditional commands here.]
12  [IF] If $CHOICE$ Equal To (=) "1" Then
13    [Prompt For Folder: "Which folder to copy to " for Folder Assign to variable "$COPYFOLDER$"]
14    [Copy Files "$CurrentDirectory$\$FileName$.$Extension$" to "$COPYFOLDER$"]
15    [IF] If File Exists ("$COPYFOLDER$\$FileName$.$Extension$") Then
16      [Comment: Please enter the conditional commands here.]
17      [Message Box: "File Copy Success"]
18    [Else]
19      [Message Box: "File Copy Failed"]
20    [End If]
21  [End If]

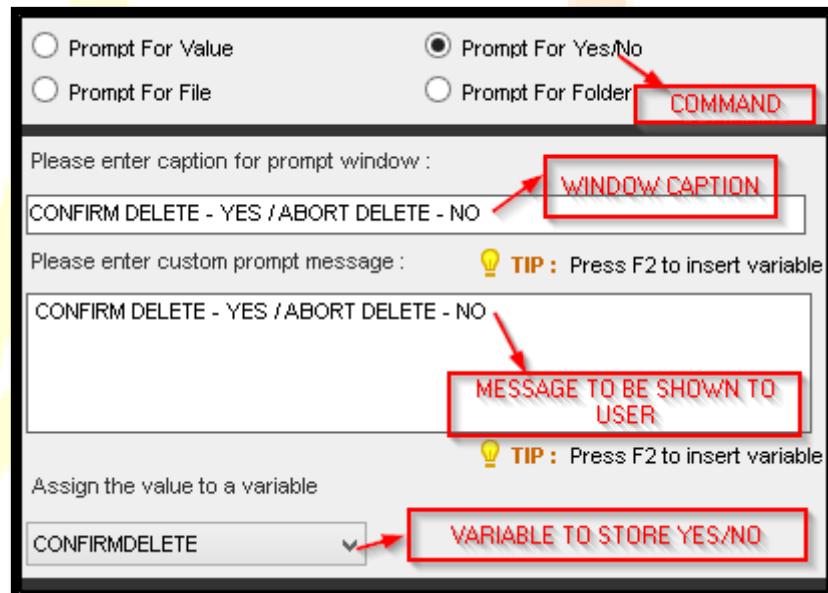
22  [IF] If $CHOICE$ Equal To (=) "2" Then
23    [Comment: Please enter the conditional commands here.]
24    [Prompt Message: "$FileName$.$Extension$ : New File Name" in "Don't Write In Any Window" Assign value to Variable: $RENAMEFILE$]
25    [Rename Files "$CurrentDirectory$\$FileName$.$Extension$" to "$RENAMEFILE$"]
26    [IF] If File Exists ("$CurrentDirectory$\$RENAMEFILE$") Then
27      [Comment: Please enter the conditional commands here.]
28      [Message Box: "File Renamed Successfully"]
29    [Else]
30      [Message Box: "File was not renamed"]
31    [End If]
32  [End If]
33  [End Loop]
34  [Excel] Close Spreadsheet. Session: Default

```

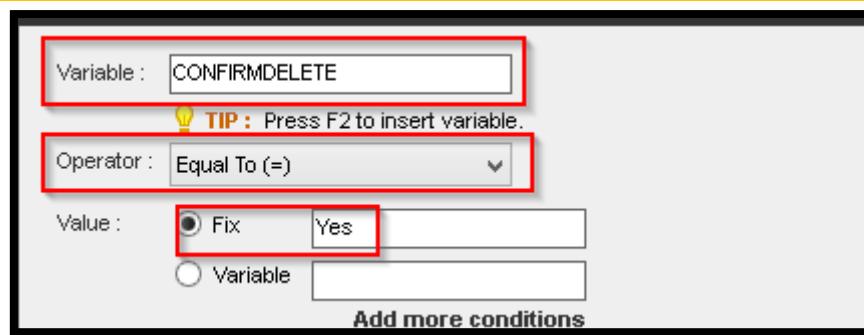
- Next we need to check for the condition where “Choice = 3” , i.e user chooses to delete file as below , Again we choose the “If/Else – Variable” command , which opens the following dialog as below :



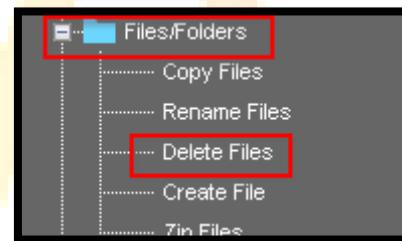
- Now we want to double check with the user if he wants to actually delete for the same, we drag an “Prompt – Prompt For Yes No” command and make entries as below



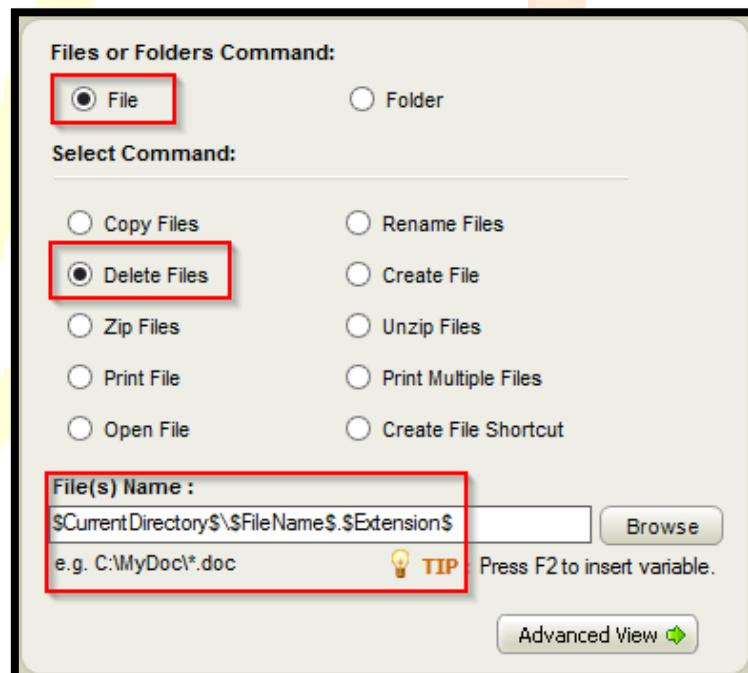
- We will then check using “If Else – Variable “ whether user chose “Yes” as below



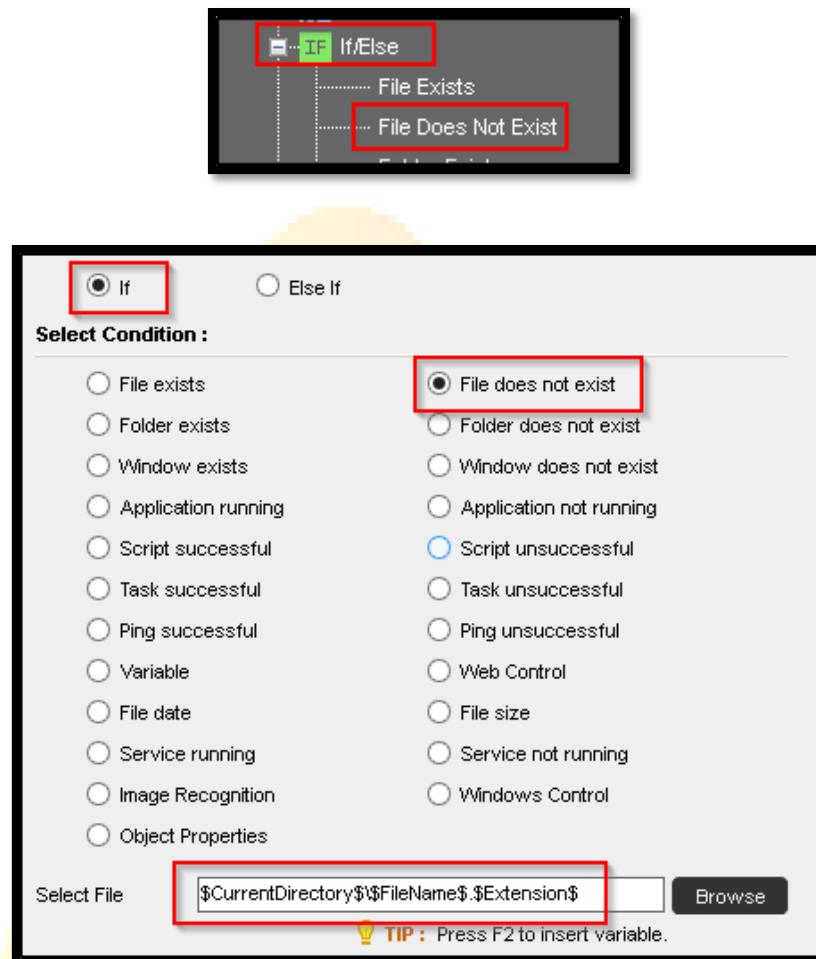
- If so , we will choose the “Files/Folders – Delete Files” command as below



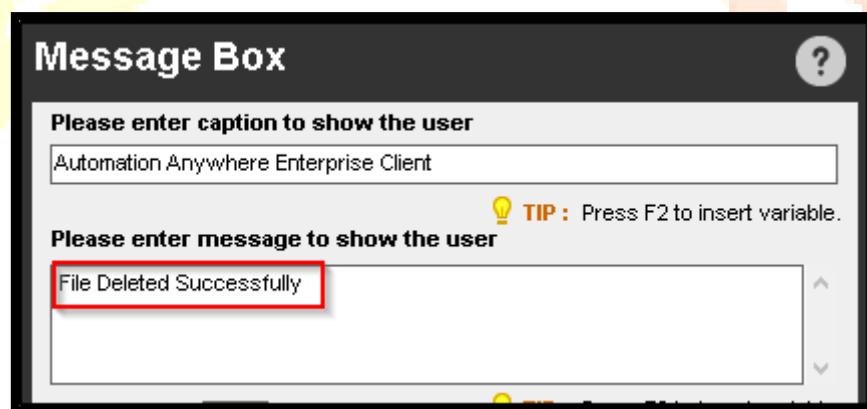
- This opens the following dialog as below



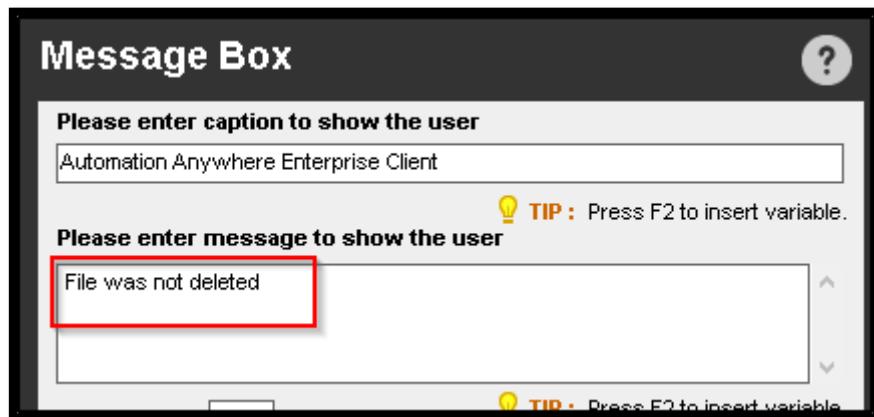
- Next , we check if the file did get deleted or not using “If/Else – File Does Not Exist” command as below :



- Next we put in a message box , to display the message if the condition is true , following are the contents of our message box as below



- Next we put in a “Else” part , for the eventuality where the “If” part may not be true, subsequently we put in a message box for the “Else” part , contents of the Message box would be as below



- Our Script Looks as below

```

1  Excel: Open Spreadsheet "E:\Ajay.Mehta\Desktop\Training\FilesNFolders\FILESNFOLDERS.xlsx". ActiveSheet: "Default". Session: Default
2  Excel: Go to cell "A1". Session: Default
3  Start Loop "Each File In Folder E:\Ajay.Mehta\Desktop\Training\FilesNFolders\FOLDERTOORGANIZE"
4    Comment: Please enter your commands to loop. Use $filename$.$extension$ variable for each file name in the Loop.
5  Excel: Set value of Active Cell with "$CurrentDirectory$". Session: Default
6  Excel: Go to one cell right of active cell. Session: Default
7  Excel: Set value of Active Cell with "$FileName$.$Extension$". Session: Default
8  Excel: Go to one cell below the active cell. Session: Default
9  Excel: Go to beginning of the row of the active cell. Session: Default
10  Prompt Message: "$FileName$.$Extension$ : 1)COPY 2)RENAME 3)DELETE" in "Don't Write In Any Window" Assign value to Variable: $CHOICE$
11  Comment: Please enter the conditional commands here.
12  IF $CHOICE$ Equal To (=) "1" Then
13    Prompt For Folder: "Which folder to copy to " for Folder Assign to variable "$COPYFOLDER$"
14    Copy Files "$CurrentDirectory$\$FileName$.$Extension$" to "$COPYFOLDER$"
15    IF File Exists ("$COPYFOLDER$\$FileName$.$Extension$") Then
16      Comment: Please enter the conditional commands here.
17      Message Box: "File Copy Success"
18    ELSE
19      Message Box: "File Copy Failed"
20    End If
21  End If

```

```

22  IF      If $CHOICE$ Equal To (=) "2" Then
23  | Comment: Please enter the conditional commands here.
24  | A Prompt Message: "$FileName$.{$Extension$} : New File Name" in "Don't Write In Any Window" Assign value to Variable: $RENAMEFILE$
25  | F Rename Files "$CurrentDirectory$\{$FileName$.{$Extension$}" to "$RENAMEFILE$"
26  IF      If File Exists ("$CurrentDirectory$\{$RENAMEFILE$") Then
27  | Comment: Please enter the conditional commands here.
28  | M Message Box: "File Renamed Successfully"
29  ELSE
30  | M Message Box: "File was not renamed"
31  | X End If
32  | X End If
33  IF      If $CHOICE$ Equal To (=) "3" Then
34  | Comment: Please enter the conditional commands here.
35  | A Prompt For Yes/No: "CONFIRM DELETE - YES / ABORT DELETE - NO" for File Assign to variable "$CONFIRMDELETE$"
36  IF      If $CONFIRMDELETE$ Equal To (=) "Yes" Then
37  | Comment: Please enter the conditional commands here.
38  | F Delete Files "$CurrentDirectory$\{$FileName$.{$Extension$}"
39  IF      If File Does Not Exist ("$CurrentDirectory$\{$FileName$.{$Extension$}") Then
40  | Comment: Please enter the conditional commands here.
41  | M Message Box: "File Deleted Successfully"
42  ELSE
43  | M Message Box: "File was not deleted"
44  | X End If
45  | X End If
46  | X End If
47  | X End Loop

```

48 Excel: Close Spreadsheet. Session: Default

- Next we “Save Task” and then “Run Task”, the script picks up every file in the folder, and based on user choice , proceeds to “Copy/Rename/Delete” the file. Following which it confirms whether the action was completed successfully or not.

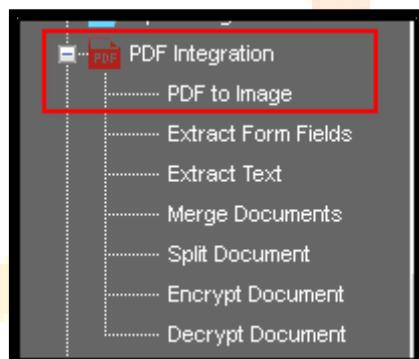
## 7) PDF Integration

### Objective

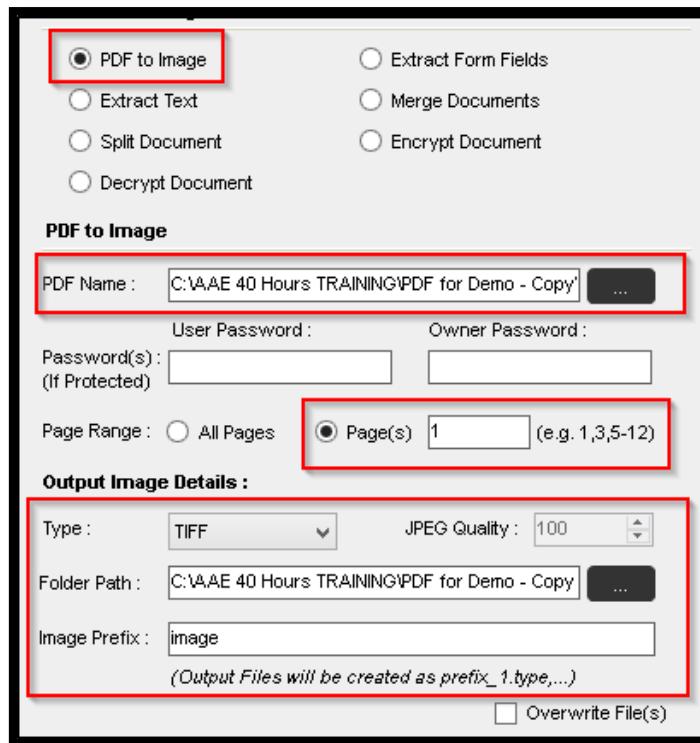
- Step by step demonstration of PDF Integration and its various facets, like
  - a) Converting a PDF document into an Image.
  - b) Extracting Data from PDF forms.
  - c) Extracting Text from PDF documents.
  - d) Merging and Splitting PDF documents.
  - e) Encrypting and Decrypting PDF documents.

### Example: PDF Integration

- As usual we will first of all proceed to open the Automation Anywhere Enterprise client after ascertaining that the Automation Anywhere Control Room Service is up and running.
- We open the Task Editor to create a new task for this particular activity.
- Next we point to the PDF Integration commands as seen here



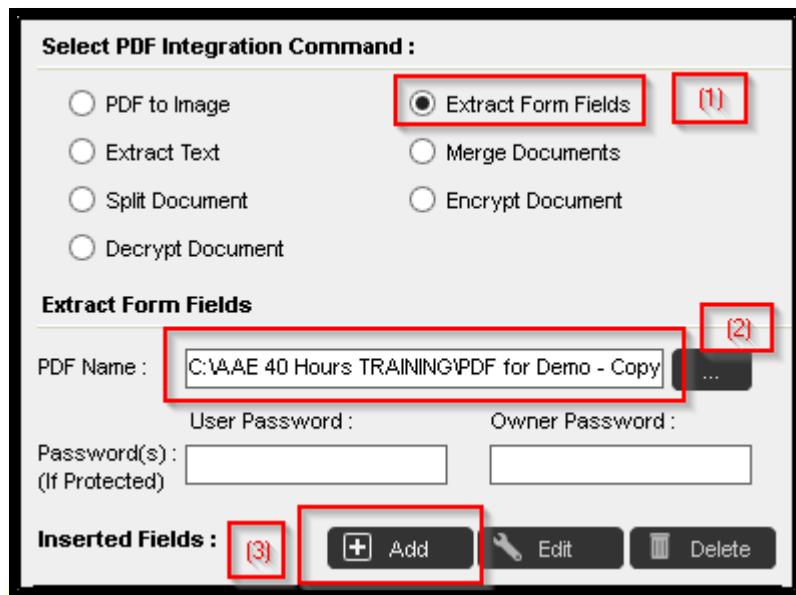
- We will start with “PDF Integration – PDF to Image” command , as seen below



- Here we specify Page(s) that need to be converted to Image. Every page will be a separate image that gets created.
- We can specify what is the Type/Format of the image that we want to be created.
- In case there are multiple Images, the Image Prefix will throw a new image for every page on the lines of ImagePrefix\_1.type, ImagePrefix\_2.type and so on.
- The above command will extract specified pages from above document and save them as image's.
- Next, we want to extract certain specific portions out of our pdf document as below, and store them into variables.

Given Name:	Automation		
Family Name:	Anywhere		
Address 1:	Vadodara	House nr:	
Address 2:	Vadodara		

- For this we will call the “PDF Integration – Extract Form Fields” command, and begin the data extraction process as below, fill in data in the sequence Specified below:



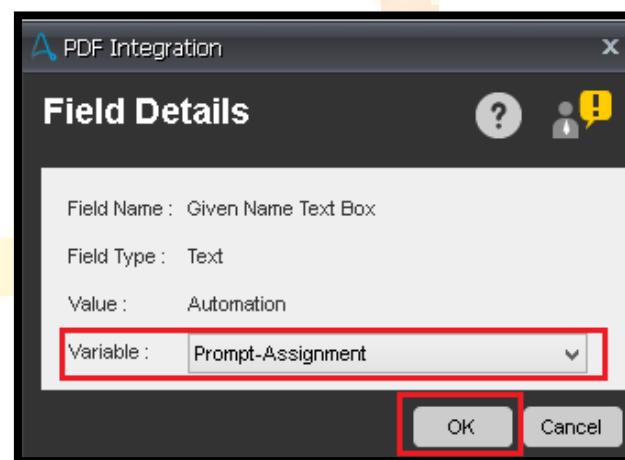
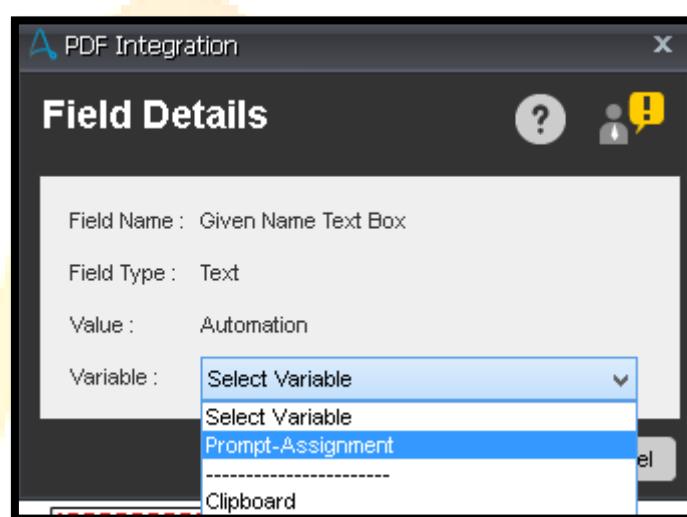
- Clicking on “Add” opens the following dialog, where we specify which data which must be extracted

Given Name:	Automation
Family Name:	Anywhere
Address 1:	Vadodara
Address 2:	Vadodara
Postcode:	
	City: Vadodara

- Right click on the field which needs to be extracted and choose “Add Field”, which then allows you to store the data in the variable of your choice as below

Given Name:	Automation	Add Field
Family Name:	Anywhere	

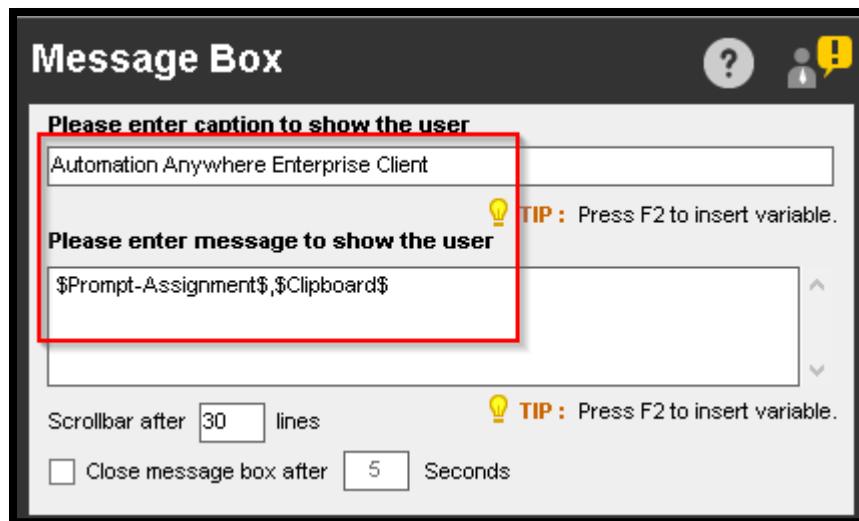
- The following dialog pops up on choosing “Add Field”, as below



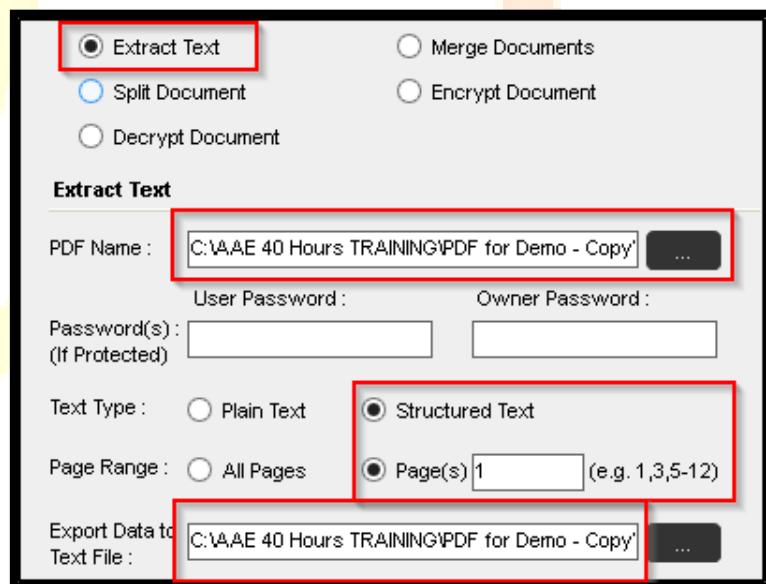
- We can see our value getting stored inside the variable we specified as below

Inserted Fields :			
Field Name	Field Type	Field Value	Select Variab
Given Nam...	Text	Automation	Prompt-Assi

- We next call in a message box to display the value that we captured inside the variables in the previous step, as below

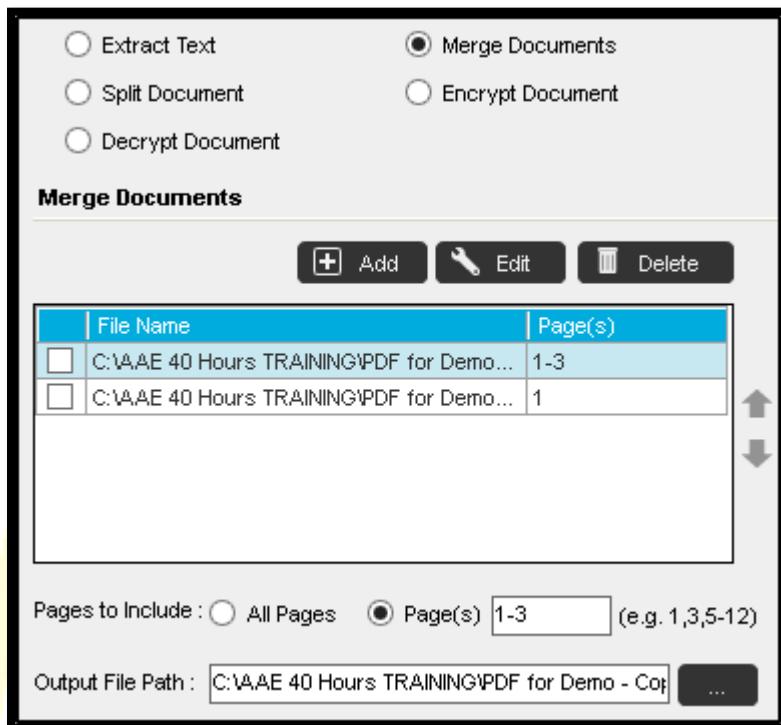


- Next we want to extract all the text from certain pages of a pdf document and store it as a .txt document , for which we use the “PDF Integration – Extract Text” command as below

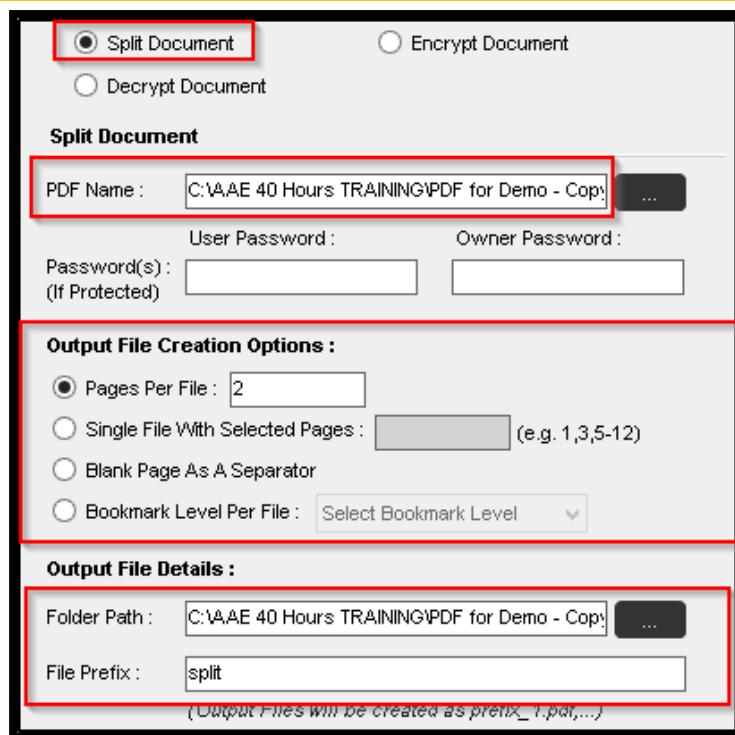


- As we can see, in the above we specified the document name, the Text Type of the document (Structured or Plain), number of pages to extract, and the location where we want the Text file to be created.

- Next, we want to take a couple of documents, choose specific pages from them and create a new document from the same, for this we will use the “PDF Integration – Merge Documents” command, as below:



- In the above image , we keep clicking on “Add” button to keep adding documents , subsequently , we click on each document picked up and specify whether we want to Pick up “All Pages” or selective “Page(s)” from the document in perspective.
- Next, we will look at the “PDF Integration – Split Command”, meant to break down a PDF into smaller pdf's as per user specification, as seen here



- While splitting , we can split on the following lines :
  - a) Pages per file : Will try to split based on number of Pages specified , i.e create pdf's of that many number of pages.
  - b) Single File With Selected Pages : Creates a single file composed of pages chosen.
  - c) Blank Page As A Seperator : Breaks into smaller documents using blank pages within the PDF document as a marker.
  - d) Bookmark Level Per File : Splits based on Bookmarks if they exist in the PDF .
  - e) File Prefix: Let's Say we give the prefix as "split", then in case multiple pdfs are created they will be named as split\_1, split\_2, split\_n .....
- Next, we want to encrypt a pdf document, by setting user and owner level passwords and assigning permissions “PDF Integration – Encrypt Document”, as below

Split Document       Encrypt Document

Decrypt Document

**Encrypt Document**

PDF to Encrypt : C:\AAE 40 Hours TRAINING\PDF for Demo - Co

User Password :   
Set Password(s) :

Owner Password :

Select User Permissions to Apply :

<input checked="" type="checkbox"/> Print	<input checked="" type="checkbox"/> Form Fill	<input checked="" type="checkbox"/> Annotation
<input checked="" type="checkbox"/> Modify	<input checked="" type="checkbox"/> Document Assembly	<input checked="" type="checkbox"/> Accessibility
<input checked="" type="checkbox"/> Copy		

Select Encryption Level :

RC4 40-bit       RC4 128-bit       AES 128-bit

Save Encrypted PDF as : C:\AAE 40 Hours TRAINING\PDF for Demo - Co

- a) Owner Password : The creator/owner of the PDF will be having this password , which allows him/her to have all access to the PDF.
- b) User Password : This password will be assigned by the owner to his/her team members who need to be able to access this pdf based on user permissions applicable.
- An, encrypted PDF document can be decrypted later using “PDF Integration – Decrypt Document” if required

Decrypt Document

**Decrypt Document**

PDF to Decrypt : C:\AAE 40 Hours TRAINING\PDF for Demo - Cop

User/Owner Password :

Save Decrypted PDF As : C:\AAE 40 Hours TRAINING\PDF for Demo - Cop

Overwrite File

- For decrypting the pdf , the User/Owner password , having required permissions will have to be supplied .
- Save and Run this script, we should get the desired output. Our script looks like as below

```

1 PDF Integration: Convert "C:\AAE 40 Hours TRAINING\PDF for Demo - Copy\22_43_23.pdf" to images as "image" under folder "C:\AAE 40 Hours TRAINING\PDF for Demo - Copy"
2 PDF Integration: Extract Form Fields from "C:\AAE 40 Hours TRAINING\PDF for Demo - Copy\DoPdfFormExample(1).pdf"
3 Message Box: "$PromptAssignment$.$Clipboard$"
4 PDF Integration: Extract Text from "C:\AAE 40 Hours TRAINING\PDF for Demo - Copy\22_43_23.pdf" to "C:\AAE 40 Hours TRAINING\PDF for Demo - Copy\extracted.txt"
5 PDF Integration: Merge Documents into "C:\AAE 40 Hours TRAINING\PDF for Demo - Copy\merged.pdf"
6 PDF Integration: Split "C:\AAE 40 Hours TRAINING\PDF for Demo - Copy\merged.pdf" into files as "split" under folder "C:\AAE 40 Hours TRAINING\PDF for Demo - Copy"
7 PDF Integration: Encrypt Document "C:\AAE 40 Hours TRAINING\PDF for Demo - Copy\merged.pdf"; Save it as "C:\AAE 40 Hours TRAINING\PDF for Demo - Copy\encrypted.pdf"
8 PDF Integration: Decrypt Document "C:\AAE 40 Hours TRAINING\PDF for Demo - Copy\encrypted.pdf"; Save it as "C:\AAE 40 Hours TRAINING\PDF for Demo - Copy\decrypted.pdf"

```

## 8) String Operations

### Objective

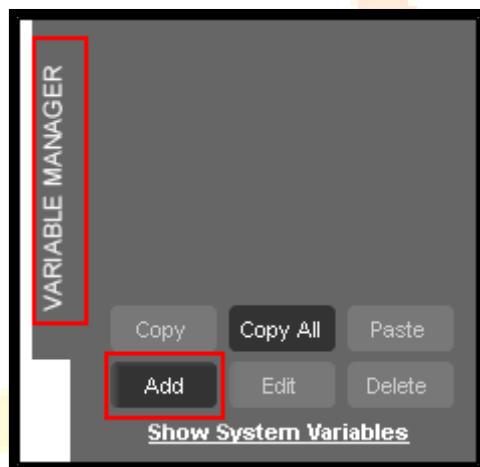
- Step by step demonstration of String Operation Commands.

### Example: String Operation Commands

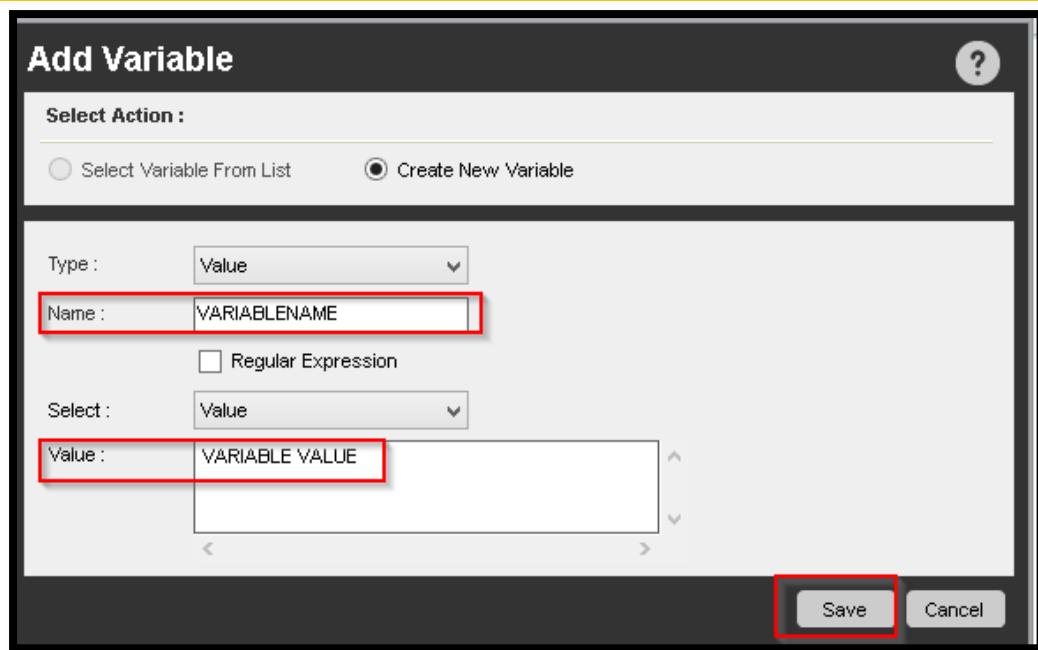
- In a New Task in the Task Editor, create 5 Variables and put values in them as below

- 1) STRING1 = ABCD 1234 EFGH
- 2) STRING2 = ABCD
- 3) STRING3 = EFGH
- 4) STRING4 = 1234
- 5) STRING5 = ABCD, EFGH, IJKL, MNOP, QRST

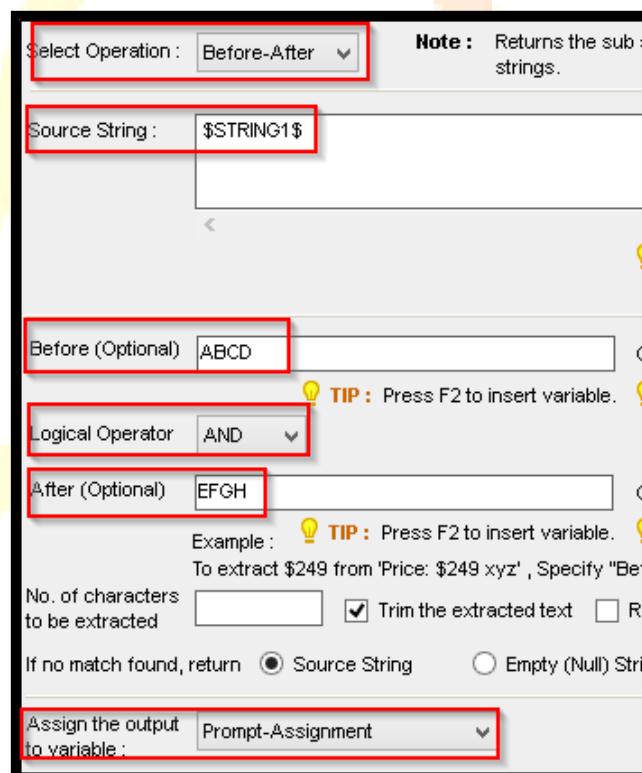
- To Create a Variable process is
- Choose “Add” in Variable Manager



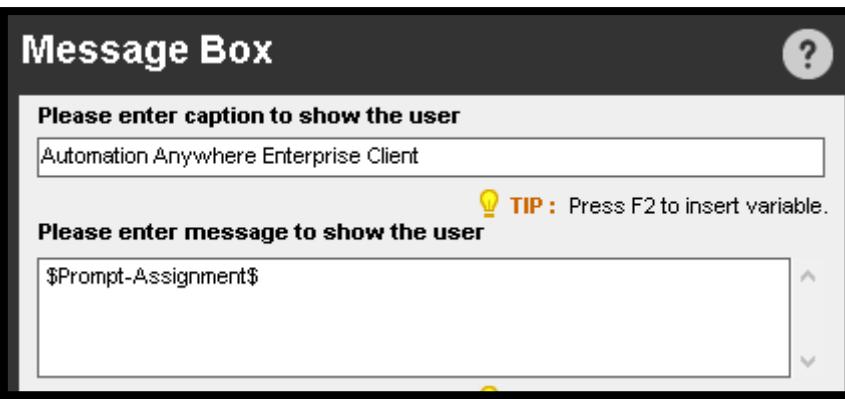
- Enter Variable Name and Value as shown , do it for all 5 variables above



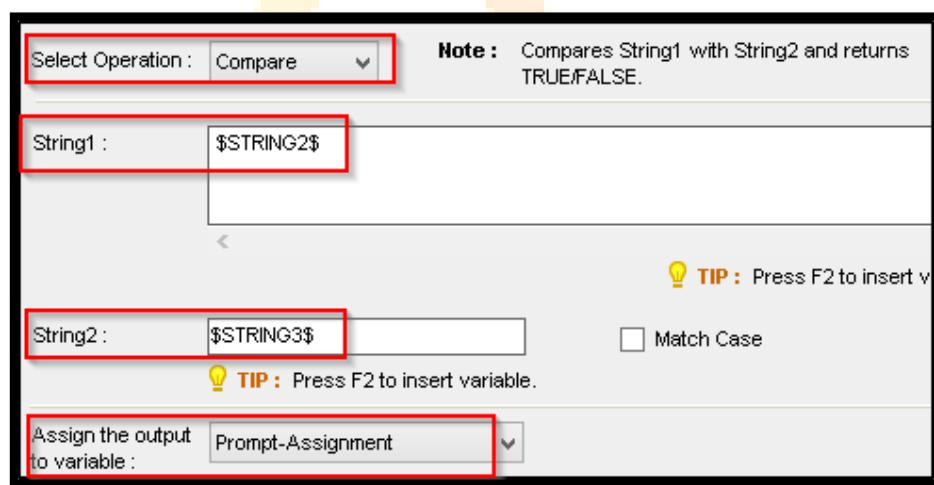
- Drag a “Before-After” command to the right



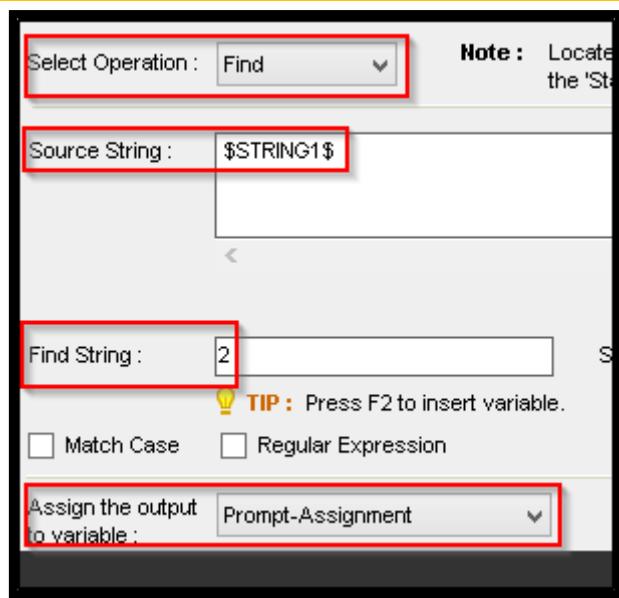
- Drag a Message Box, display prompt Assignment, it should show “1234” when you save and run the script



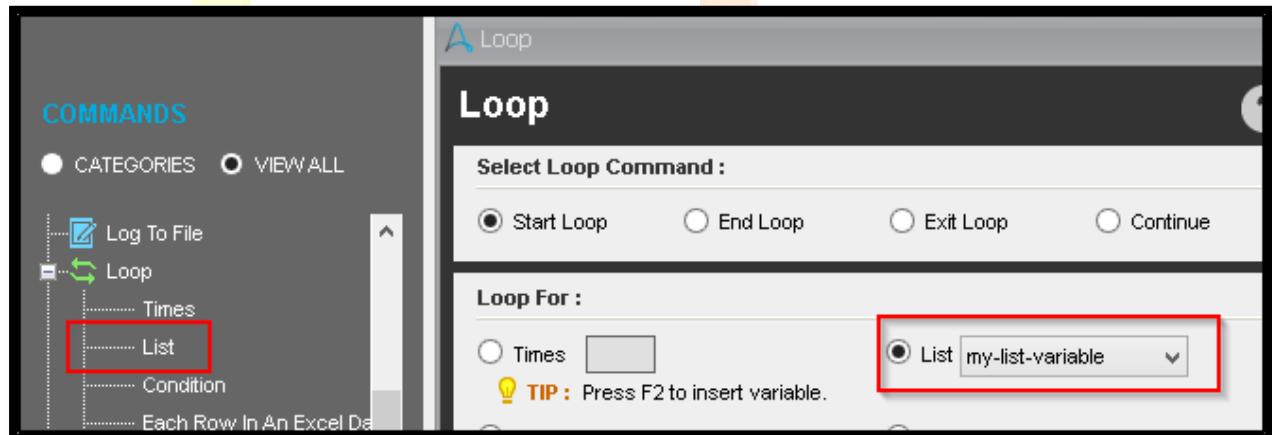
- Next drag a “Compare” command to the right as below



- As done previously display the value of “Prompt-Assignment” in a Message box, output for above would be “FALSE”  
Note: Select line 3, Right Click – Run from This Step, to just see output of the above.
- Drag a Find command to the right next in the Task Editor as below



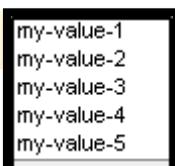
- As done previously display the value of “Prompt-Assignment” in a Message box, output for above would be 7
- Note: Select line 5, Right Click – Run from This Step, to just see output of the above.
- Drag a “Loop-List” to the Task Editor as below



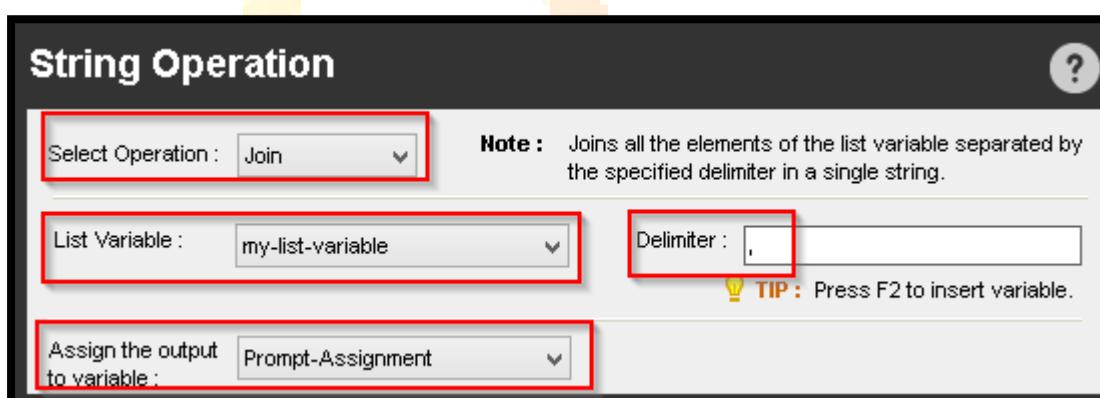
- In the loop that shows up, drag a Message box in the loop, display the value of “my-list-variable” as below



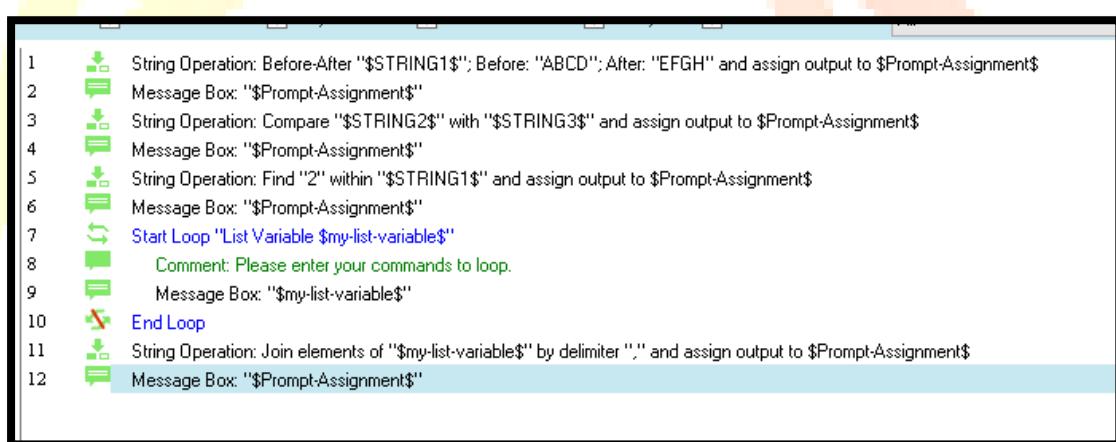
- It will show the following values one row at a time



- Next Drag a “String Operations - Join” command to the right as below

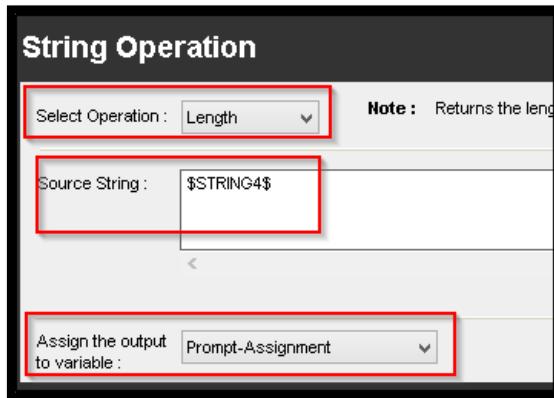


- As previously done we will display the value of Prompt-Assignment in a Message box, it will show “myvalue-1, -----, myvalue-5”

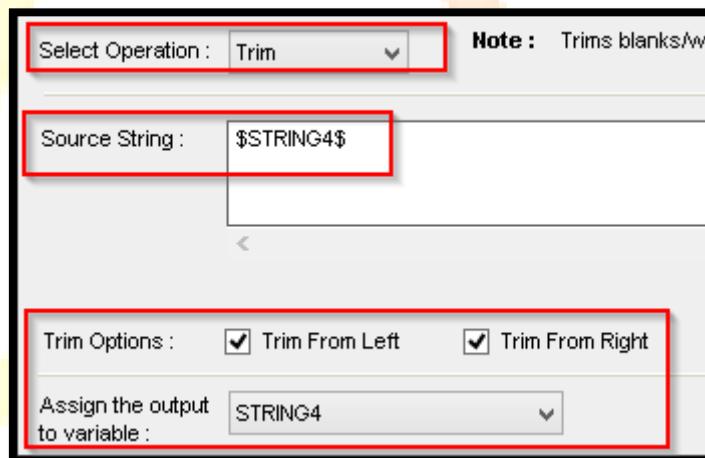


- Note: Right-Click on Line 11, Run from this step just to see above result

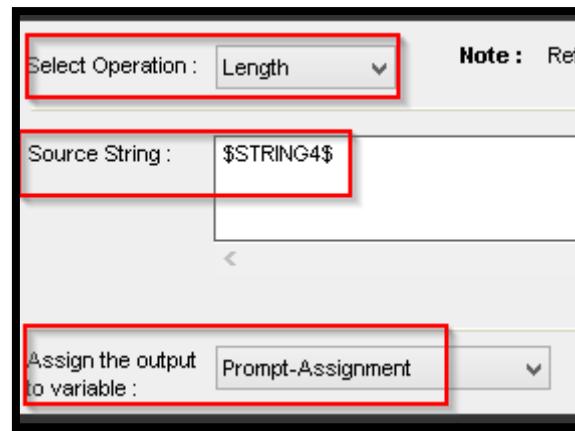
- Next Drag a “Length” command to the right as below



- Again, display “Prompt-Assignment” in a Message Box, depending on spaces put in we should get some number larger than 4.
- Right Click Line 13 – Run from This step to get last results only
- Next we call a ‘Trim’ command to the right as below



- Then we recall the “Length” command as below



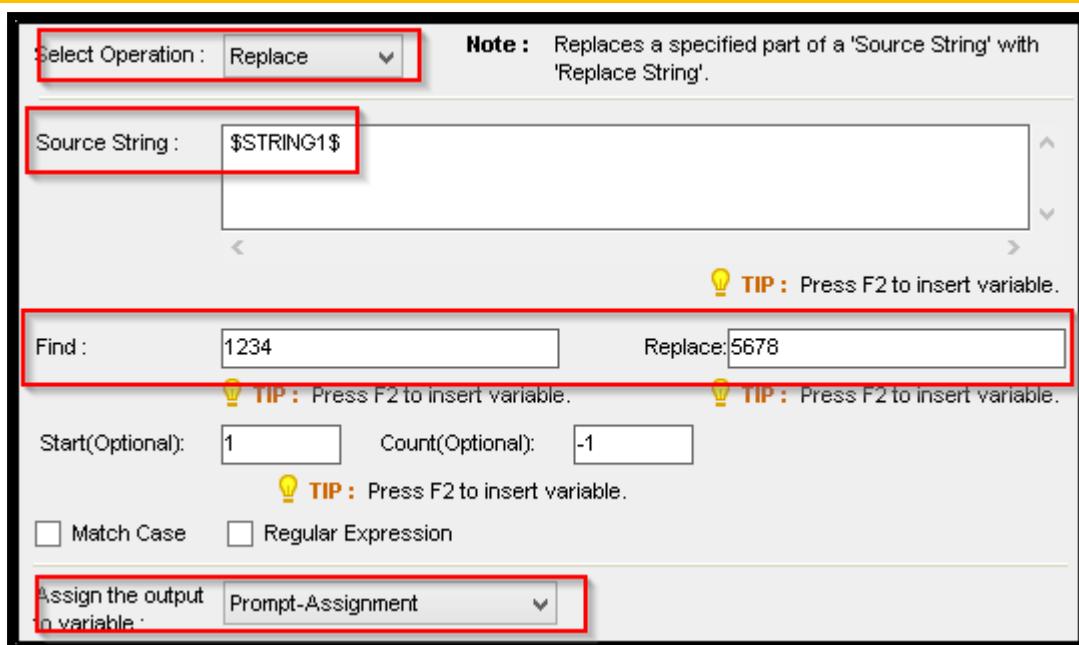
- Then we call a Message box and display Prompt-Assignment (Displays 4) as usual, our script at this point is as below

```

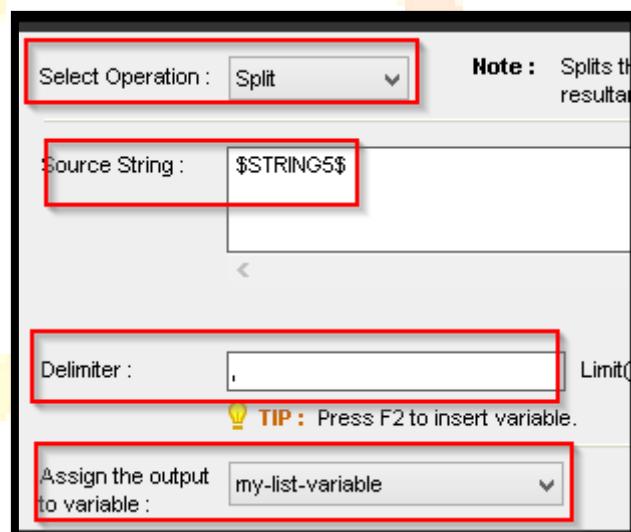
1  ↴ String Operation: Before-After "$STRING1$"; Before: "ABCD"; After: "EFGH" and assign output to $Prompt-Assignment$
2  ↴ Message Box: "$Prompt-Assignment$"
3  ↴ String Operation: Compare "$STRING2$" with "$STRING3$" and assign output to $Prompt-Assignment$
4  ↴ Message Box: "$Prompt-Assignment$"
5  ↴ String Operation: Find "2" within "$STRING1$" and assign output to $Prompt-Assignment$
6  ↴ Message Box: "$Prompt-Assignment$"
7  ↳ Start Loop "List Variable $my-list-variable$"
8    ↴ Comment: Please enter your commands to loop.
9    ↴ Message Box: "$my-list-variable$"
10   ↳ End Loop
11  ↴ String Operation: Join elements of "$my-list-variable$" by delimiter "," and assign output to $Prompt-Assignment$
12  ↴ Message Box: "$Prompt-Assignment$"
13  ↴ String Operation: Get length of "$STRING4$" and assign output to $Prompt-Assignment$
14  ↴ Message Box: "$Prompt-Assignment$"
15  ↳ String Operation: Trim "$STRING4$" and assign output to $STRING4$ (Line 15)
16  ↴ String Operation: Get length of "$STRING4$" and assign output to $Prompt-Assignment$
17  ↴ Message Box: "$Prompt-Assignment$"

```

- To get last output only, run from line 15
- Next Drag a “Replace Command” as below



- As usual display "Prompt-Assignment" in Message Box, Output = ABCD 5678 EFGH
- Next Drag a "Split" command to the right as below



- Next, Copy these lines

```

7  ↗ Start Loop "List Variable $my-list-variable$"
8  ↗ Comment: Please enter your commands to loop.
9  ↗ Message Box: "$my-list-variable$"
10 ↘ End Loop
    
```

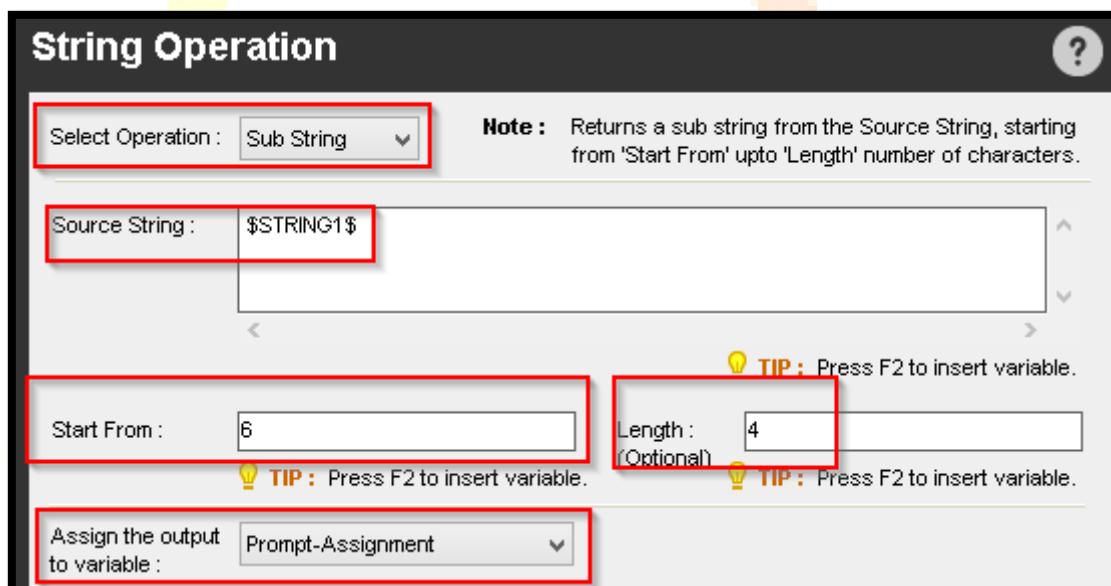
- From the top to just below our split command above, script should look like as below

```

1  String Operation: Before-After "$STRING1$"; Before: "ABCD"; After: "EFGH" and assign output to $Prompt-Assignment$
2  Message Box: "$Prompt-Assignment$"
3  String Operation: Compare "$STRING2$" with "$STRING3$" and assign output to $Prompt-Assignment$
4  Message Box: "$Prompt-Assignment$"
5  String Operation: Find "2" within "$STRING1$" and assign output to $Prompt-Assignment$
6  Message Box: "$Prompt-Assignment$"
7  Start Loop "List Variable $my-list-variable$"
8    Comment: Please enter your commands to loop.
9    Message Box: "$my-list-variable$"
10   End Loop
11   String Operation: Join elements of "$my-list-variable$" by delimiter "," and assign output to $Prompt-Assignment$
12   Message Box: "$Prompt-Assignment$"
13   String Operation: Get length of "$STRING4$" and assign output to $Prompt-Assignment$
14   Message Box: "$Prompt-Assignment$"
15   String Operation: Trim "$STRING4$" and assign output to $STRING4$
16   String Operation: Get length of "$STRING4$" and assign output to $Prompt-Assignment$
17   Message Box: "$Prompt-Assignment$"
18   String Operation: Replace "1234" with "5678" in "$STRING1$" and assign output to $Prompt-Assignment$
19   Message Box: "$Prompt-Assignment$"
20   String Operation: Split "$STRING5$" with delimiter "," and assign output to $my-list-variable$
21   Start Loop "List Variable $my-list-variable$"
22     Comment: Please enter your commands to loop.
23     Message Box: "$my-list-variable$"
24   End Loop

```

- Run from Line 20, you should see the values of STRING5 overwriting the values of my-list-variable
- Next drag a “substring” to the right as below



- Again, display prompt-assignment in Message Box, it will show 1234, our script should be as below

```

1  String Operation: Before-After "$STRING1$"; Before: "ABCD"; After: "EFGH" and assign output to $Prompt-Assignment$
2  Message Box: "$Prompt-Assignment$"
3  String Operation: Compare "$STRING2$" with "$STRING3$" and assign output to $Prompt-Assignment$
4  Message Box: "$Prompt-Assignment$"
5  String Operation: Find "2" within "$STRING1$" and assign output to $Prompt-Assignment$
6  Message Box: "$Prompt-Assignment$"
7  Start Loop "List Variable $my-list-variable$"
8    Comment: Please enter your commands to loop.
9    Message Box: "$my-list-variable$"
10   End Loop
11   String Operation: Join elements of "$my-list-variable$" by delimiter "," and assign output to $Prompt-Assignment$
12   Message Box: "$Prompt-Assignment$"
13   String Operation: Get length of "$STRING4$" and assign output to $Prompt-Assignment$
14   Message Box: "$Prompt-Assignment$"
15   String Operation: Trim "$STRING4$" and assign output to $STRING4$
16   String Operation: Get length of "$STRING4$" and assign output to $Prompt-Assignment$
17   Message Box: "$Prompt-Assignment$"
18   String Operation: Replace "1234" with "5678" in "$STRING1$" and assign output to $Prompt-Assignment$
19   Message Box: "$Prompt-Assignment$"
20   String Operation: Split "$STRING5$" with delimiter "," and assign output to $my-list-variable$
21   Start Loop "List Variable $my-list-variable$"
22   Comment: Please enter your commands to loop.
23   Message Box: "$my-list-variable$"
24   End Loop
25   String Operation: Extract substring from "$STRING1$" and assign output to $Prompt-Assignment$
26   Message Box: "$Prompt-Assignment$"

```

- Run from line 25, to get last output.

## 9) PGP

### Objective

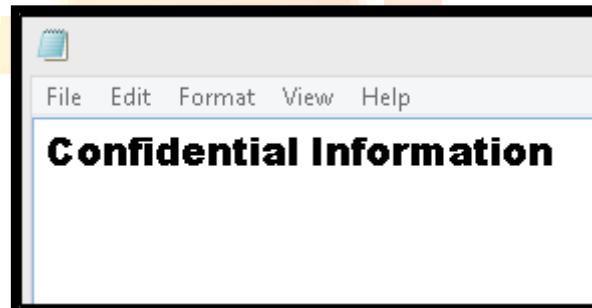
- Step by step demonstration of PGP and its various facets, like
  - a) Encrypting/Decrypting files with passwords.
  - b) Creating Public/Private key pairs.
  - c) Encrypting/Decrypting documents with key pairs.

### Example: PGP

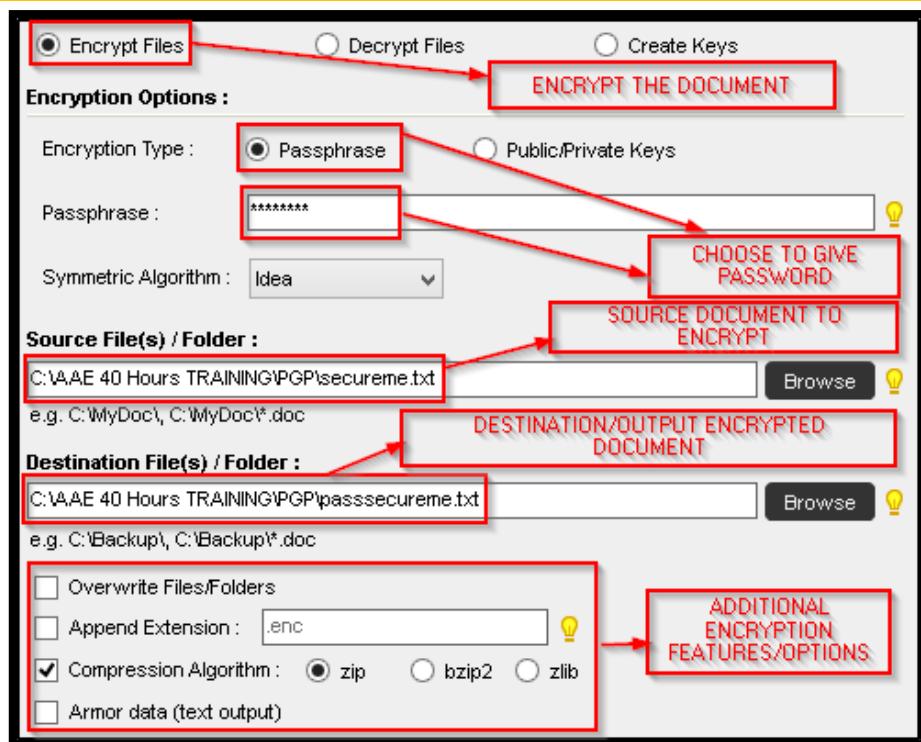
- As usual we will first proceed to open the Automation Anywhere Enterprise client after ascertaining that the Automation Anywhere Control Room Service is up and running.
- We open the Task Editor to create a new task for this activity.
- Next we point to the PGP commands as seen here below



- Now, we want to encrypt the following document below



- We call the "PGP – Encrypt Document" command, which looks like as below:



- Specify the options as indicated in the dialog above . Save And Run .
- We open the encrypted document, as we can see below it is not readable

**œJ L̄ ð=♀Èy>ÒQ v.ÁT6?L}·ÖB!!ÁËZî>’Èâ"NËØ3Ó#où¤♀}jK  
÷ HâéþPc}Àctøëí¶žc LcmòBŽ[iÆçá—+NO~à|ç;÷**

- This encrypted document can be sent via digital medium to the recipient , the passphrase can be sent using a different digital medium.
- Next, we want to decrypt the document, using “PGP – Decrypt Files “as below:

**Decryption Options :**

Encryption Type :	<input checked="" type="radio"/> Passphrase	<input type="radio"/> Public/Private Keys
Passphrase :	*****	

**Source File(s) / Folder :**

C:\AAE 40 Hours TRAINING\PGP\passsecureme.txt	Browse	
e.g. C:\MyDoc1, C:\MyDoc1*.doc		

**Destination File(s) / Folder :**

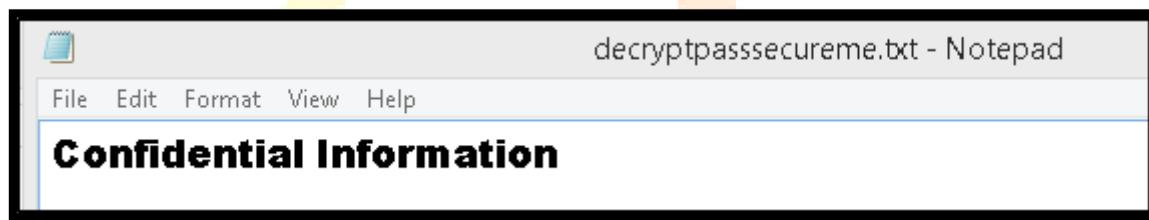
C:\AAE 40 Hours TRAINING\PGP\decryptpasssecureme.txt	Browse	
e.g. C:\Backup1, C:\Backup1*.doc		

**PASSWORD FOR DECRYPTION**

**ENCRYPTED DOCUMENT TO DECRYPT**

**DECRYPTED DOCUMENT**

- We run from this line , and get to see our decrypted document as below, which is readable. Using steps mentioned above recipient can get hold of the original document which he/she was meant to receive as below



- Next, we want to generate the Public/Private key pairs for encryption/decryption, for that we call the “PGP – Create keys” command as below:

**Select Option :**

<input type="radio"/> Encrypt Files	<input type="radio"/> Decrypt Files	<input checked="" type="radio"/> Create Keys
-------------------------------------	-------------------------------------	--

**Key File Details :**

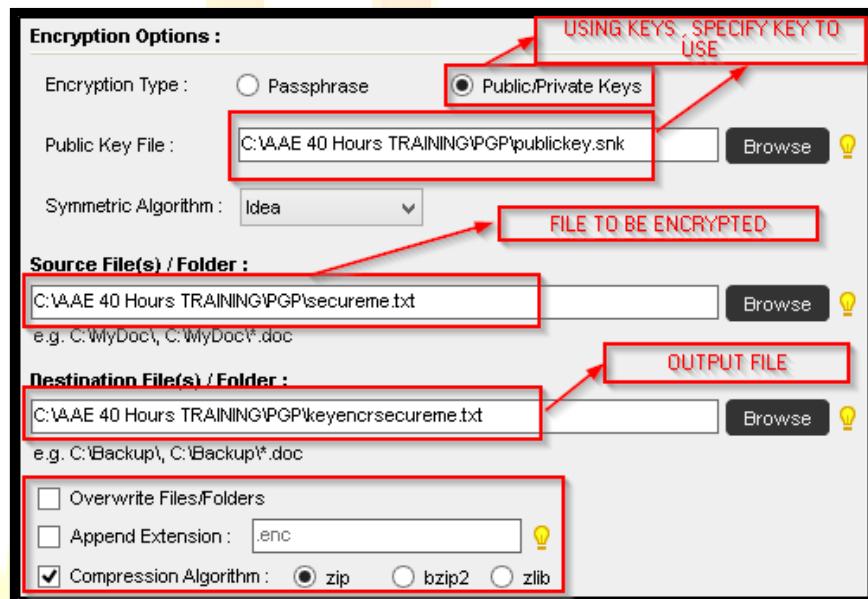
Public Key File :	C:\AAE 40 Hours TRAINING\PGP\publickey.snk	Browse
Private Key File :	C:\AAE 40 Hours TRAINING\PGP\privatekey.snk	Browse
Password : (Optional)	*****	<input type="checkbox"/> Overwrite Files

- The public and private key pairs are specific to each other , a file encrypted by a public key can be decrypted by it's specific private key only . The password is optional .

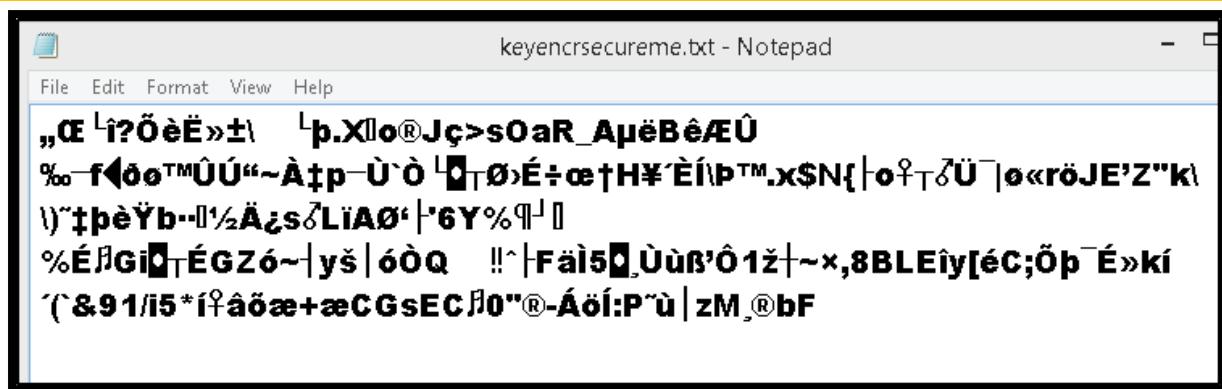
- This will generate the key pairs as below:

 privatekey.snk	7/21/2016 8:16 PM	Visual Studio Strong Name Key File	1 KB
 publickey.snk	7/21/2016 8:16 PM	Visual Studio Strong Name Key File	1 KB

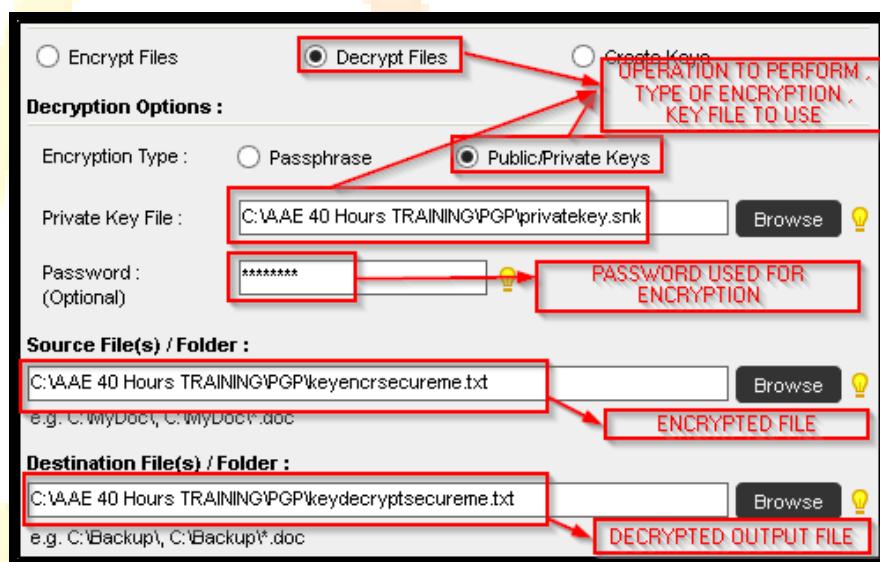
- Next, we want to encrypt and decrypt our documents using key pairs.
- We call the “PGP – Encrypt Files” command, and proceed to encrypt using public key as below



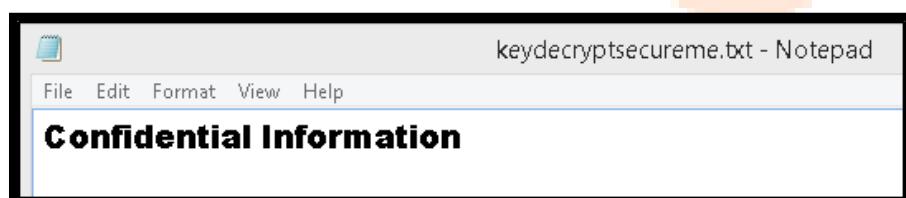
- The above step would be performed by the sender before sending the document over digital medium.
- Save And Run from last line, we get the following encrypted file as output below



- The above document with his private key and password would be sent to the receiver.
- Now, we want to decrypt this document, for which we use the “PGP – Decrypt files” command, and specify options as below



- The receiver will decrypt the document using above illustrated step.
- Save and Run from last line, we get the decrypted file which is readable.



- Our Script Looks like as below

```
1  ↗ PGP: Encrypt Files using Passphrase; Source: "C:\AAE 40 Hours TRAINING\PGP\secureme.txt"; Destination: "C:\AAE 40 Hours TRAINING\PGP\passsecureme.txt"
2  ↗ PGP: Decrypt Files using Passphrase; Source: "C:\AAE 40 Hours TRAINING\PGP\passsecureme.txt"; Destination: "C:\AAE 40 Hours TRAINING\PGP\decryptpasssecureme.txt"
3  ↗ PGP: Create Keys; Public Key File: "C:\AAE 40 Hours TRAINING\PGP\publickey.snk"; Private Key File: "C:\AAE 40 Hours TRAINING\PGP\privatekey.snk"
4  ↗ PGP: Encrypt Files using Public Key File; Source: "C:\AAE 40 Hours TRAINING\PGP\secureme.txt"; Destination: "C:\AAE 40 Hours TRAINING\PGP\keyencrsecureme.txt"
5  ↗ PGP: Decrypt Files using Private Key File; Source: "C:\AAE 40 Hours TRAINING\PGP\keyencrsecureme.txt"; Destination: "C:\AAE 40 Hours TRAINING\PGP\keydecryptsecureme.txt"
```

## 10) Email Automation

### Objective

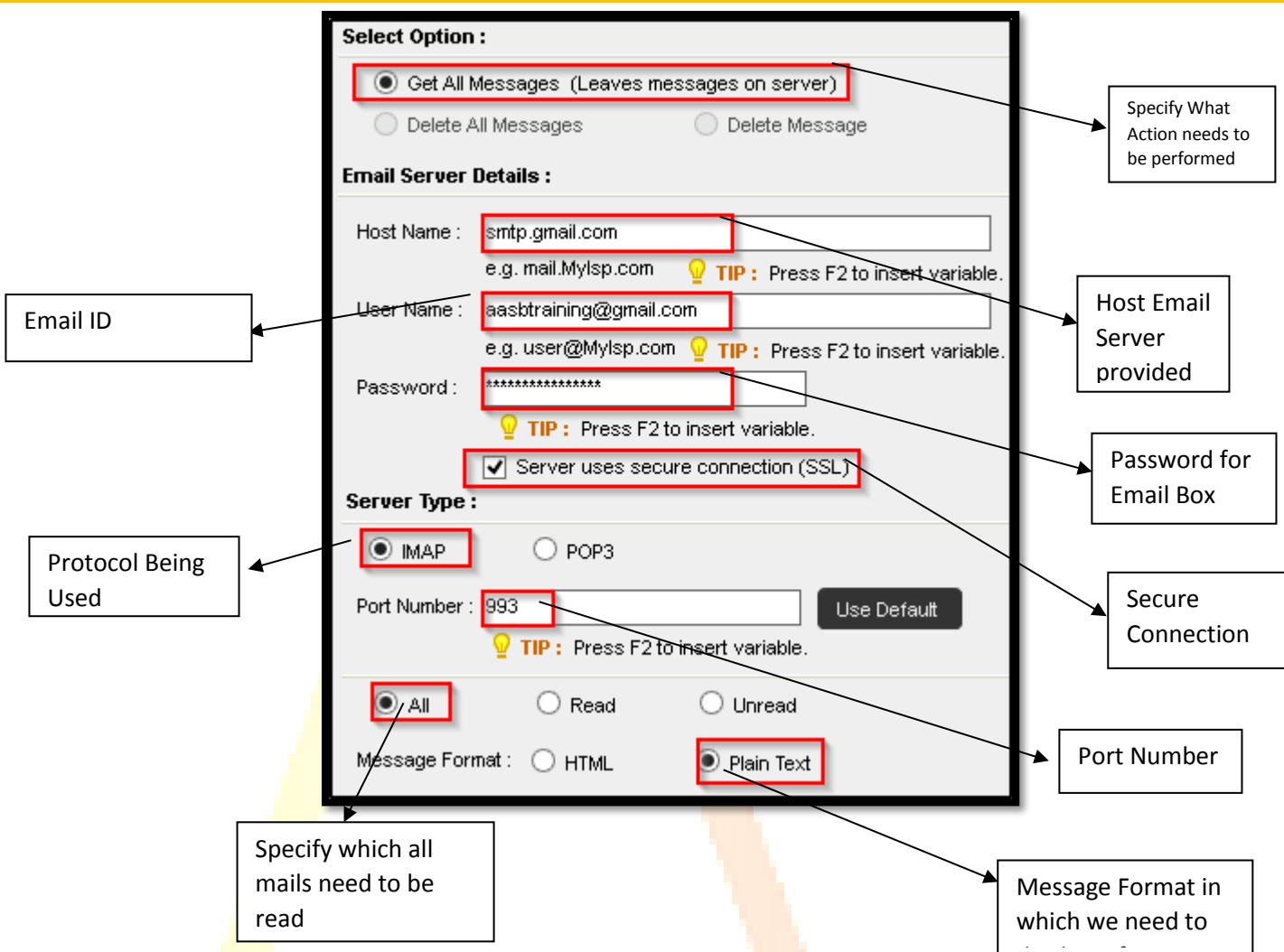
- Step by step demonstration of Email Automation and its various facets, like Trigger.

### Example: Email Automation

- As usual we will first proceed to open the Automation Anywhere Enterprise client after ascertaining that the Automation Anywhere Control Room Service is up and running.
- We open the Task Editor to create a new task for this activity.
- Let us open our Gmail box in the browser, we see the following screen as below:

Archive Report Spam Delete More Actions... Go Refresh		
1 - 50 of about 91 <a href="#">Older &gt;</a>		
<input type="checkbox"/> BrowserStack	Debugging on Mobile Devices Made Easy - Email not displaying correctly? View in browser.	Aug 9
<input type="checkbox"/> FlipHTML5	FlipHTML5 for Windows/Mac New Version Update - FlipHTML5 Release Notes Windows Version	Aug 9
<input type="checkbox"/> me	<b>Test</b>	Aug 4
<input type="checkbox"/> me	<b>Test</b>	Aug 4
<input type="checkbox"/> Quora Digest	What are the Indian petrol pump scams that everyone needs to be warned about? - Quora -	Aug 4
<input type="checkbox"/> me	Email configuration settings - Hello, Email configuration settings have been successfully validated.	Aug 3
<input type="checkbox"/> me	<b>Test</b>	Aug 3
<input type="checkbox"/> me	Automation Anywhere finished executing the task. - Message from Automation Anywhere. Task	Aug 2

- Our first objective is to forge a generic connection with this mail box , and iterate over it's contents , picking up all mails one by one and going over them , displaying their contents on the screen.
- Next, we want to drag an “Email Automation – Get All Messages” command to the right and specify, details about our mailbox, and connection details to access its content as below:



- This places a loop inside our script, where we will call a message box which will display the following information, as below

EMAIL FROM : \$Email From\$

---

EMAIL SUBJECT : \$Email Subject\$

---

EMAIL MESSAGE : \$Email Message\$

- We save and run the script, and we can see our mails bouncing off the screen one at a time as below

**EMAIL FROM :** BrowserStack <hello@browserstack.com>

**EMAIL SUBJECT :** Debugging on Mobile Devices Made Easy

**EMAIL MESSAGE :**

Hi Automation,

Introducing Developer Tools on Mobile Devices

We have now made it easier than ever to debug CSS and Javascript on mobile devices! We have expanded Chrome developer tools to debug websites on physical iOS and Android devices straight from your browser. No need to buy new devices and install extra drivers. Happy testing!

Try Devtools on iPhone 6S

[https://www.browserstack.com/start?e=aasbtraining%40gmail.com&utm\\_campaign=mobile-devtools&utm\\_medium=email&utm\\_source=newletter-mobile-devtools-jul2016-start-testing#device=iPhone%206S&device\\_browser=safari&os\\_version=9.0&openDevtools=true&start=true&os=ios](https://www.browserstack.com/start?e=aasbtraining%40gmail.com&utm_campaign=mobile-devtools&utm_medium=email&utm_source=newletter-mobile-devtools-jul2016-start-testing#device=iPhone%206S&device_browser=safari&os_version=9.0&openDevtools=true&start=true&os=ios)

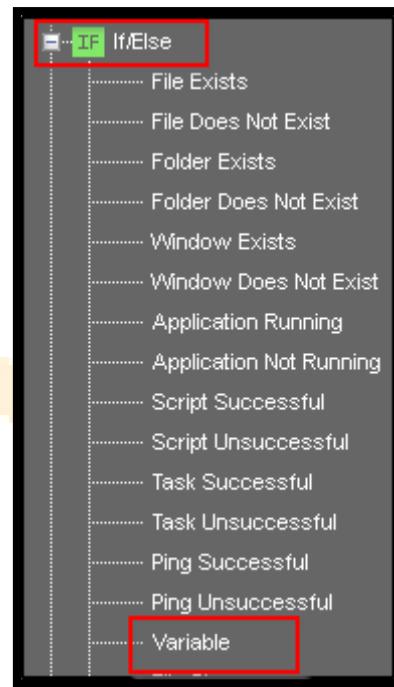
Happy Testing!  
The BrowserStack Team

Please unsubscribe if you prefer not to receive these emails:  
[https://www.browserstack.com/unsubscribe?email=aasbtraining%40gmail.com&token=532d7ba1afdb94156f93bafc7a2348ac&utm\\_campaign=mobile-devtools&utm\\_medium=email&utm\\_source=newletter-mobile-devtools-jul2016-unsubscribe](https://www.browserstack.com/unsubscribe?email=aasbtraining%40gmail.com&token=532d7ba1afdb94156f93bafc7a2348ac&utm_campaign=mobile-devtools&utm_medium=email&utm_source=newletter-mobile-devtools-jul2016-unsubscribe)

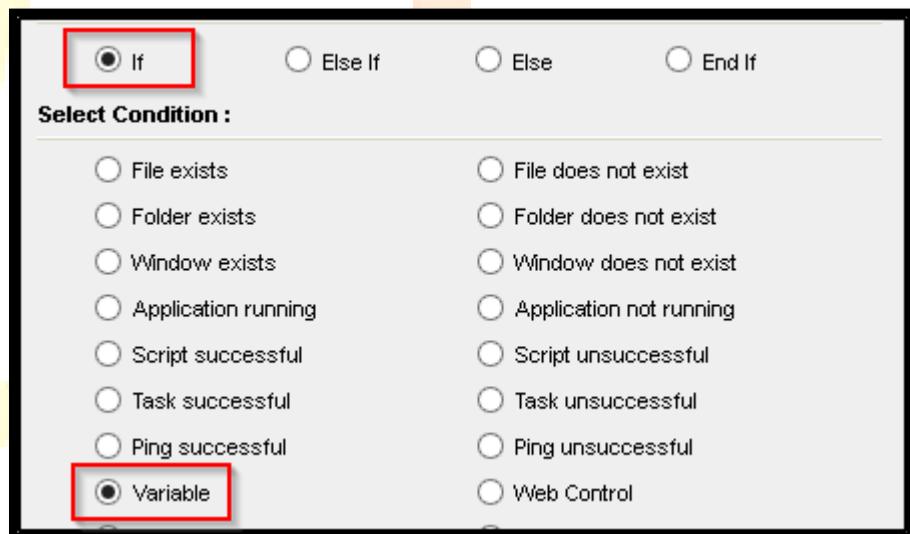
- We could successfully make a connection to our mail box and pick mails bouncing off the same.
- Now, we want to pick up only those mails which have “Test” keyword in the subject line anywhere, as seen below

		Archive	Report Spam	Delete	More Actions...	Go	Refresh	1 - 50 of about 91 <a href="#">Older &gt;</a>
<input type="checkbox"/>	BrowserStack	<a href="#">Debugging on Mobile Devices Made Easy</a> - Email not displaying correctly? View in browser.						Aug 9
<input type="checkbox"/>	FlipHTML5	<a href="#">FlipHTML5 for Windows/Mac New Version Update</a> - FlipHTML5 Release Notes Windows Version						Aug 9
<input type="checkbox"/>	me	<a href="#">Test</a>						Aug 4
<input type="checkbox"/>	me	<a href="#">Test</a>						Aug 4
<input type="checkbox"/>	Quora Digest	<a href="#">What are the Indian petrol pump scams that everyone needs to be warned about? - Quora -</a>						Aug 4
<input type="checkbox"/>	me	<a href="#">Email configuration settings</a> - Hello, Email configuration settings have been successfully validated.						Aug 3
<input type="checkbox"/>	me	<a href="#">Test</a>						Aug 3
<input type="checkbox"/>	me	<a href="#">Automation Anywhere finished executing the task.</a> - Message from Automation Anywhere. Task						Aug 2

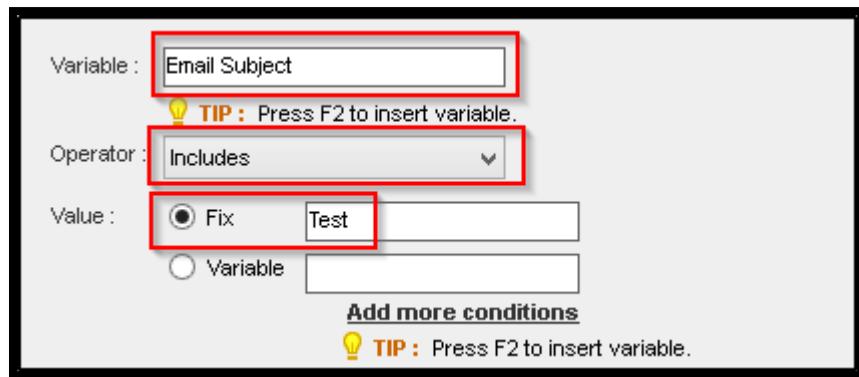
- For this, we bring in an “If/Else – Variable” command, as below



- The following dialog opens, as below



- We click on "Edit" in the dialog above, which opens the following window



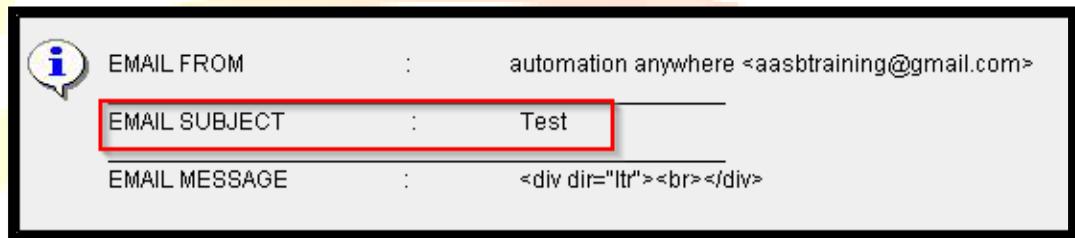
- Here we are specifying, that we want to pick only those mails with “Test” in their Subject line. Kindly note that we have specified the Operator as “Includes” and not as “Equal to (=)” which means that the “Test” keyword can be anywhere in the subject line.
- Our script looks like as below:

```

1  ↗ Start Loop 'Each message on server: smtp.gmail.com, User Name: aasbtraining@gmail.com, SSL, ServerType: IMAP, Message Format: Plain Text'
2  ↗ Comment: Please enter your commands to loop. Use Email Automation variables for each email on Server.(e.g. $Email From$, $Email To$, etc.)
3  ↗ IF '$Email Subject$ Includes: "Test" Then
4    ↗ Comment: Please enter the conditional commands here.
5  ↗ Message Box: "EMAIL FROM" : $Email From$ EMAIL SUBJECT : $Email Subject$_
6  ↗ End If
7  ↗ End Loop

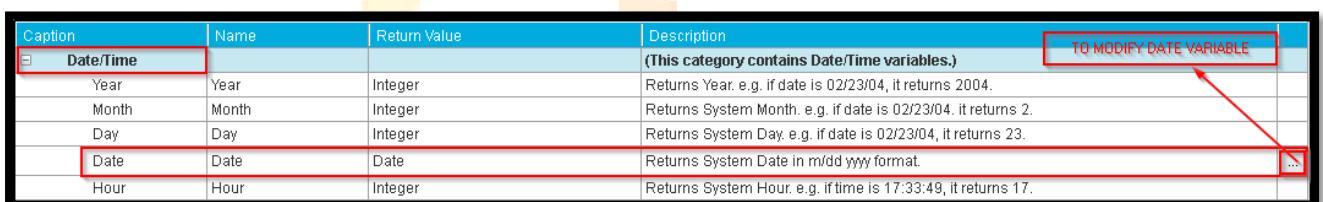
```

- Now, when we run the script, we get to see only those messages that have “Test” somewhere in their subject line as below



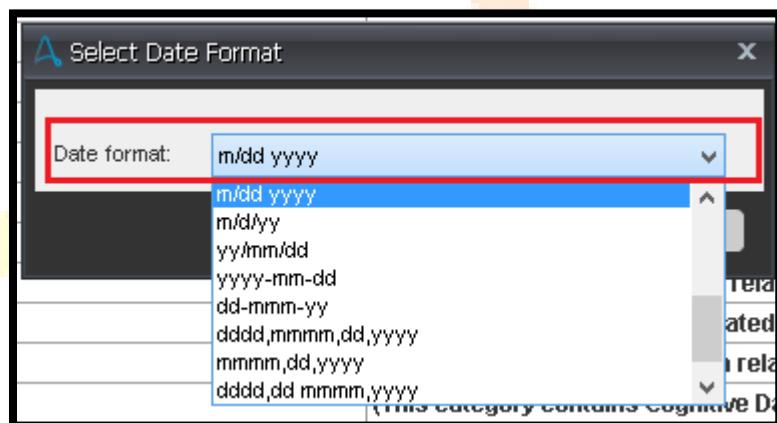
- Our current script has the capability to look into our mail box, pick up mails that have “Test” anywhere in their subject line and then pop them up on the screen. However, the catch is that it will go through all the mails in the server every time it is run, which could be very time consuming, so we need to restrict it to search for only amongst those mails which came in the current date.

- For this we will first have to set up our Date variable in such a way that it is restricted to only the day, month, and year. Let us understand why
- Let's say the mail was received on 15<sup>th</sup> Aug 2016 11:30:15
- Now if we run our script looking for mails in today's date runs even 1 second later then it's time would be 15<sup>th</sup> Aug 2016 11:30:16 which will not match up with 15<sup>th</sup> Aug 2016 11:30:15. Similarly matching on minute's level may also be an issue.
- So, if we are looking for mails within current day then we click on “**VARIABLE MANAGER**” , then we choose “**Show System Variables**” which takes us into the following screen as below

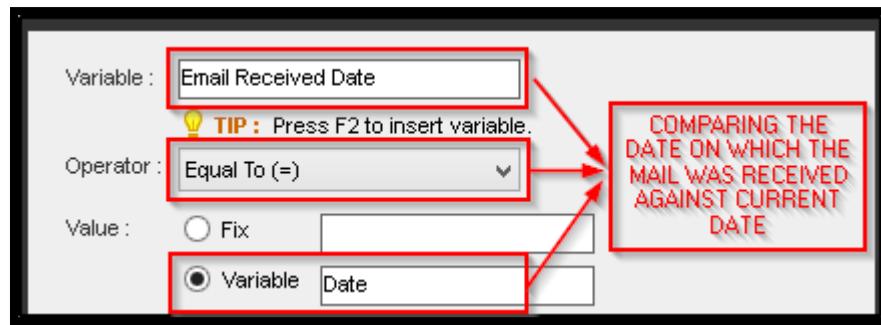


Caption	Name	Return Value	Description	
Date/Time			(This category contains Date/Time variables.)	<b>TO MODIFY DATE VARIABLE</b>
Year	Year	Integer	Returns Year. e.g. if date is 02/23/04, it returns 2004.	
Month	Month	Integer	Returns System Month. e.g. if date is 02/23/04, it returns 2.	
Day	Day	Integer	Returns System Day. e.g. if date is 02/23/04, it returns 23.	
<b>Date</b>	<b>Date</b>	<b>Date</b>	Returns System Date in m/dd yyyy format.	
Hour	Hour	Integer	Returns System Hour. e.g. if time is 17:33:49, it returns 17.	

- Choose any date format as per preference, which does not include Hours, minutes and seconds, to prevent any conflicting issues over there as below:



- Now we want to keep our previously made IF Construct (Checking for “Test” in subject line) within this new IF Construct (where we will be checking if mail came in within current date) which we will now make.
- We drag a “If Else – Variable” command and do the following as below:



- Our Script looks like as below

```

1  ↗ Start Loop "Each message on server: smtp.gmail.com, User Name: aasbtraining@gmail.com, SSL, ServerType: IMAP, Message Format: Plain Text"
2  ↗ Comment: Please enter your commands to loop. Use Email Automation variables for each email on Server.(e.g. $Email From$, $Email To$, etc.)
3  ↗ IF
4  ↗ Comment: Please enter the conditional commands here.
5  ↗ IF
6  ↗ Comment: Please enter the conditional commands here.
7  ↗ Message Box: "EMAIL FROM" : $Email From$ _____ EMAIL SUBJECT
8  ↗ End If
9  ↗ End If
10 ↗ End Loop

```

- Now, we need to exit prematurely from the loop if we encounter a mail that is in yesterday's date or the day before it or any date that is not the current date.
- For that purpose, we drag an "If/Else – Else" and position it after Line 8 in the screen shot above, immediately after this "Else" we will position an "Loop – Exit Loop" command, so that our script looks like as below

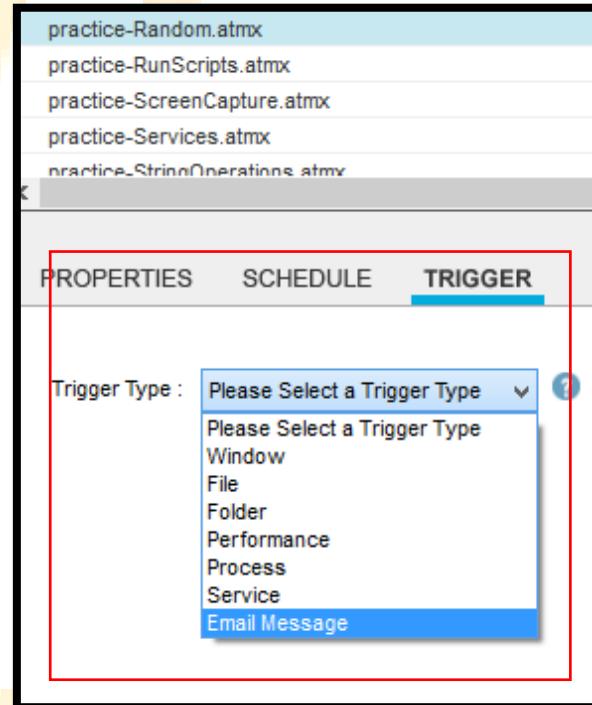
```

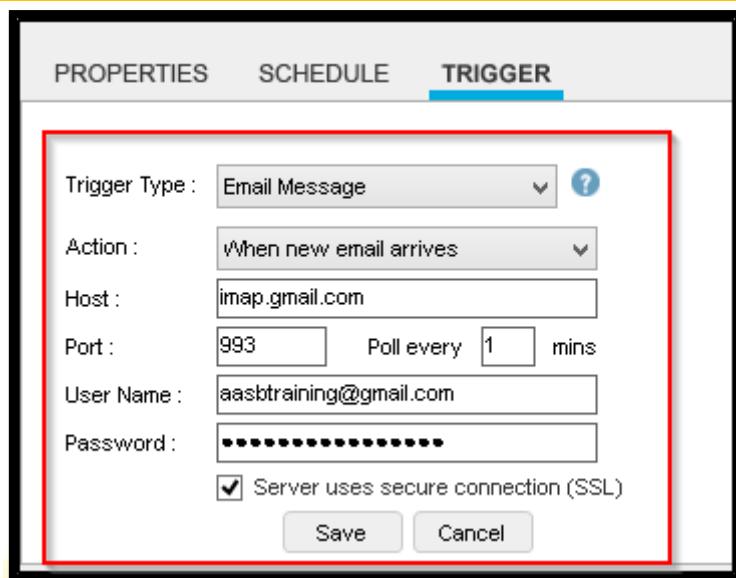
1  ↗ Start Loop "Each message on server: smtp.gmail.com, User Name: aasbtraining@gmail.com, SSL, ServerType: IMAP, Message Format: Plain Text"
2  ↗ Comment: Please enter your commands to loop. Use Email Automation variables for each email on Server.(e.g. $Email From$, $Email To$, etc.)
3  ↗ IF
4  ↗ Comment: Please enter the conditional commands here.
5  ↗ IF
6  ↗ Comment: Please enter the conditional commands here.
7  ↗ Message Box: "EMAIL FROM" : $Email From$ _____ EMAIL SUBJECT
8  ↗ End If
9  ↗ ELSE
10 ↗ Else
11  ↗ Exit Loop
12  ↗ End If
13  ↗ End Loop

```

- Now whenever our script is executed, it does the following
  - Keeps reading mails in a loop from our mailbox one after other

- a. Checks if mails came in current date
  - i. If Yes – Checks if subject line includes “Test”
    - 1. If Yes – Displays mail content
    - 2. If No – Does not display anything
  - ii. If No – Exits Loop
- To make this Task fire, every time a mail comes inside this mailbox, we must create an Email Trigger on the same inbox and point to the current task.





- Now , whenever a mail comes into the specified mail box , the generic email trigger is fired , which then causes our Task to be fired , which has the capability to detect mails in current date with specific subject line and then take some action on top of it.

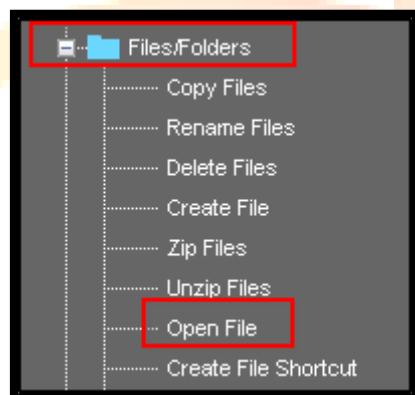
## 11) Error Handling

### Objective

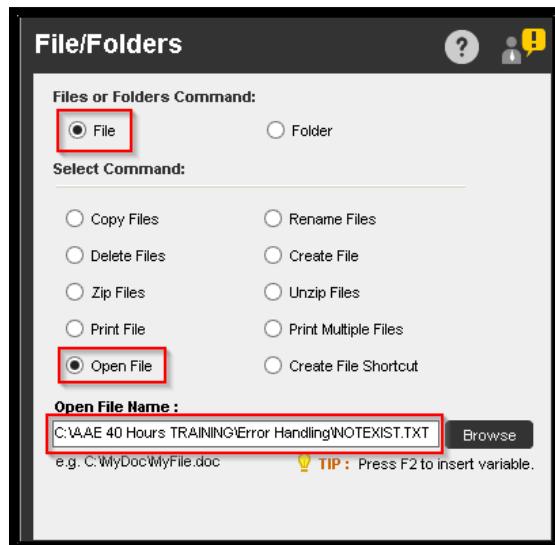
- Step by step demonstration of Error Handling and its various facets, like:
  - a) Setting up a Begin – End Error Handling Section, with its various sub options.
  - b) Setting up a Troubleshooting block, where we try to fix the error.
  - c) Understanding the Error View.
  - d) Setting up and working in the Debugging Mode
  - e) Setting Breakpoints.
  - f) Working with Variable Watch Table.
  - g) Stepping over Breakpoints, when running the script.

### Example: Error Handling

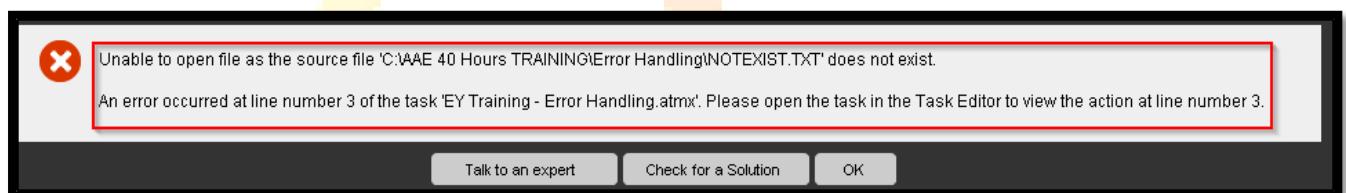
- As usual we will first proceed to open the Automation Anywhere Enterprise client after ascertaining that the Automation Anywhere Control Room Service is up and running.
- We open the Task Editor to create a new task for this activity.
- We next drag a “Files/Folders -> Open File” command, and try to open a file which is not existent, which will throw an error once we execute it, snapshot below:



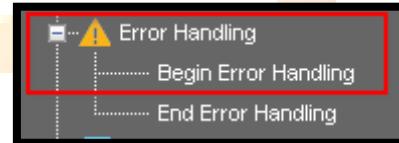
- The open file command dialog looks like as below:



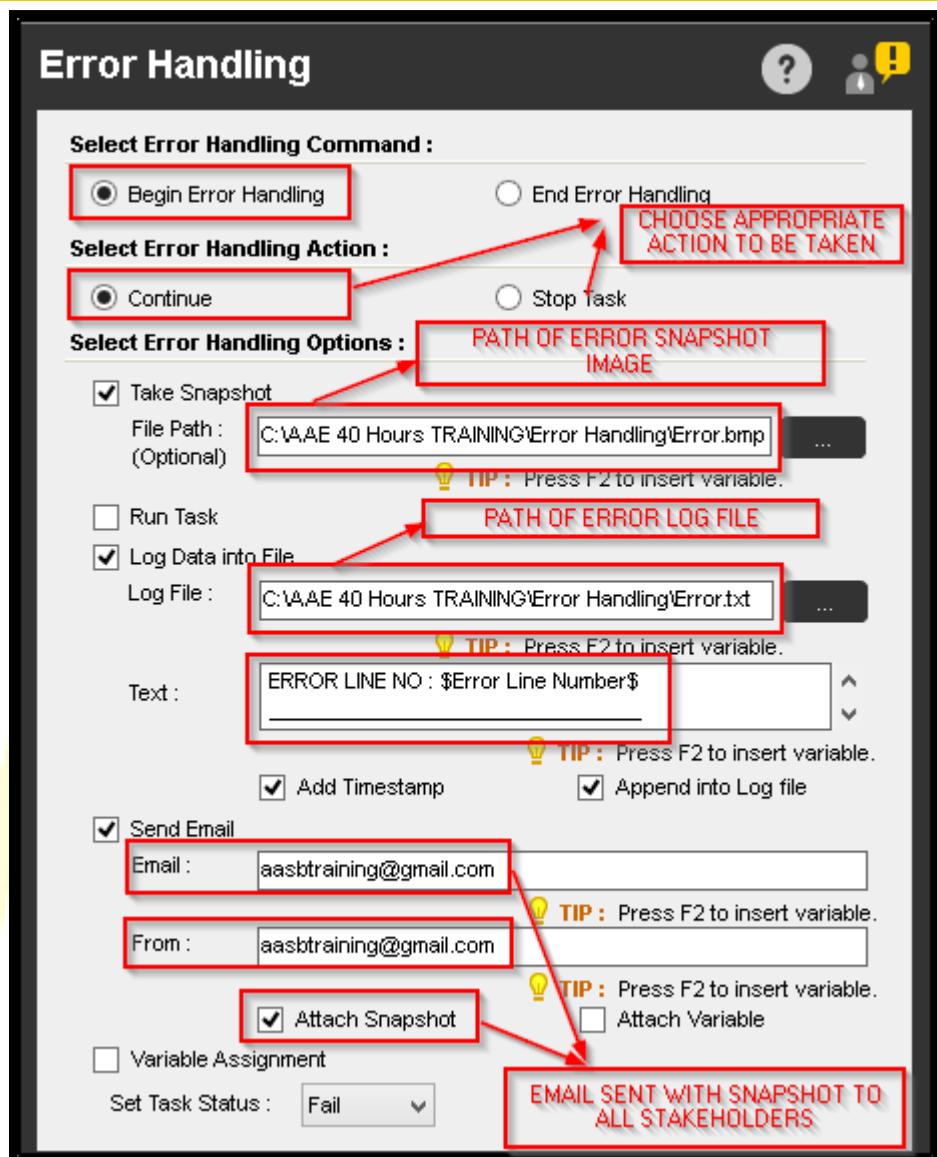
- Next we run the command above, which will lead to the following error snapshot coming up on our screen as below



- Next we drag a “Error Handling – Begin Error Handling” command to the right as below



- This opens the following dialog on our screen as below

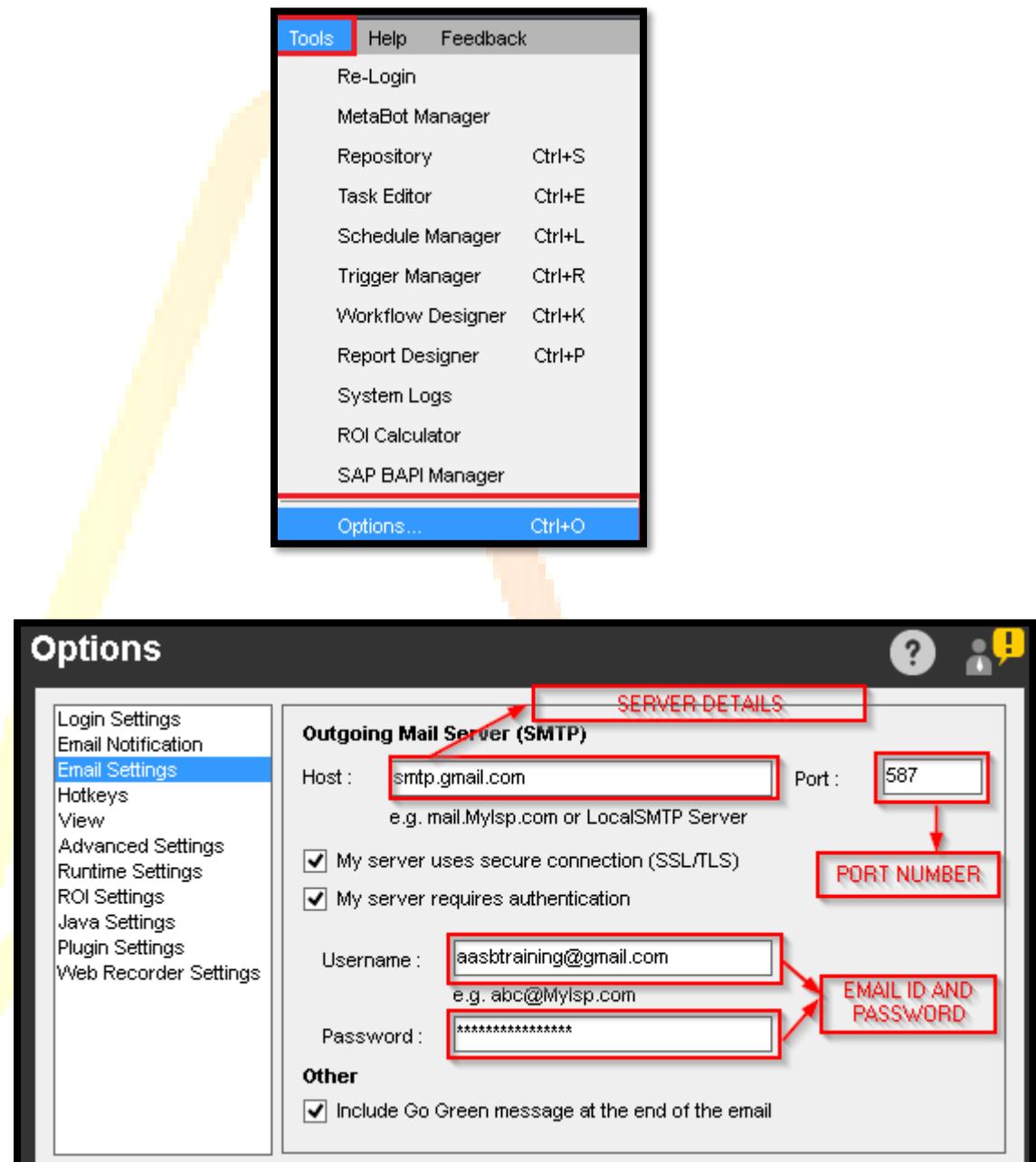


- Select Error Handling Action:
  - Continue: Throws Control to the next line immediately after the End Error Handling statement, ignoring remaining lines between the line that threw the error and other lines that may be following it inside the Begin – End Error handling block.
  - Stop Task: Stops the task at that point itself, i.e. at the line where the error came.
- Select Error Handling Options:
  - Log Data into File:

Text: ERROR LINE NO: \$Error Line Number\$

ERROR DESCRIPTION: \$Error Description\$

- b) Kindly note that for the Email Notification Feature to work, we should have specified the Outgoing email server details, under the main screen of our Software inside “Tools -> Options -> Email Settings” as seen in the snapshot below:



- Our Script should look like as below:

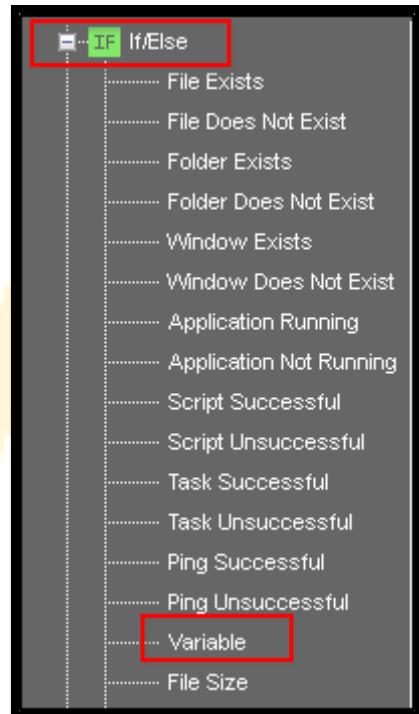
1		Begin Error Handling; Action: Continue; Options: Take Snapshot, Log to File, Send Email, Task Status: Fail
2		Comment: Please enter the commands to handle errors.
3		Open File "C:\VAE 40 Hours TRAINING\Error Handling\NOTEXIST.TXT"
4		End Error Handling

- When we run this script, post saving it, we will again get to see the error pop up, however since this time the error causing line is embedded between the Begin and End error handling statements, we do not need to close the error pop up by ourselves, it is automatically closed, also post running we can see that the error snapshot and error log file have been created successfully at our pre-specified location as below:

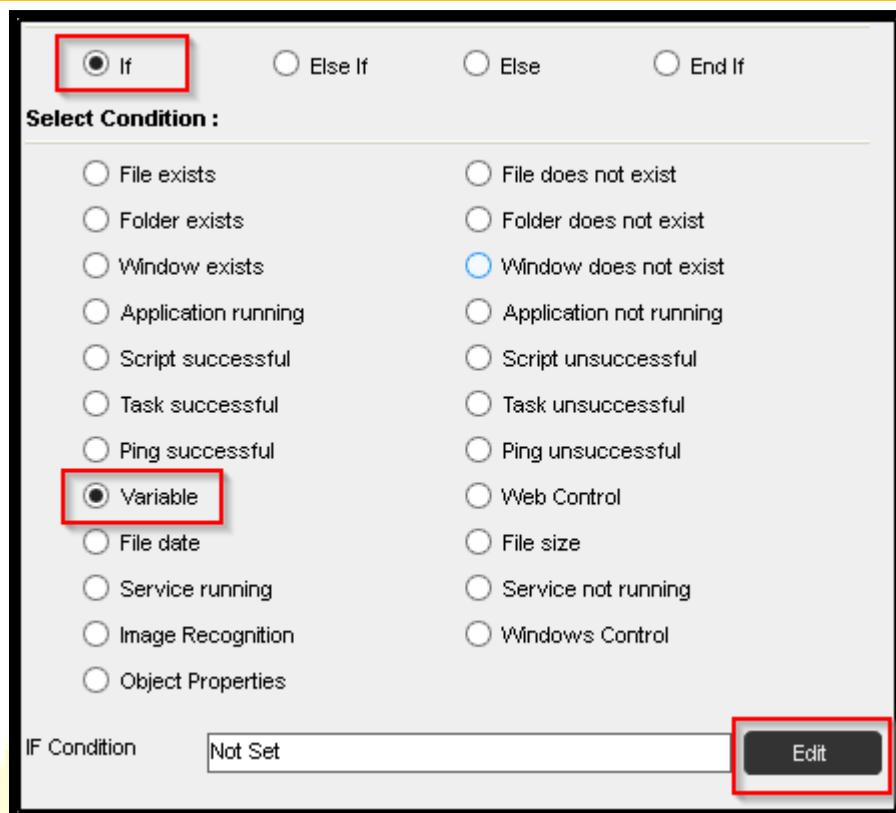
AE 40 Hours TRAINING > Error Handling			
Name	Date modified	Type	Size
Error.bmp	7/21/2016 6:29 PM	Bitmap image	147 KB
Error.txt	7/21/2016 6:29 PM	Text Document	1 KB

- Also, if we open our mail box, we would have received an email notification from the software that an error came, with the snapshot attached.
- Now that we could get the required response from within the Begin and End Error Handling block, let us try to create the Troubleshooting block where we will try to get some basic Error Handling done.
- For this we will make use of our inbuilt Error Handling Variables which we are also capturing inside of our log files.
  - Error Line Number Variable: Gives us the line number on which an error came in the script. In case no error comes in our script this variable contains 0.
  - Error Description Variable: Gives us the detailed description of the error that came in. In case no error comes in our script this variable contains nothing.

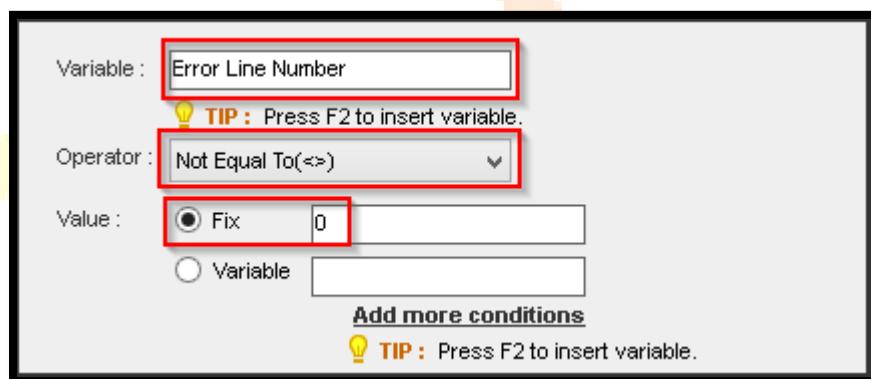
- To check if our Error Line Number variable does point to some line number we will make use of the “If/Else – Variable” command as seen here.



- This shows the following dialog, click on Edit

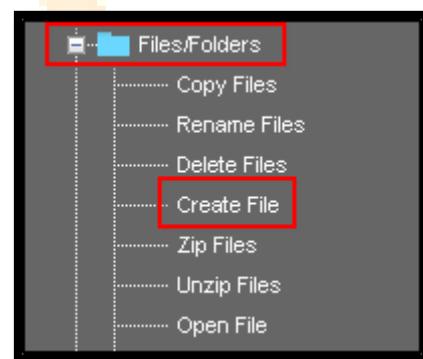


- In the dialog below, we specify what value we are trying to check as below i.e. whether the Error Line Number variable is zero or not.

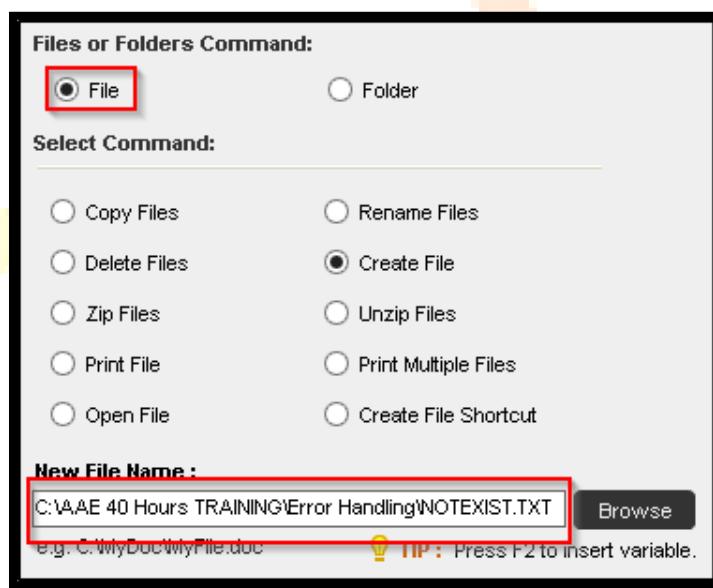


- Within this “If-End if” block we can plan for, what troubleshooting capabilities do we want to provide the software. That way the script can tackle those errors that we can foresee, beyond that manual intervention may be required.

- In our scenario, the file is missing, which is also what our error snapshot capture and log file tell us. There are two possibilities here:
  - a) The file is empty; in which case, we need to create a new file.
  - b) The file had some data in it, in that case we can copy the file from its backup location to the location from which it is missing
- We will presume that it is the first case, and will create a new file here using “files/folders –Create File” command as below



- This opens the following dialog where in we set the name of the new file that has to be created as below



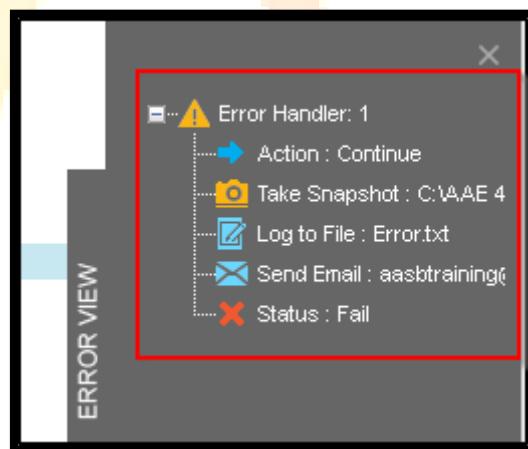
- Our Script looks like as below

```

1  ⚠ Begin Error Handling; Action: Continue; Options: Take Snapshot, Log to File, Send Email, Task Status: Fail
2  📄 Comment: Please enter the commands to handle errors.
3  📁 Open File "C:\VAAE 40 Hours TRAINING\Error Handling\NOTEXIST.TXT"
4  ⚠ End Error Handling
5  ⚡ If $Error Line Number$ Not Equal To (<>) "0" Then
6  📄 Comment: Please enter the conditional commands here.
7  📁 Create File "C:\VAAE 40 Hours TRAINING\Error Handling\NOTEXIST.TXT"
8  ✗ End If

```

- Next, we click on “Error View”, the following window opens, which captures all the error handling actions in a nutshell as can be seen below



- Next, we click on “Enable Debugging”, this will bring up some more buttons as seen here



- Also, it brings up the following variable watch table, where in we can monitor our key variables as below



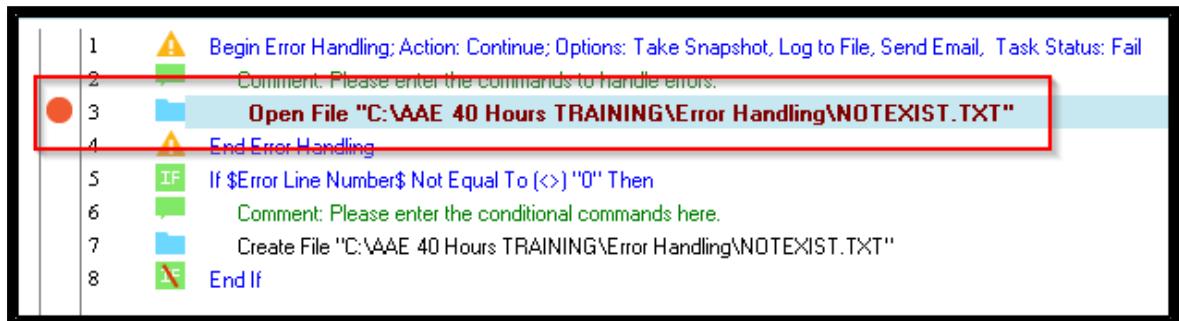
- We click on “Add” button above to add our variables as seen below

<input type="checkbox"/> Table Column	Loop	Use in Loop for each row in a Table. Returns single row of Table.
<input type="checkbox"/> Excel Column	Loop	Use in Loop for each row in Excel. Returns single row of excel.
<input checked="" type="checkbox"/> Error Line Number	Loop	Returns Automation Anywhere task error line number.
<input checked="" type="checkbox"/> Error Description	Loop	Returns Automation Anywhere task error line description.
<input type="checkbox"/> Filedata Column	Loop	Use in Loop for each record in CSV/Text file. Returns single node of File.
<input type="checkbox"/> XML Data Node	Loop	Use in Loop for each node in XML file. Returns single node of File.
<input type="checkbox"/> Event Cell Column	Event	Returns the column of the Event on which the operation has been performed.

- Our Variable watch window looks as below



- Now we will add breakpoints , by clicking on “ Toggle BreakPoint” command at the top after clicking on line 3.
- This will add a breakpoint as seen below



The screenshot shows a script editor window with the following code:

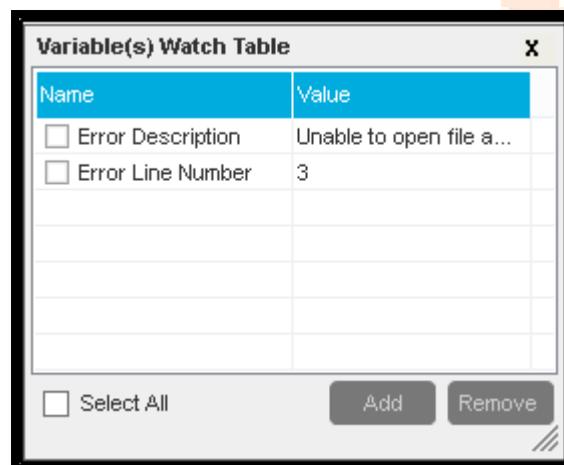
```

1   Begin Error Handling; Action: Continue; Options: Take Snapshot, Log to File, Send Email, Task Status: Fail
2   Comment: Please enter the commands to handle errors.
3   Open File "C:\VAE 40 Hours TRAINING\Error Handling\NOTEXIST.TXT"
4   End Error Handling
5   If $Error Line Number$ Not Equal To (<>) "0" Then
6     Comment: Please enter the conditional commands here.
7     Create File "C:\VAE 40 Hours TRAINING\Error Handling\NOTEXIST.TXT"
8   End If

```

A red box highlights the line `Open File "C:\VAE 40 Hours TRAINING\Error Handling\NOTEXIST.TXT"`, which contains a red circle indicating a breakpoint.

- Now let's run our script, the following will happen
  - The script will start running in debugging mode.
  - A yellow-orange line starts running from top to bottom telling us which line is currently getting executed.
  - When it comes to the line with the Breakpoint, it will halt.
  - To make it go ahead press “ Step Over (F10)” and it will go ahead.
  - The error popup comes and control will then jump to line 5
  - At this point our Variable Watch Table gets populated as below



Name	Value
Error Description	Unable to open file a...
Error Line Number	3

Buttons at the bottom:  Select All, Add, Remove.

g) Again we press “ Step Over (F10)” to go across line 5

h) It comes to line no 7 , where we again press “ Step Over (F10)” so that it executes and creates the missing file.

i) Subsequently it comes to line no 8 , where we again have to press

“ Step Over (F10)” to make it run.

- This concludes our error handling , we see that the missing file got created in the specified directory as below



- Now we clean up what we did under Error Handling as below

• We choose “ Clear All Breakpoints”

• Next we choose “ Disable Debugging”

• This brings us back to our original screen with the error resolved .

## 12) File Transfer Protocol

### Objective

- Step by step demonstration of FTP and its various facets, like:
  - a) Connecting/Disconnecting to a FTP site.
  - b) Uploading/Downloading, Files/Folders.
  - c) Renaming files
  - d) Creating Folders on FTP site
  - e) Setting up a new folder as default location for uploading/downloading from FTP.

### Example: File Transfer Protocol

- As usual we will first proceed to open the Automation Anywhere Enterprise client after ascertaining that the Automation Anywhere Control Room Service is up and running.
- We open the Task Editor to create a new task for this activity.
- We will use our FTP server configured on IIS for this demonstration as below

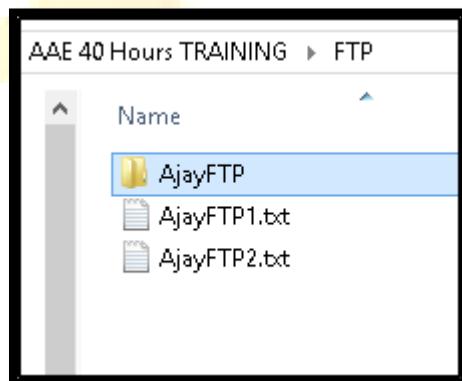
File Edit View Favorites Tools Help

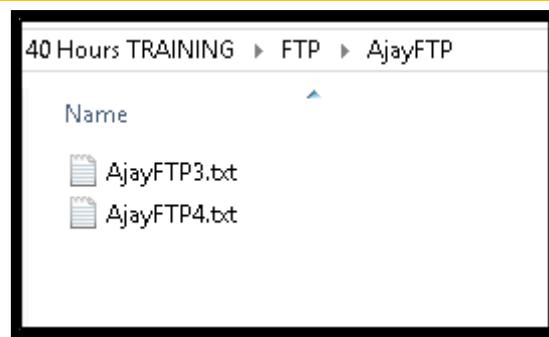
## FTP root at 127.0.0.1

To view this FTP site in File Explorer: press Alt, click View, and then click Open FTP Site in File Explorer.

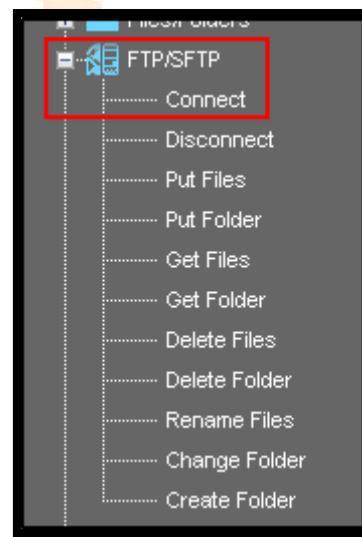
07/24/2015 09:15PM	347	<a href="#">abc.xml</a>
04/29/2016 07:53PM	Directory	<a href="#">ART</a>
05/26/2015 08:49PM	Directory	<a href="#">aspnet_client</a>
12/09/2015 11:27PM	Directory	<a href="#">AT&amp;TDecWeek</a>
04/14/2016 10:38PM	Directory	<a href="#">AT&amp;TWF11thApr</a>
12/18/2015 12:13AM	Directory	<a href="#">AT&amp;TWF14thDec</a>
03/18/2016 10:33PM	Directory	<a href="#">AT&amp;TWF15thMar</a>
03/23/2016 11:48PM	Directory	<a href="#">AT&amp;TWF21stMar</a>
03/31/2016 10:45PM	Directory	<a href="#">AT&amp;TWF28thMar</a>
04/09/2016 12:01AM	Directory	<a href="#">AT&amp;TWF4thApr</a>
06/07/2016 09:10PM	Directory	<a href="#">BFR</a>
05/21/2016 02:58AM	Directory	<a href="#">CHFTP</a>
06/28/2016 09:19PM	Directory	<a href="#">Comcast</a>
07/21/2016 01:27AM	Directory	<a href="#">EyTraining</a>
01/21/2016 11:05PM	Directory	<a href="#">HCA-HealthCare</a>
05/12/2015 09:56PM	0	<a href="#">IroncadDPAT&amp;TCalMFftp1.txt</a>
05/12/2015 09:56PM	0	<a href="#">IroncadDPAT&amp;TCalMFftp2.txt</a>
03/12/2016 01:10AM	Directory	<a href="#">Microsoft</a>
11/13/2015 11:40PM	0	<a href="#">SheetzFTP1.txt</a>
11/13/2015 11:40PM	0	<a href="#">SheetzFTP2.txt</a>
03/01/2016 11:19PM	Directory	<a href="#">TopTech29thFeb</a>
09/30/2015 01:08AM	Directory	<a href="#">VerosCredit</a>
09/11/2015 11:14PM	Directory	<a href="#">Whirlpool</a>
02/19/2016 11:07PM	Directory	<a href="#">whirlpool2</a>
11/07/2015 04:48AM	Directory	<a href="#">YuMe</a>

- Also, we have a local folder on the hard disk, which we will be using as the source for picking up files/folders as below





- Initially we need to connect to our FTP server, for which we choose ‘FTP -> Connect’ command as below



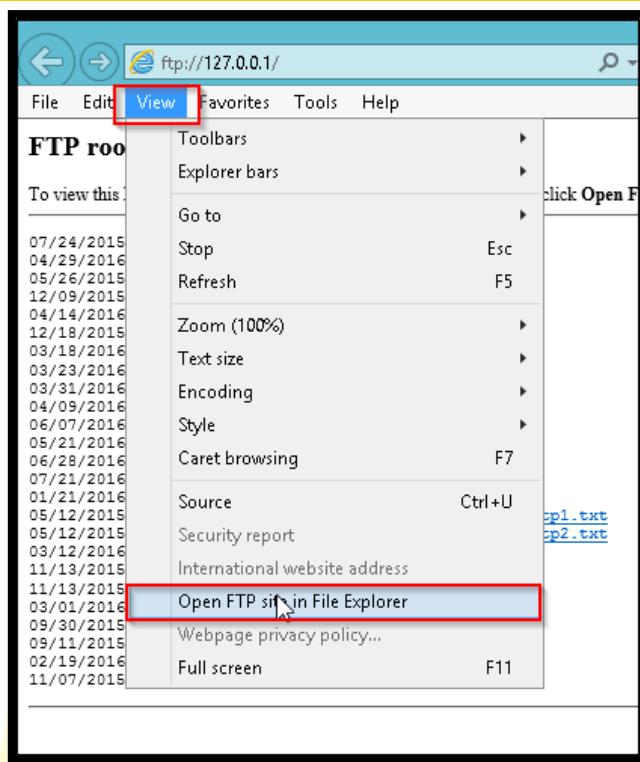
- This opens the following dialog as seen here , where we specify details like our FTP server name , credentials to connect to the FTP site , and other properties as seen below :

**FTP/SFTP**

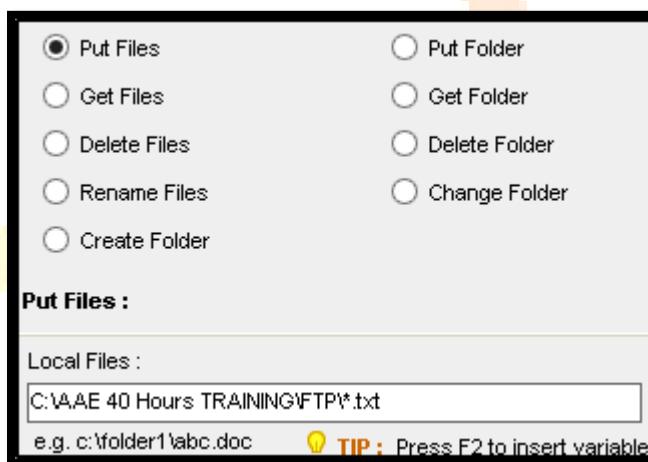
[?](#)


<b>Select FTP Command :</b>	
<input checked="" type="radio"/> Connect	<input type="radio"/> Disconnect
<input type="radio"/> Put Files	<input type="radio"/> Put Folder
<input type="radio"/> Get Files	<input type="radio"/> Get Folder
<input type="radio"/> Delete Files	<input type="radio"/> Delete Folder
<input type="radio"/> Rename Files	<input type="radio"/> Change Folder
<input type="radio"/> Create Folder	
<b>Connection Details :</b>	
FTP Server :	<input type="text" value="127.0.0.1"/>
User Name :	<input type="text" value="automation"/>
>Password :	<input type="password" value="*****"/>
<span style="color: orange;">💡 TIP : Press F2 to insert variable.</span>	
<input type="checkbox"/> Anonymous Login <input type="checkbox"/> Secure FTP	
<input type="button" value="Test Connection"/>	
<input type="button" value="Normal View"/>	

- Next, we want to open our FTP site inside a Windows Explorer window, for that we choose “View -> Open FTP Site in File Explorer” as seen here



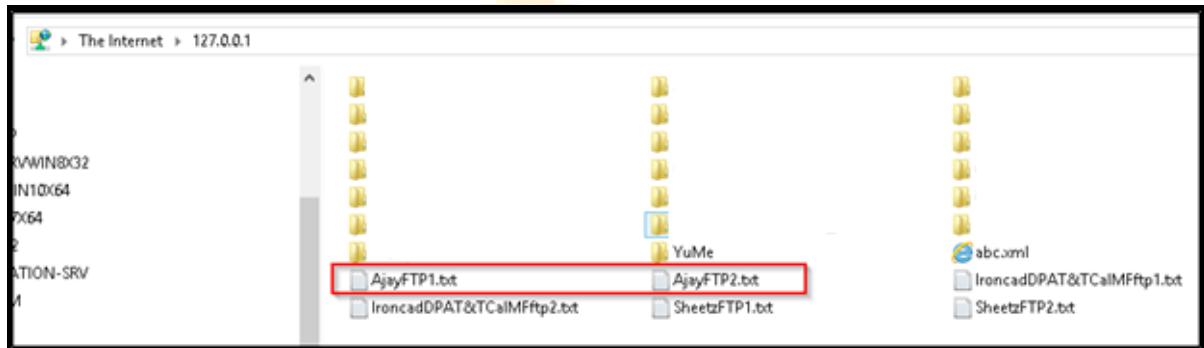
- Now, first, we want to upload all “txt” files lying in our folder to the FTP server, for that we will use the “FTP -> put files” command as seen here



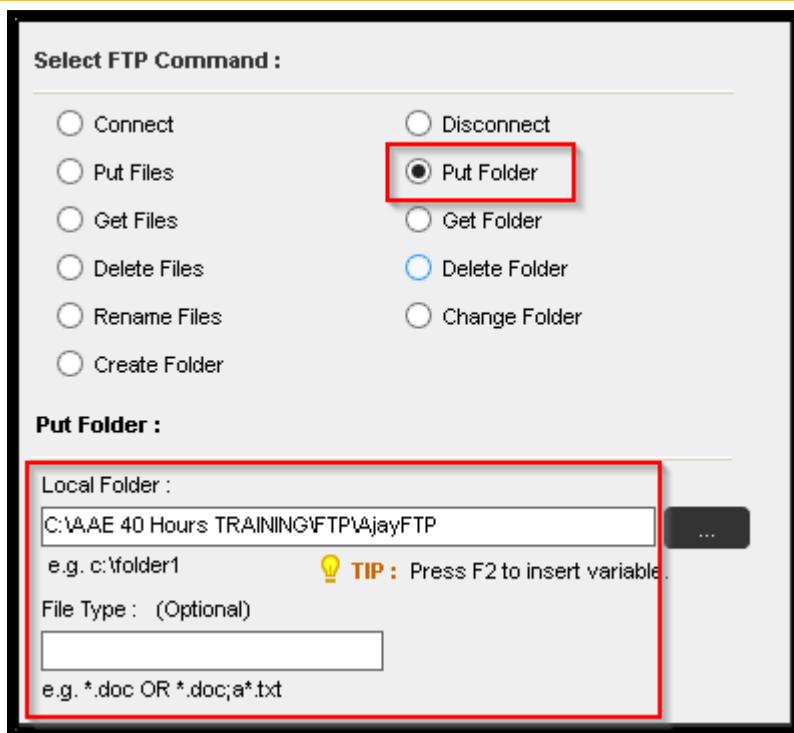
- We then drag a “FTP -> disconnect” command to the right, which will terminate our FTP connection as below



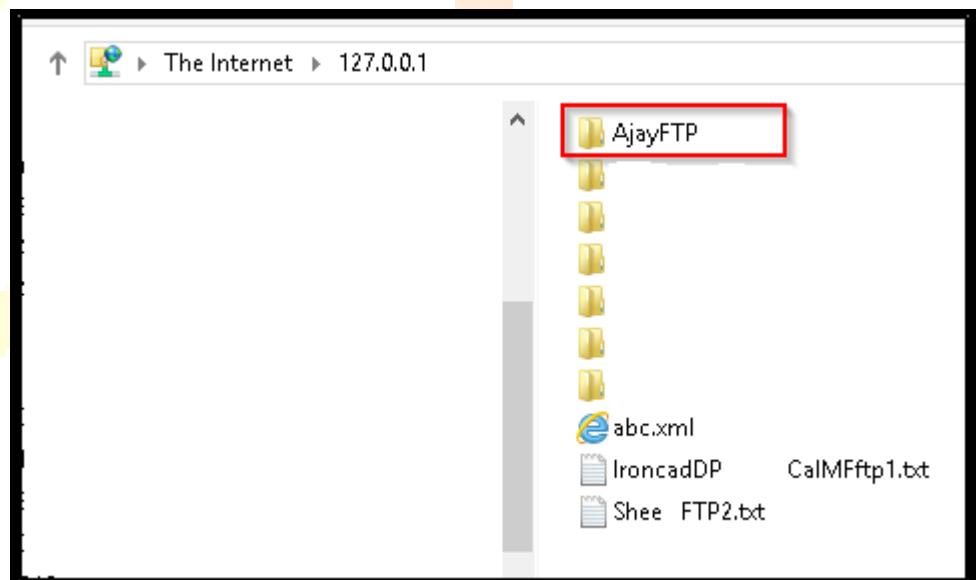
- We save the entire script and then run it. We see that our files have been uploaded to the FTP server as seen below



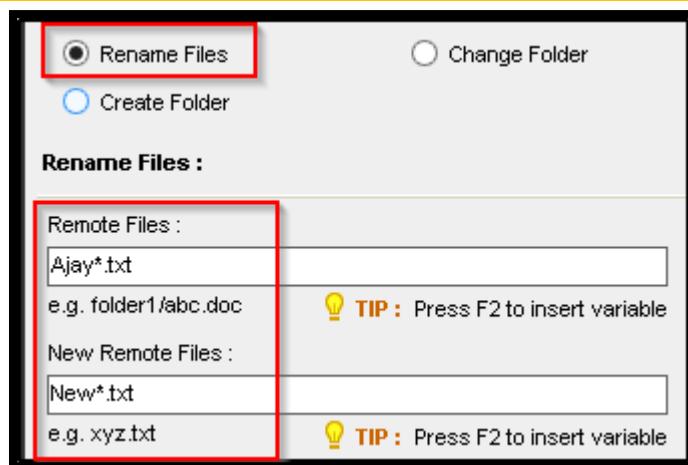
- We disable the “put files command” and drag a “FTP -> Put Folder” command to the right . This command allows us to put an entire folder from our local machine on to the FTP site.
- We want to take the folder “AjayFTP” and all it’s content files and upload it to the server , as seen in snapshot below
- Our “put folder” command looks like as below



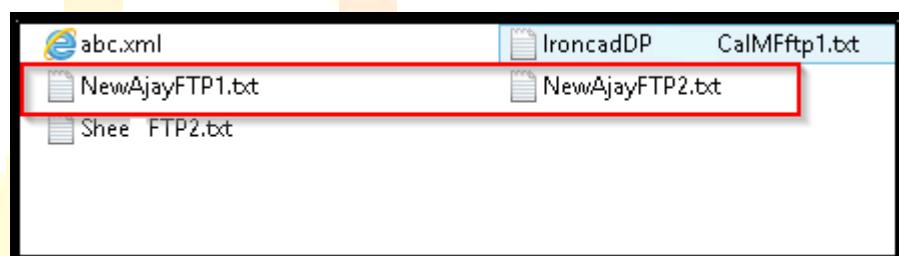
- We save and run the script and thus we see that our folder has been uploaded on the FTP site along with all its contents, as below:



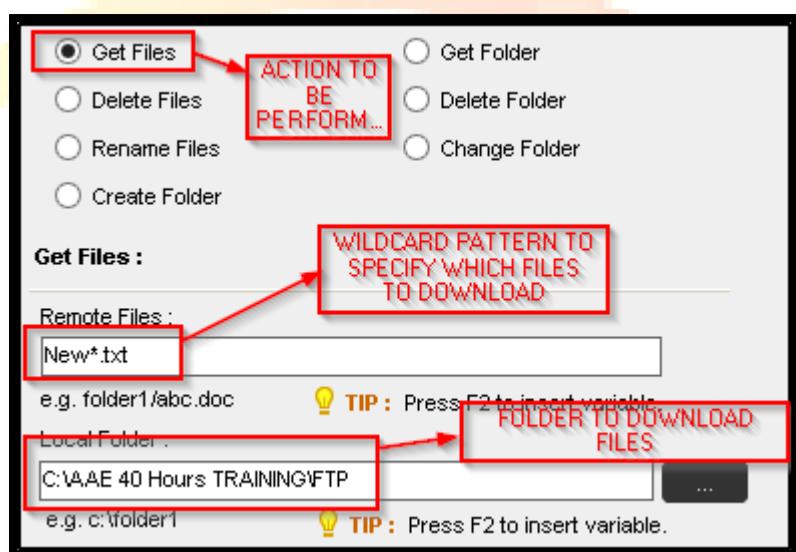
- Disable the “Put Folder” command also.
- Presume that we are required to rename the files lying on the FTP server, for this we will bring in the “FTP -> Rename files” command. The command looks like as below:



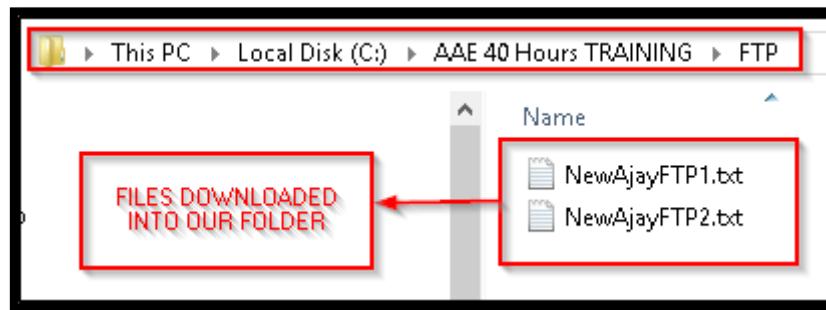
- We can see that our files have been renamed successfully as seen in snapshot below:



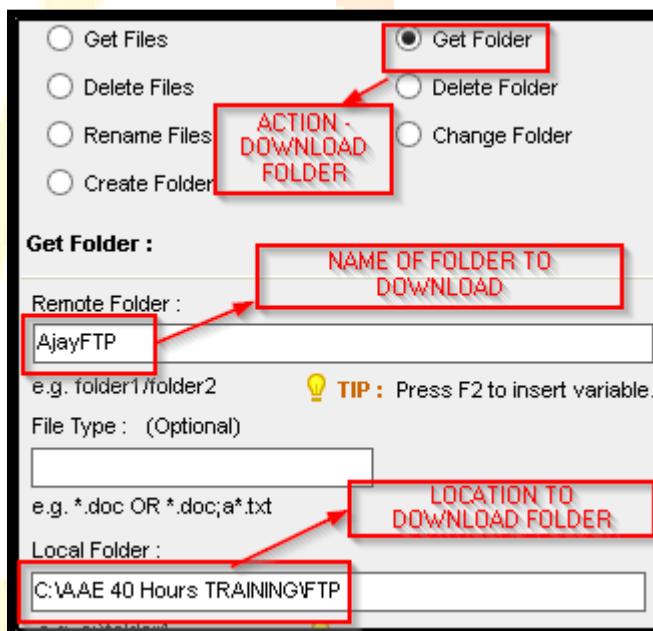
- Disable “Rename Files” command
- Next, we delete all files from our local folder, so that we can download our uploaded
- We want to download the files that we had uploaded earlier, for that we will use the “FTP – Get Files” command as below



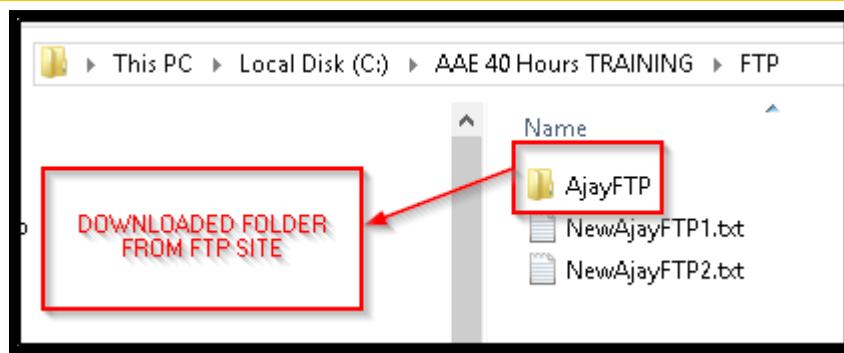
- Save and Run script , we can see our files have been downloaded as below



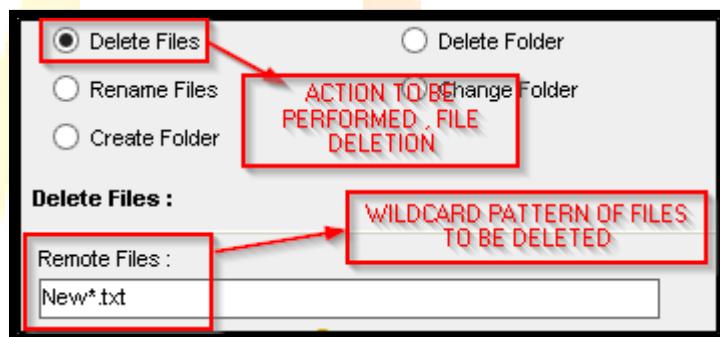
- Disable the “Get Files command” we just ran above.
- Next we want to bring down the folder that we had uploaded earlier into our FTP site, for that we will bring in the “FTP – Get Folder” command as below



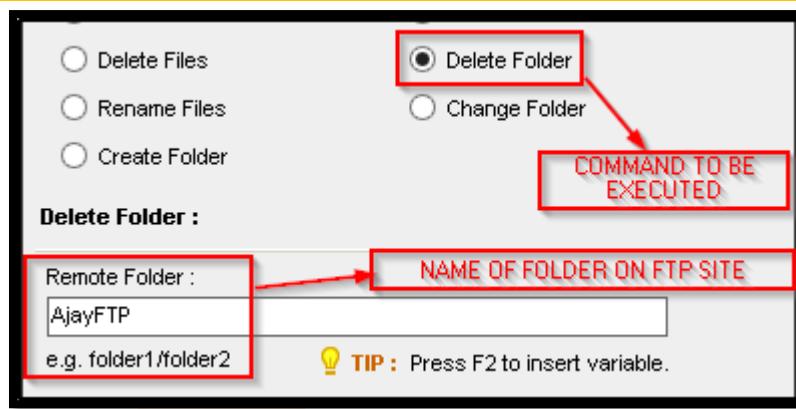
- Save And Run, we should see our folder getting downloaded from the FTP site into our Local Folder on our machine as below:



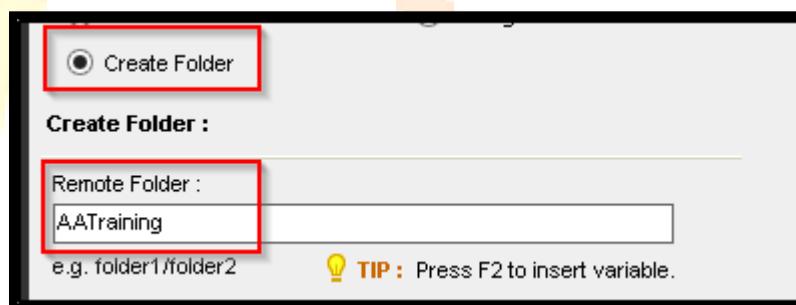
- We then disable the “Get Folder” command we ran earlier, and next drag a “FTP – Delete Files” command over, so that we can delete those files lying on the FTP site which are no longer required to be shared, as below



- Save And Run the script. We can see that our specified files have been deleted.
- We disable the “Delete Files” command, and bring over a “FTP – Delete Folder” command.
- Next, we want to delete the folder “AjayFTP” which we had uploaded on the FTP site, for that we will use the “FTP – Delete Folder” command as below



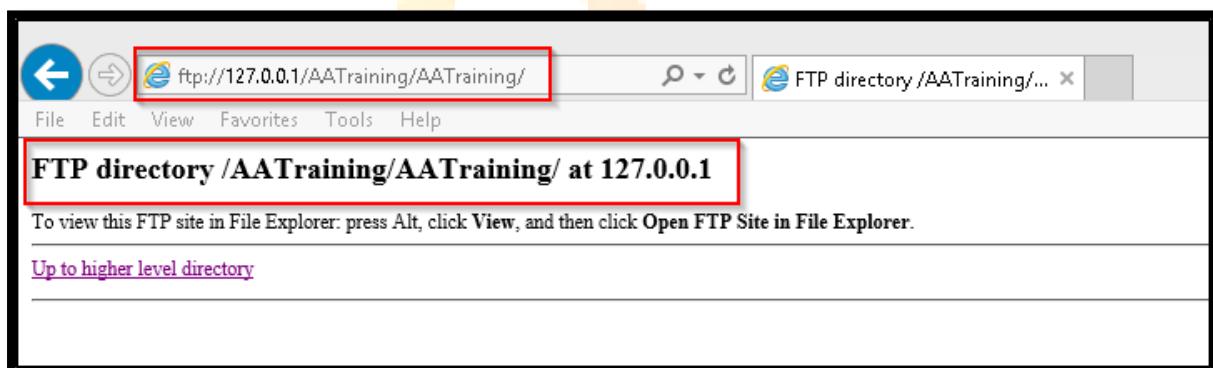
- We “Save” and “Run” our script, which leads to the folder being deleted from the FTP site.
- We disable the “FTP – Delete folder” command that we just made above.
- Next we want to create our own hierarchy of folders on the FTP site, for which we will use the “FTP – Create Folder” command.



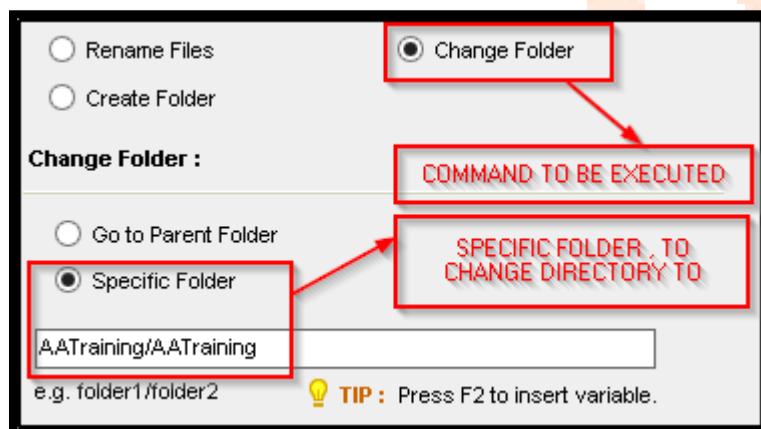
- The above command will go and create a folder called “AATraining” in the root directory of the FTP site.
- Within the Folder “AATraining” we want another folder “AATraining” should be created, for that we will bring in another “FTP – Create Folder” command within the previous command that we have put in, as below:



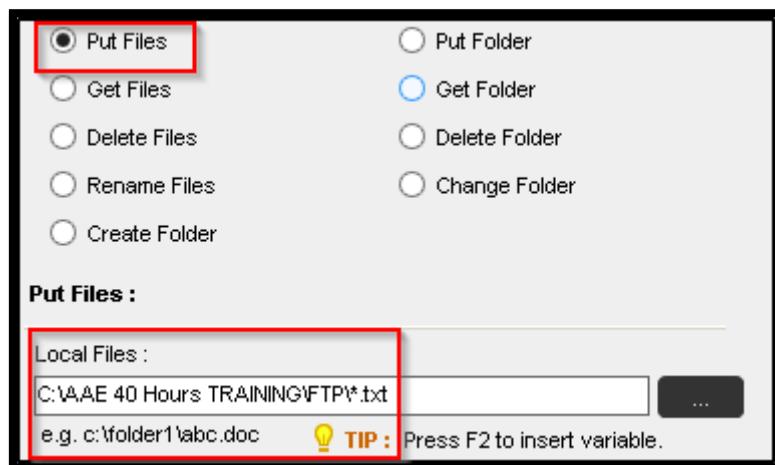
- We “Save” and “Run” the script, which gives us the desired result as below:



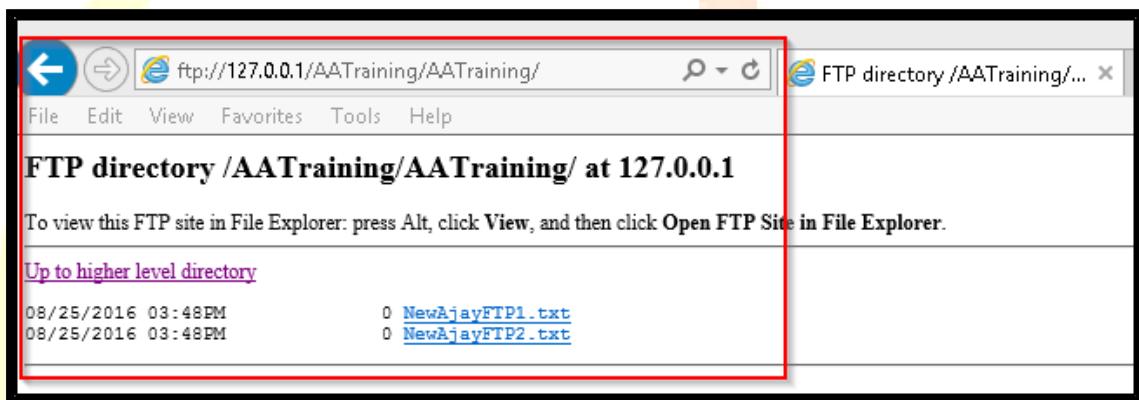
- We can see that in our FTP server , we have been able to create a directory structure, which has an AATraining folder within another AATraining folder .
- We disable the above two “FTP – Create Folder” commands.
- Next we drag in an “FTP – Change Folder” command over, and here we want to change our current working directory to the new directory structure that we have created above, i.e. AATraining/AATraining, as below:



- To see the effect of this command, we will re-run the “FTP – Put Files” command, which will upload our files on the FTP server. We will specify the files to be uploaded as below:

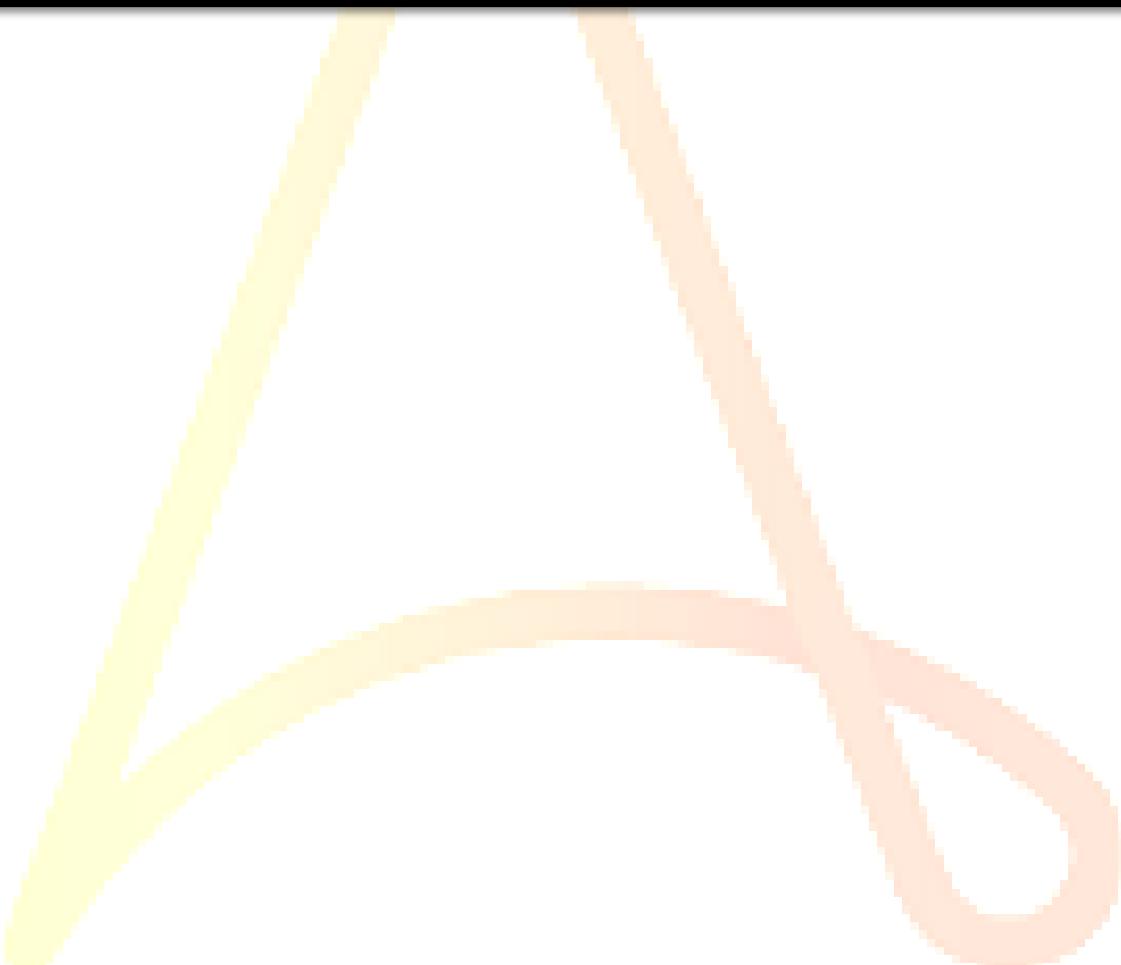


- We Save And Run the script, and as below we see that this time our files have been uploaded to the location “AATraining/AATraining” as can be seen below:



- Our entire script looks like as below :

FILTERS		<input checked="" type="checkbox"/> Mouse Moves	<input checked="" type="checkbox"/> Keystrokes	<input checked="" type="checkbox"/> Mouse Clicks	<input checked="" type="checkbox"/> Delays	<input checked="" type="checkbox"/> Other	Windows	All
1								FTP/SFTP : Connect to "127.0.0.1"
2								FTP/SFTP: Put All Files in Folder "C:\AAE 40 Hours TRAINING\FTP" with criteria "C:\AAE 40 Hours TRAINING\FTP\*.txt"
3								FTP/SFTP: Put Folder "C:\AAE 40 Hours TRAINING\FTP\AjayFTP"
4								FTP/SFTP: Rename Remote File "Ajay.txt" to "New.txt"
5								FTP/SFTP: Get Files "New.txt"
6								FTP/SFTP: Get Folder "AjayFTP"
7								FTP/SFTP: Delete Remote File "New.txt"
8								FTP/SFTP: Delete Folder "AjayFTP"
9								FTP/SFTP: Create Folder "AATraining"
10								FTP/SFTP: Create Folder "AATraining/AATraining"
11								FTP/SFTP: Change Folder "AATraining/AATraining"
12								FTP/SFTP: Put All Files in Folder "C:\AAE 40 Hours TRAINING\FTP" with criteria "C:\AAE 40 Hours TRAINING\FTP\*.txt"
13								FTP/SFTP: Disconnect



## 13) XML

### Objective

- Step by step demonstration of XML commands like
  - a) Connecting/Disconnecting to an XML File
  - b) Validating an XML document.
  - c) Inserting/Updating/Deleting Nodes in an XML document.
  - d) Fetching the value of a Single XML node or a group of nodes.
  - e) Execute XPath Function.

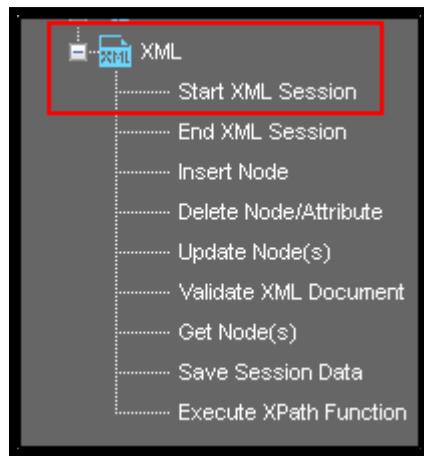
### Example: XML Document

- As usual we will first of all proceed to open the Automation Anywhere Enterprise client after ascertaining that the Automation Anywhere Control Room Service is up and running.
- We open the Task Editor to create a new task for this activity.
- We will work with the following XML document first, as seen in the Internet explorer below:

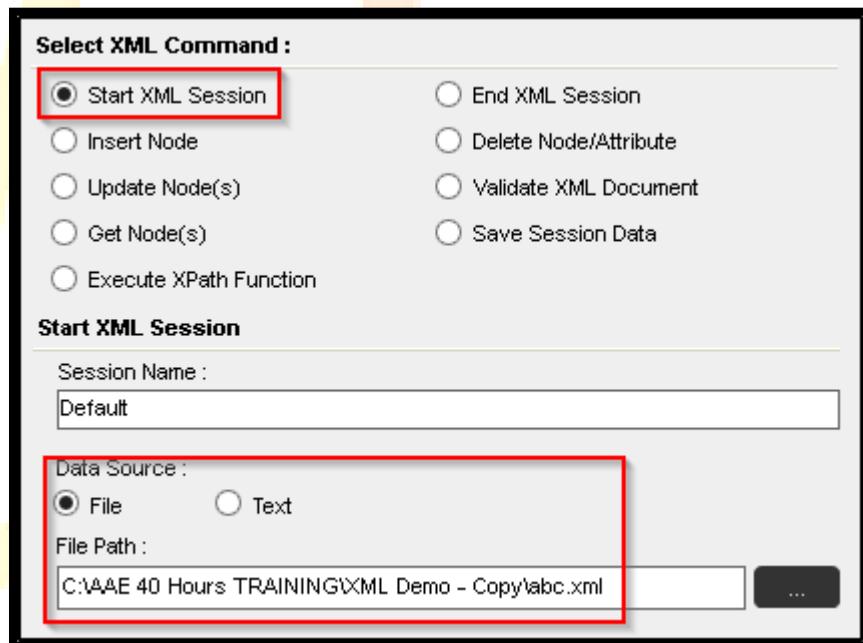
```

<?xml version="1.0" encoding="UTF-8"?>
- <GeoIP xmlns="http://www.webservicex.net/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ReturnCode>1</ReturnCode>
  <IP>27.54.171.210</IP>
  <ReturnCodeDetails>Success</ReturnCodeDetails>
  <CountryName>India</CountryName>
  <CountryCode>IND</CountryCode>
</GeoIP>
  
```

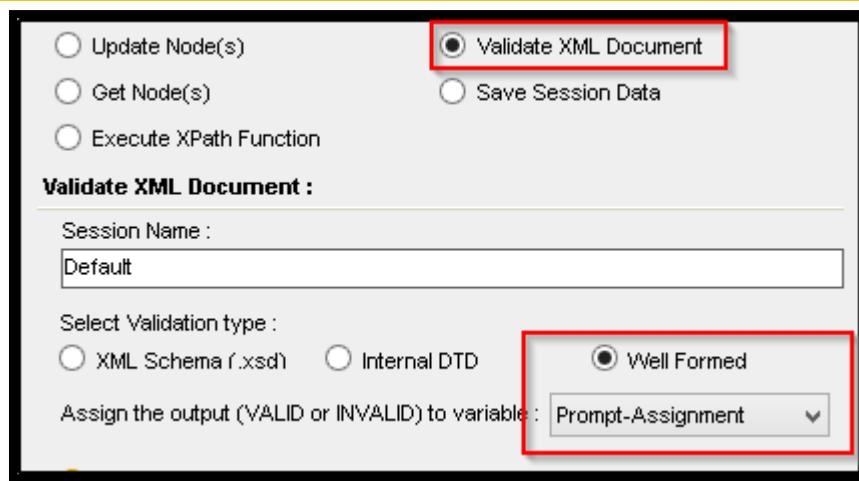
- First, we must open a connection to the XML document for which we use the “XML – Start XML Session” command as below



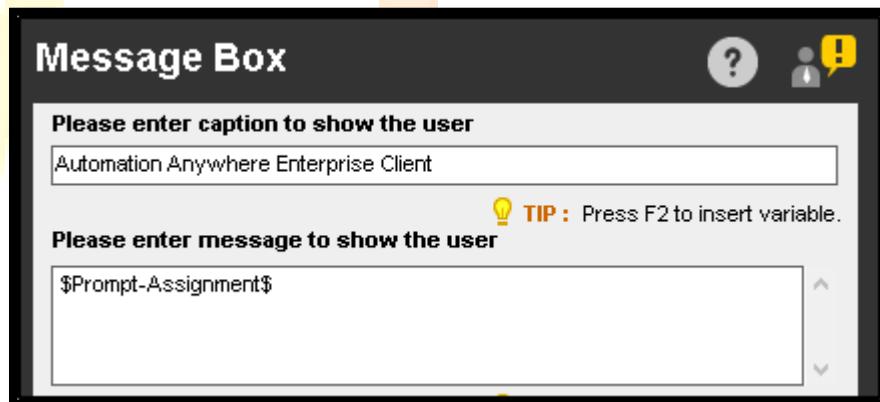
- This opens the following dialog as below, where we point to our document shown in step number 3 above.



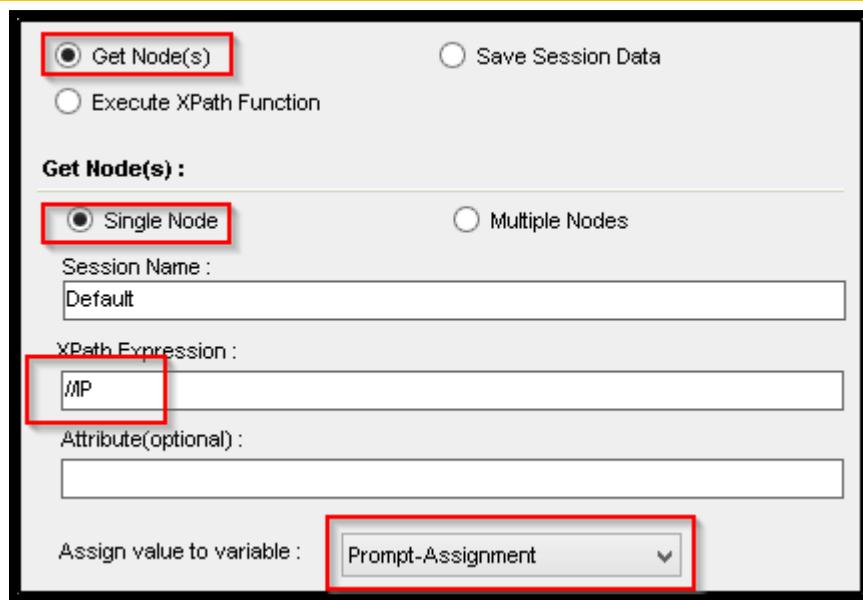
- First thing, we want to validate is, that this document is valid. For that we will bring in an “XML – Validate XML Document” command as seen below:



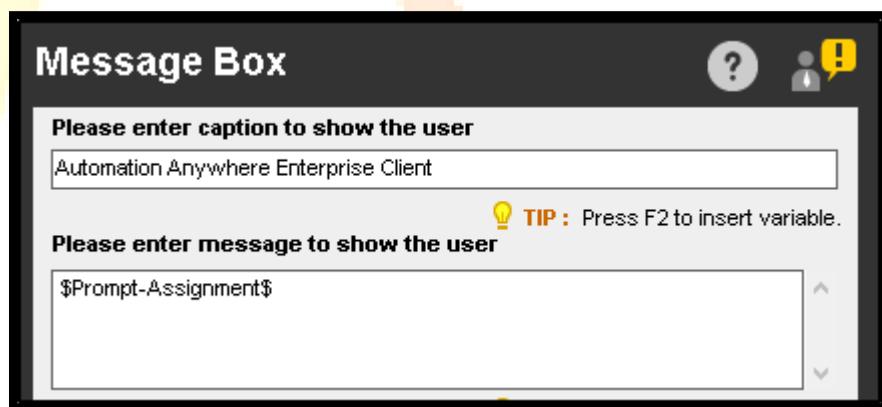
- To display the value of the captured “Prompt Assignment” variable , we will call a message box command as seen below , we also put an “XML – End XML Session” command at the end of our script :



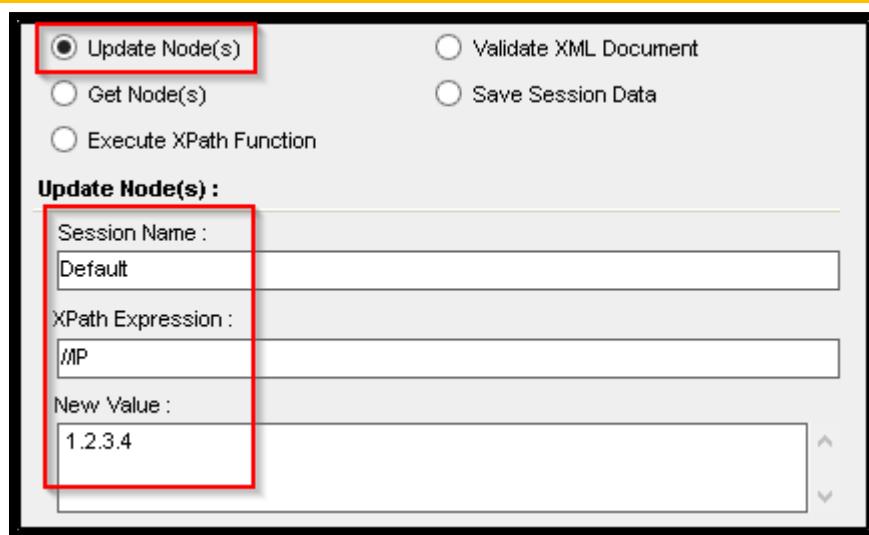
- Save and Run , We see “VALID” getting displayed .
- Let us say we want to capture and display the value of the “**<IP>27.54.171.210</IP>**” IP Node element , we can use The “XML – Get Node(s)” command for that as below , which we keep before the “XML – End XML Session” command , all commands in following steps are added after the commands in the preceding step but before the “XML – End XML Session” command .



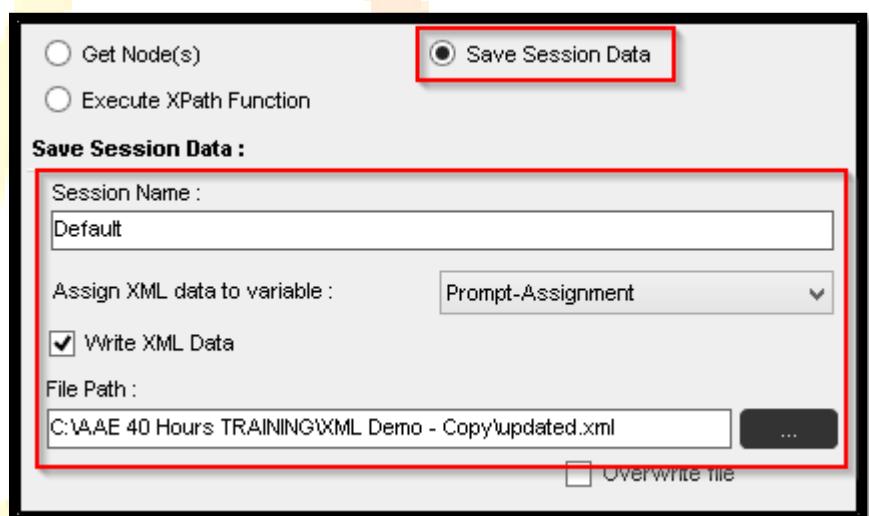
- To display the value of the captured “Prompt Assignment” variable , we will call a message box command as seen below :



- We disable the “Validate XML ” command and it’s “Messagebox” , Save and Run the script . This displays the value of the IP Node “27.54.171.210” as seen here.
- We disable the “Get Node” command and it’s “Message box” also after running it.
- Now, let us say we want to update the value of the IP Node “27.54.171.210” with some other values, say “1.2.3.4”
- For that, we use the “XML – Update Node(s)” command, as below



- To save this updated value in an XML document on the hard disk, we use the “XML – Save Session Data” command as below



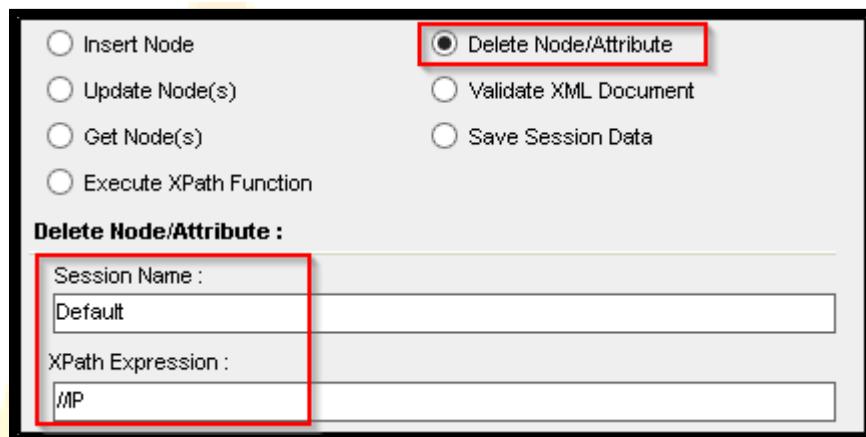
- We again save and run the script which provides us with the updated document, which we open in the browser and look, as below:

```

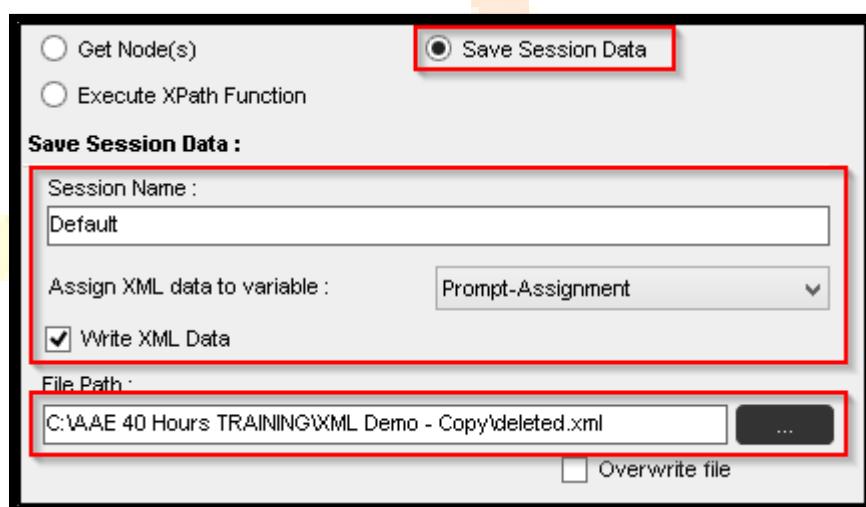
<?xml version="1.0" encoding="UTF-8"?>
- <GeoIP xmlns="http://www.webservicex.net/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ReturnCode>1</ReturnCode>
    <IP>1.2.3.4</IP>
    <ReturnCodeDetails>Success</ReturnCodeDetails>
    <CountryName>India</CountryName>
    <CountryCode>IND</CountryCode>
  </GeoIP>

```

- We can see above , the value of the IP Node has been updated .
- Disable the “Update Node” command and it’s subsequent “Save Session Data” command .
- Now we want to delete the IP Node altogether , for that we call the “XML – Delete Node” command , as seen below



- To save this deleted value in an XML document on the hard disk, we use the “XML – Save Session Data” command as below



- When we run it , it creates a deleted.xml document , which does not contain the IP node as seen below

```

<?xml version="1.0" encoding="UTF-8"?>
- <GeoIP xmlns="http://www.webservicex.net/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ReturnCode>1</ReturnCode>
  <ReturnCodeDetails>Success</ReturnCodeDetails>
  <CountryName>India</CountryName>
  <CountryCode>IND</CountryCode>
</GeoIP>

```

- We then disable the “Delete Node” command, and “Save Session Data” command post this successful run.
- Our script at this point looks like as below:

1	XML : Start XML Session using existing file "C:\AAE 40 Hours TRAINING\XML Demo - Copy\abc.xml" Session:Default
2	XML : Validate XML document is Well-formed and assign to variable "\$Prompt-Assignment\$". Session:Default
3	Message Box: "\$Prompt-Assignment\$"
4	XML : Get Single Node value from XPath location "//IP" and assign to variable "\$Prompt-Assignment\$". Session:Default
5	Message Box: "\$Prompt-Assignment\$"
6	XML : Update Node value "1.2.3.4" at XPath location "//IP". Session:Default
7	XML : Save Session Data assigned to variable "\$Prompt-Assignment\$"; Write Session Data into a new file "C:\AAE 40 Hours TRAINING\XML Demo - Copy\updated.xml". Session:Default
8	XML : Delete Node at XPath location "//IP". Session: Default
9	XML : Save Session Data assigned to variable "\$Prompt-Assignment\$"; Write Session Data into a new file "C:\AAE 40 Hours TRAINING\XML Demo - Copy\deleted.xml". Session:Default
10	XML : End XML Session. Session:Default

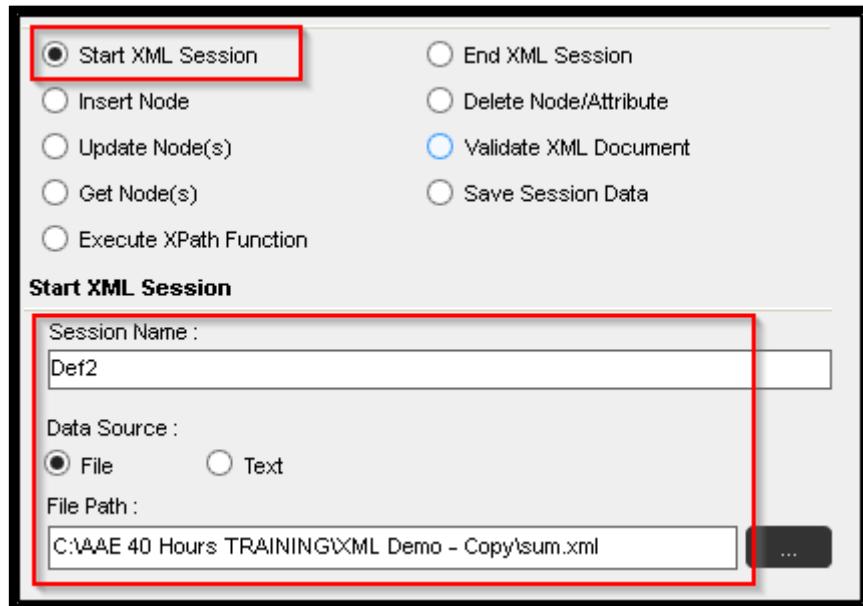
- Now we will disable the entire script made till now, and now we will begin working with the second xml document that we have, as seen below

```

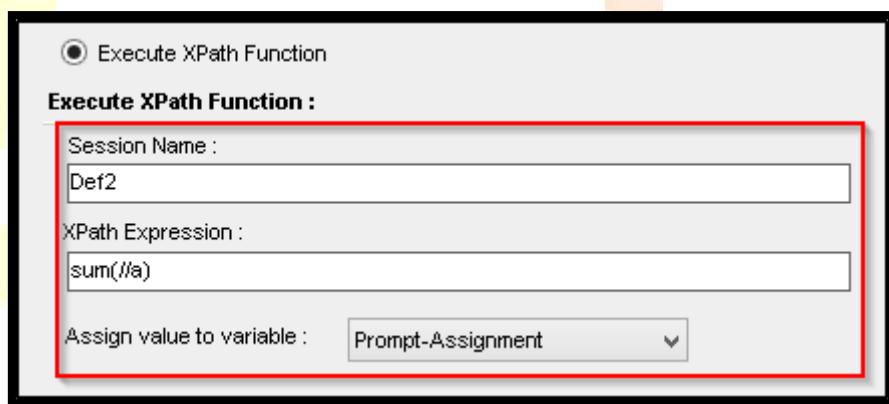
<?xml version='1.0'?>
<?xmlstylesheet type="text/xsl" href="sum.xsl"?>
<root>
  <a>10</a>
  <a>12</a>
  <a>30</a>
  <a>20</a>
</root>

```

- As previously done we will start by bringing in an “XML – Start XML Session” and point to our second xml document shown above:

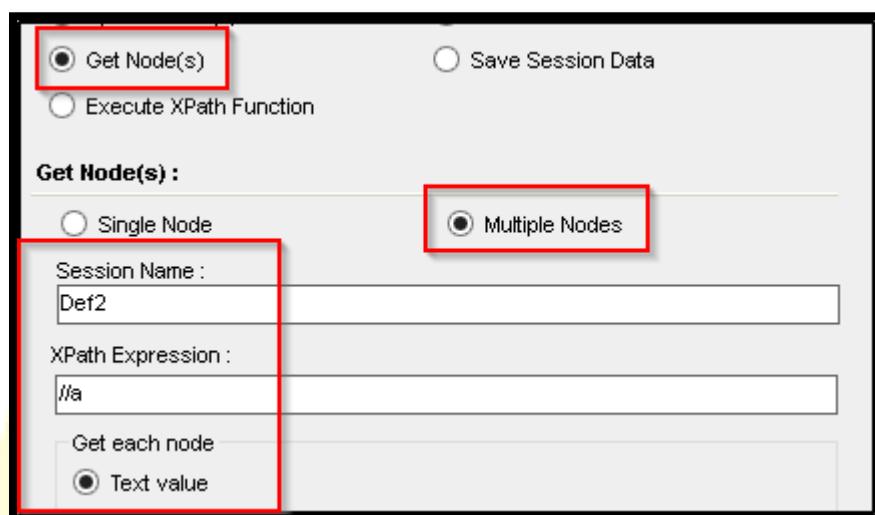


- Our objective, is to find the sum of all nodes of type “`<a>`” and display their total.
- For this we call in the “XML – Execute XPath Function” command as below:

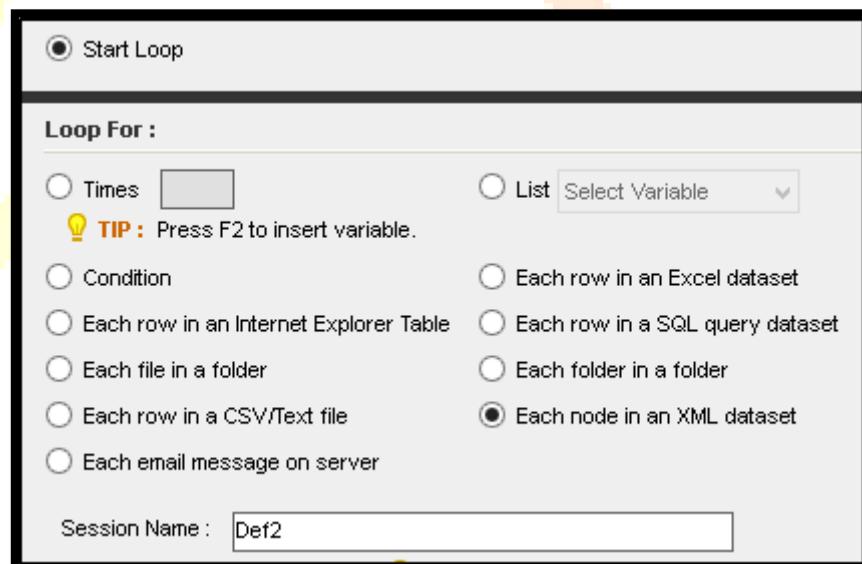


- Call a Message Box , and display the value of the Prompt-Assignment variable .
- Next we call in the “XML – End XML” session command to close the connection as below

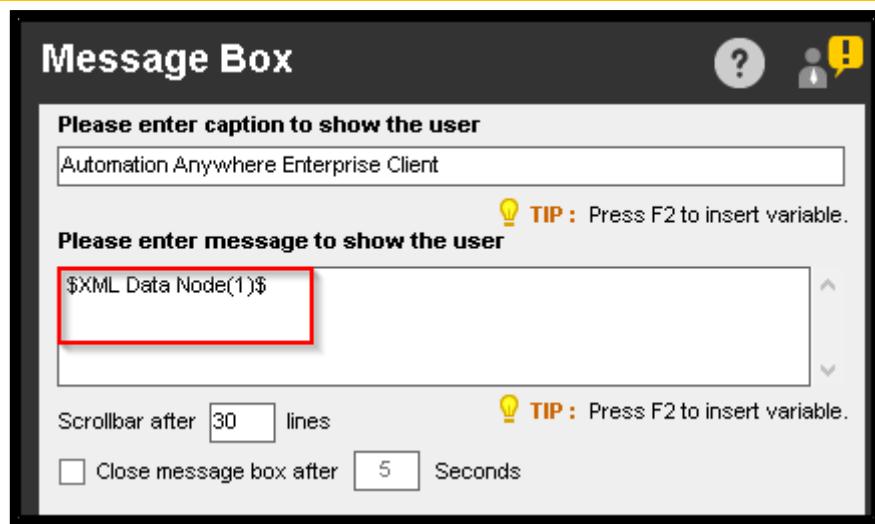
- Next, we save and run the script, we get to see the total of all the nodes of type “<a>”, which is 72.
- Next, disable “Execute XPath” function and “Message box” command.
- We now want to iterate over and display all the values of our xml document one by one, for this we call the “XML – Get Node(s)” command as below



- Next we bring in a “Loop – Each Node in an XML Dataset” command as seen below



- We then call in a Message Box and within it we display the value of the “XML Data Node” variable as seen here



- Save and Run the script, we get to see the values bouncing off our screen one value at a time.
- Entire Script

```

1 XML : Start XML Session using existing file "C:\VAAE 40 Hours TRAINING\XML Demo - Copy\abc.xml" Session:Default
2 XML : Validate XML document is Well-formed and assign to variable "$Prompt-Assignment$". Session:Default
3  Message Box: "$Prompt-Assignment$"
4 XML : Get Single Node value from XPath location "//IP" and assign to variable "$Prompt-Assignment$". Session:Default
5  Message Box: "$Prompt-Assignment$"
6 XML : Update Node value "1.2.3.4" at XPath location "//IP". Session:Default
7 XML : Save Session Data assigned to variable "$Prompt-Assignment$"; Write Session Data into a new file "C:\VAAE 40 Hours TRAINING\XML Demo - Copy\updated.xml". Session:Default
8 XML : Delete Node at XPath location "//IP". Session: Default
9 XML : Save Session Data assigned to variable "$Prompt-Assignment$"; Write Session Data into a new file "C:\VAAE 40 Hours TRAINING\XML Demo - Copy\deleted.xml". Session:Default
10 XML : End XML Session. Session:Default
11 XML : Start XML Session using existing file "C:\VAAE 40 Hours TRAINING\XML Demo - Copy\sum.xml" Session:Def2
12 XML : Execute XPath Function 'sum(/a)' and assign to variable '$Prompt-Assignment$'. Session:Def2
13  Message Box: "$Prompt-Assignment$"
14 XML : Get Multiple Nodes text value from XPath expression "//a". Session:Def2
15  Start Loop "Each Node In a XML Dataset Session: Def2"
16  Comment: Please enter your commands to loop.
17  Message Box: "$XML Data Node(1)$"
18  End Loop
19 XML : End XML Session. Session:Def2

```

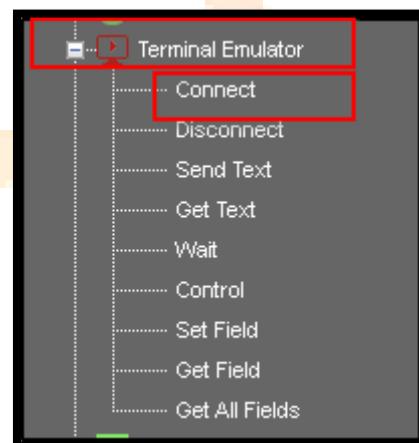
## 14) Terminal Emulator

### Objective

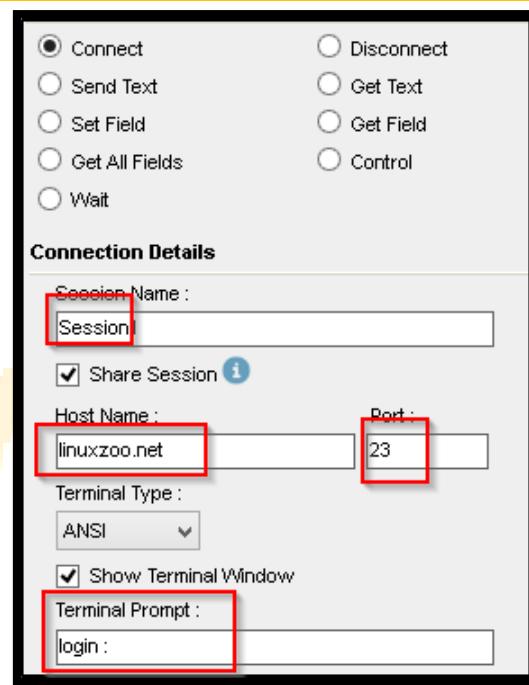
- Step by step demonstration of Terminal Emulator commands like
  - a) Connect/Disconnect to a Unix/Linux environment from AAE client.
  - b) Send Commands into the Linux/Unix Server.
  - c) Capture output from the Linux/Unix Terminal Environment.
  - d) Wait for an action to take place.
  - e) Clearing Screen, Setting Cursor position etc.

### Example: Terminal Emulator

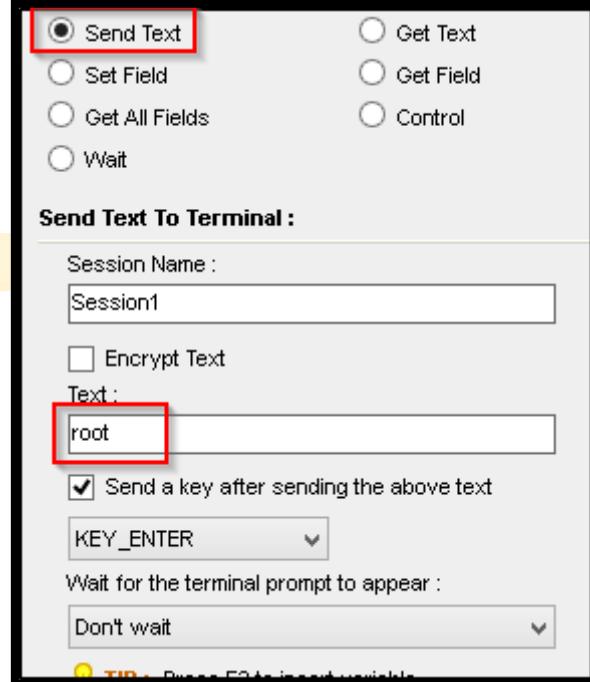
- As usual we will first proceed to open the Automation Anywhere Enterprise client after ascertaining that the Automation Anywhere Control Room Service is up and running.
- We open the Task Editor to create a new task for this activity.
- We next drag a “Terminal Emulator – Connect” command into the Task Actions List area as below:



- This opens the following dialog as below :

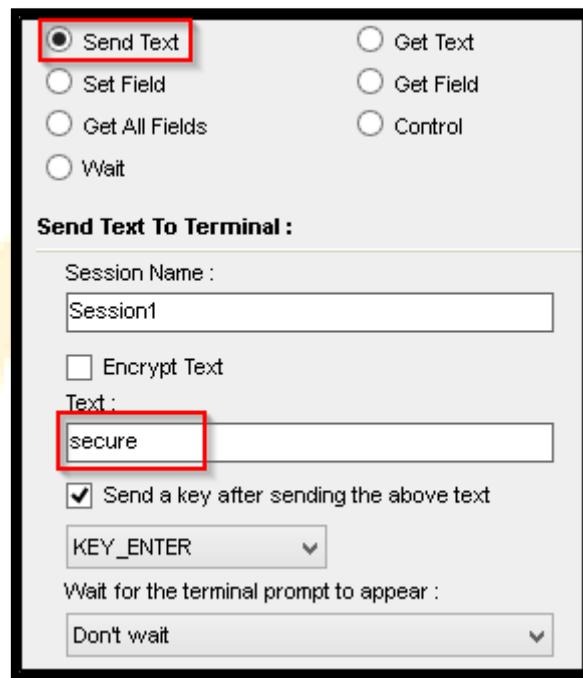


- Next we will drag a “Terminal Emulator – Send Text “command, command to the right, so that we can key in our user name to facilitate automatic login into the Linux/Unix box as below:

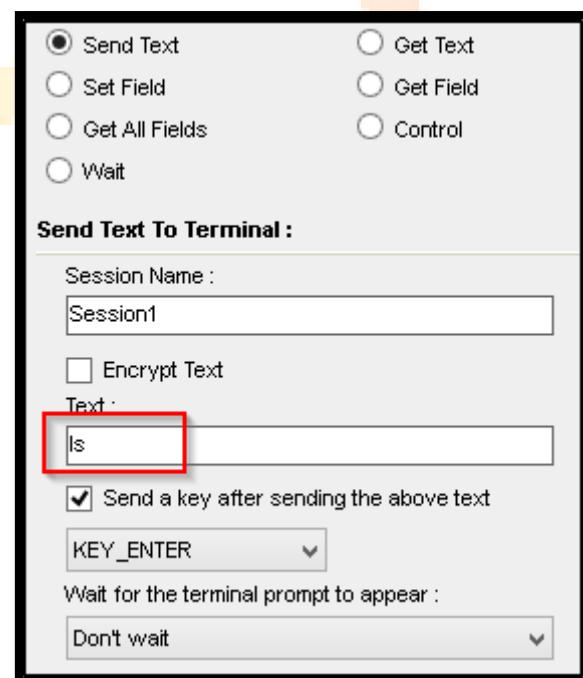


- After we have sent in our user name, the next thing that we want to send in is the password, we again drag a “Terminal Emulator – Send Text “command, command to

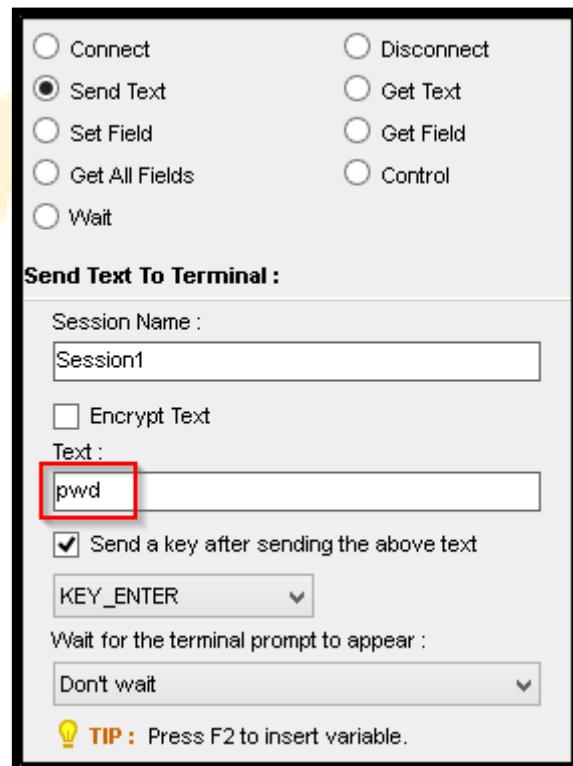
the right, so that we can key in our password to facilitate automatic login into the Linux/Unix box as below:



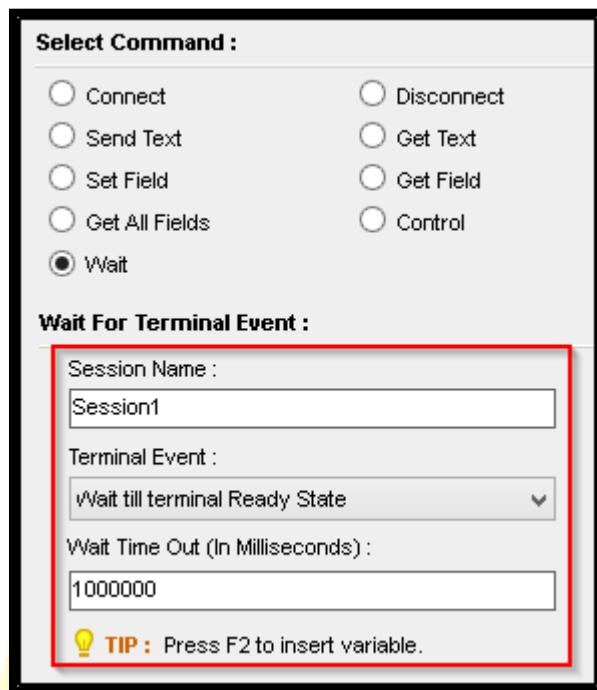
- Next we can pass in some commands that we want should be executed on the Linux/Unix box, for that again we use the “Terminal Emulator – Send Text” command as below:



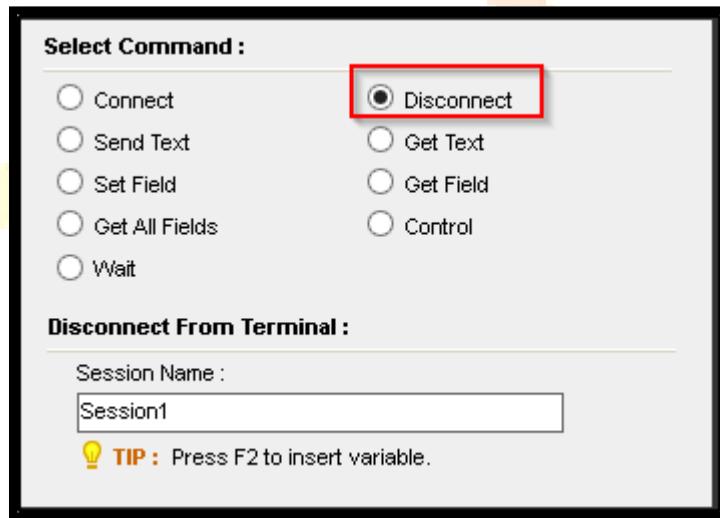
- We again drag a “Terminal Emulator – Send Text” command, and try to execute some other Linux/Unix command here as such as below



- Next we call in a “Terminal Emulator – Wait” command to make the script wait for some time as per our action specified, as below:



- Next we call in a “Terminal Emulator – Disconnect” statement to close the connection with the Linux/Unix environment



- Next, we save the script and run it, it connects to our Linux box and logs in, waits as per our specification and disconnects.

- In Linux, we have a feature to share a session across more than a single script, by

checking the “ Share Session 

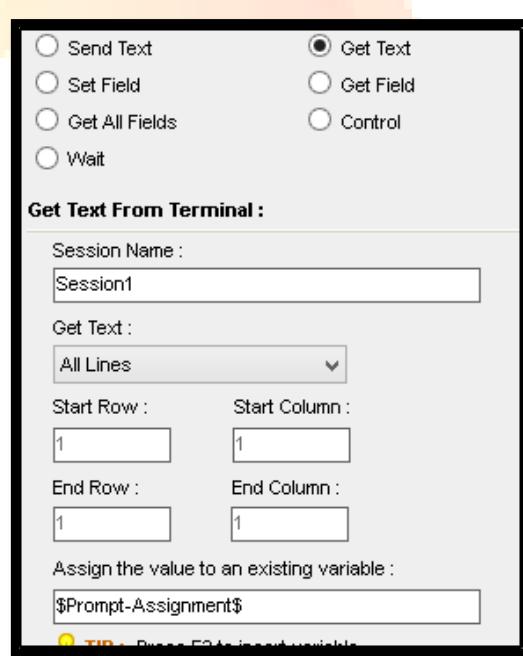
First Script

```

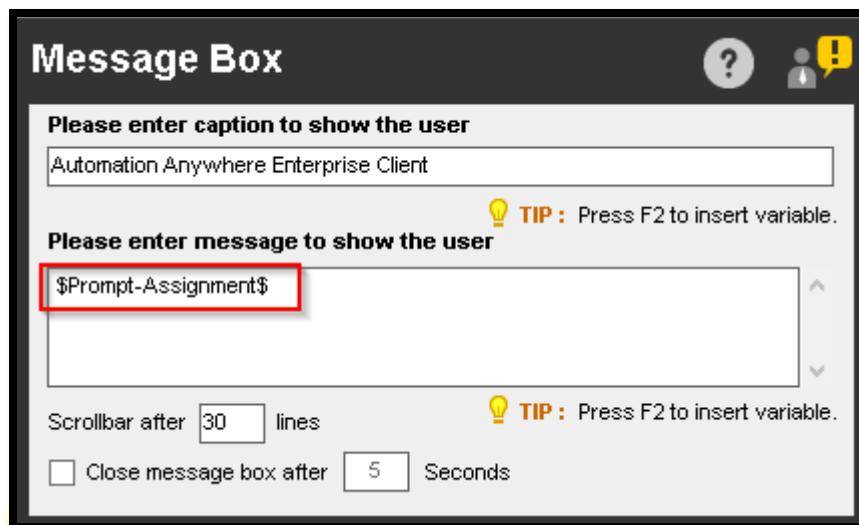
1 Terminal Emulator : Connect to Terminal "linuxzoo.net"; Type: "ANSI"; Session: "Session1"
2 Terminal Emulator: Send Text "root" followed by "KEY_ENTER" to Terminal; Session : "Session1"
3 Terminal Emulator: Send Text "secure" followed by "KEY_ENTER" to Terminal; Session : "Session1"
4 Terminal Emulator: Send Text "ls" followed by "KEY_ENTER" to Terminal; Session : "Session1"
5 Terminal Emulator: Send Text "pwd" followed by "KEY_ENTER" to Terminal; Session : "Session1" 
6 Terminal Emulator : Wait till terminal prompt appears on terminal screen; Session:"Session1"
7 Terminal Emulator : Disconnect from Terminal; Session: "Session1"

```

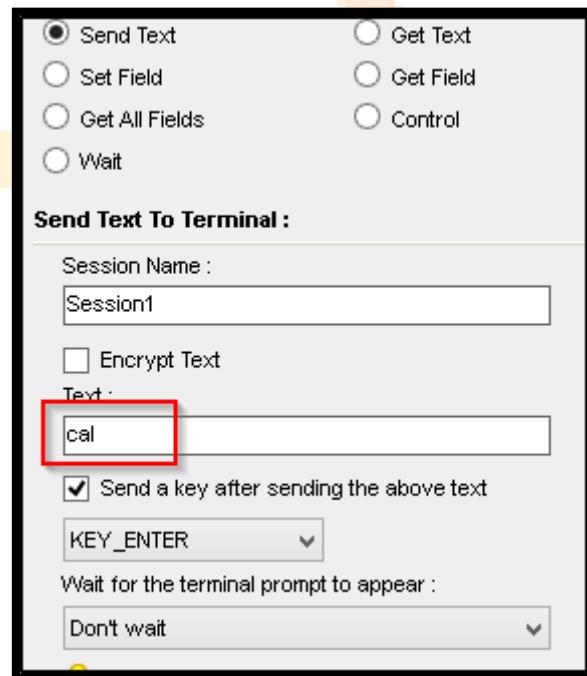
- We launch a new Tab inside the Task Editor by clicking on the “ New” option.
- Inside of our new tab, we create another script that will use the same connection as specified in the previous script
- This time inside our second script we call in the “Terminal Emulator – Get Text” command, which allows us to capture the contents of the Terminal Emulator environment as below



- Next we call in a “Terminal Emulator – Message Box” command, to display the information captured inside the Prompt Assignment variable in the Previous step.



- Again, we call in a “Terminal Emulator – Send Text” command to execute a command on the Linux/Unix emulator as below



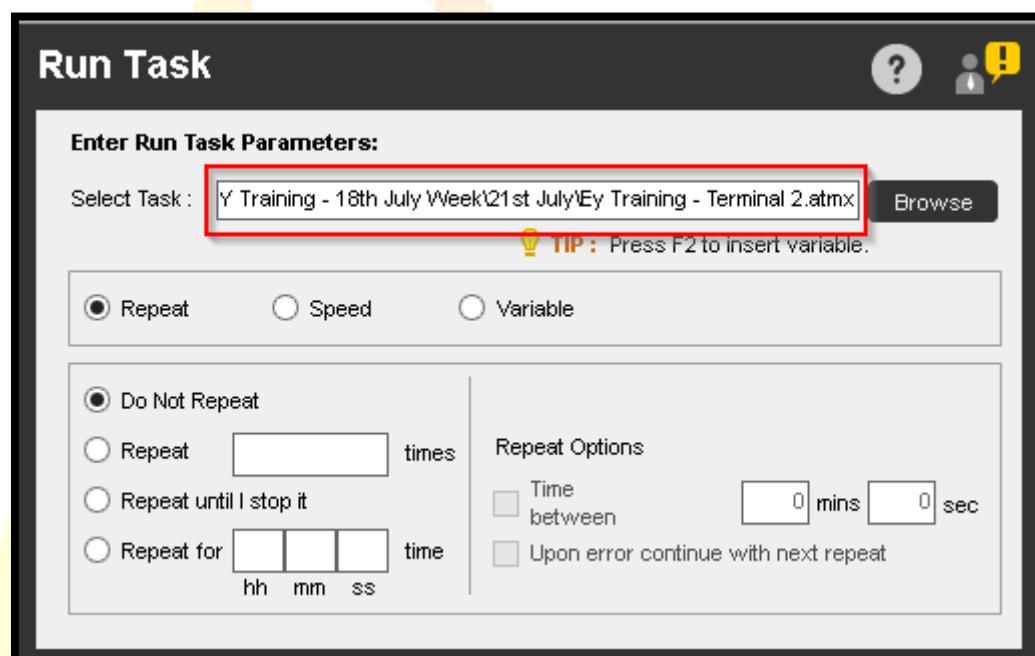
## Second Script

```

1  Terminal Emulator : Get text from Terminal to "$Prompt-Assignment$"; Session:"Session1"
2  Message Box: "$Prompt-Assignment$"
3  Terminal Emulator: Send Text "cal" followed by "KEY_ENTER" to Terminal; Session : "Session1"

```

- To send control from the first script we put in a Run Task command, after line no 4 in the first script, and point to the second script made above:



Now when we run the first script it runs till line 4 , then control goes to second script which gets executed and then control returns to line 5 of first script and starts running from there .

## First Script

```

1  Terminal Emulator : Connect to Terminal "linuxzoo.net"; Type: "ANSI"; Session: "Session1"
2  Terminal Emulator: Send Text "root" followed by "KEY_ENTER" to Terminal; Session : "Session1"
3  Terminal Emulator: Send Text "secure" followed by "KEY_ENTER" to Terminal; Session: "Session1"
4  Terminal Emulator: Send Text "ls" followed by "KEY_ENTER" to Terminal; Session : "Session1"
5  Run Task "C:\Users\ajay.mehta\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\EY Training - 18th July Week\21st July\Ey Training - Terminal 2.ahmx" @Repeat: Do Not Repeat @S
6  Terminal Emulator: Send Text "pwd" followed by "KEY_ENTER" to Terminal; Session : "Session1"
7  Terminal Emulator : Wait till terminal prompt appears on terminal screen; Session:"Session1"
8  Terminal Emulator : Disconnect from Terminal; Session: "Session1"

```

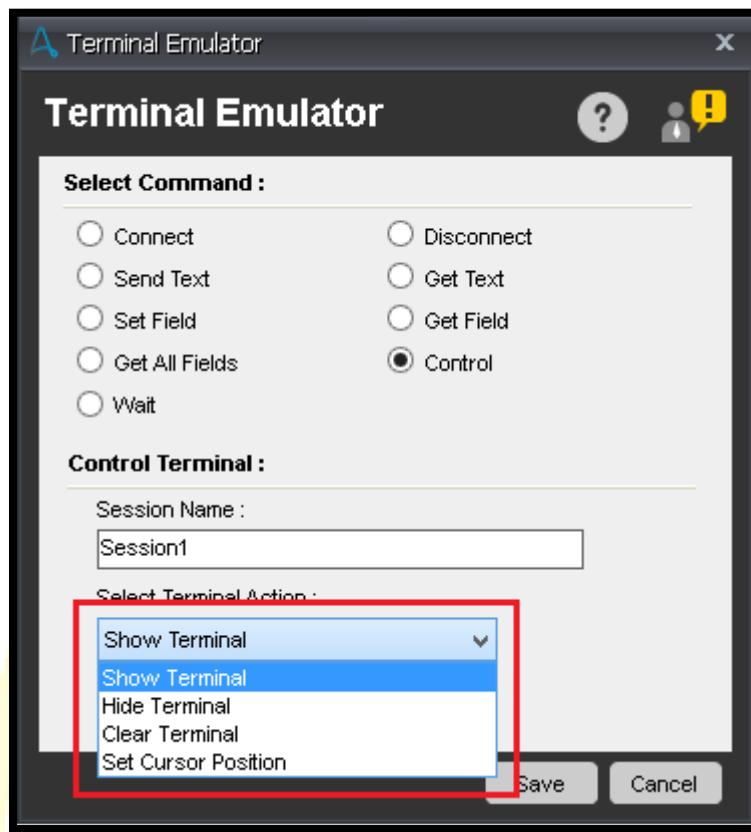
## Second Script

```

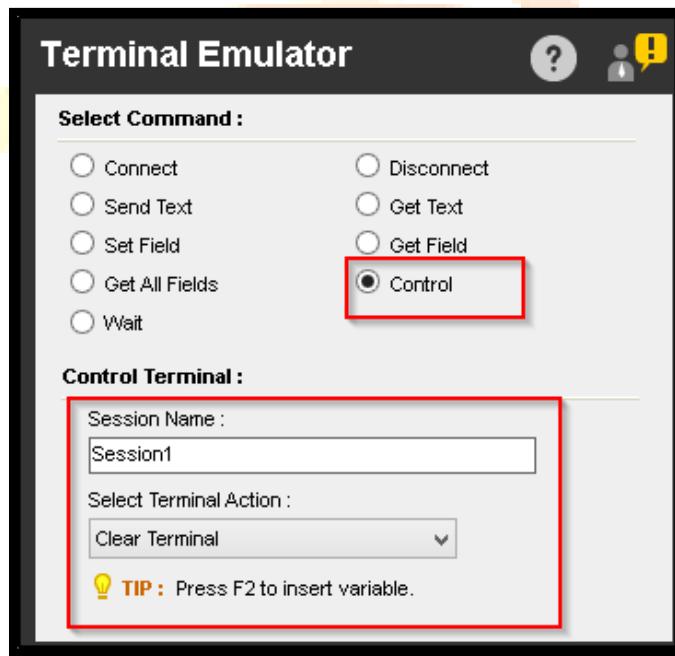
1  Terminal Emulator : Get text from Terminal to "$Prompt-Assignment$"; Session:"Session1"
2  Message Box: "$Prompt-Assignment$"
3  Terminal Emulator: Send Text "cal" followed by "KEY_ENTER" to Terminal; Session : "Session1"

```

- In the first script , after line 7 , we now proceed to add “Terminal Emulator – Control” command which will allow us to perform the following actions as below



- We chose the “Clear Terminal” action as seen below



- Our First script looks like as below:

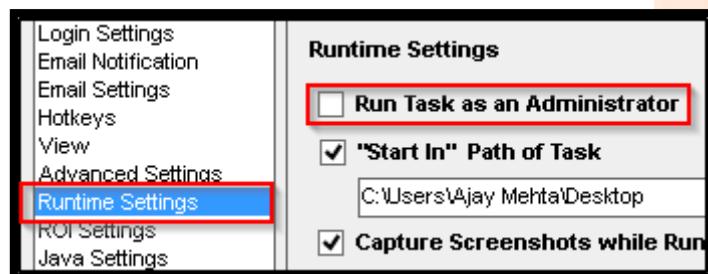
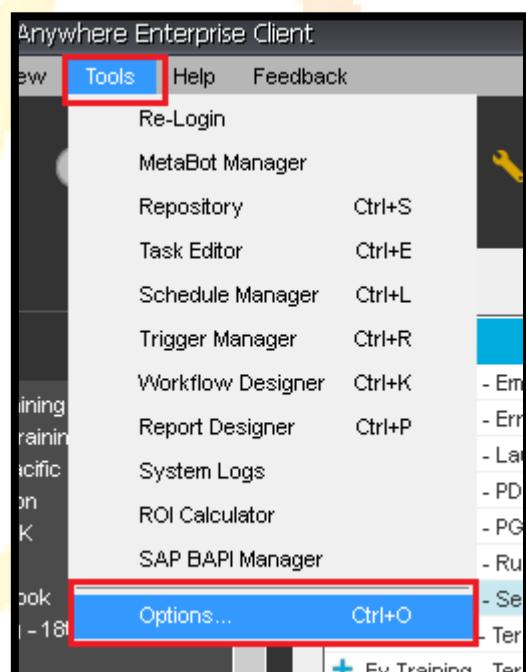
```
1  [!] Terminal Emulator : Connect to Terminal "linuxzoo.net"; Type: "ANSI"; Session: "Session1"
2  [!] Terminal Emulator: Send Text "root" followed by "KEY_ENTER" to Terminal; Session : "Session1"
3  [!] Terminal Emulator: Send Text "secure" followed by "KEY_ENTER" to Terminal; Session : "Session1"
4  [!] Terminal Emulator: Send Text "ls" followed by "KEY_ENTER" to Terminal; Session : "Session1"
5  [!] Run Task "C:\Users\rajay.mehta\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\EY Training - 18th July Week\21st July\Ey Training - Terminal 2.atmx" @Repeat: Do Not Repeat @S
6  [!] Terminal Emulator: Send Text "pwd" followed by "KEY_ENTER" to Terminal; Session: "Session1"
7  [!] Terminal Emulator : Wait till terminal ready state; Session:"Session1"
8  [!] Terminal Emulator : Clear Terminal; Session:"Session1"
9  [!] Terminal Emulator : Disconnect from Terminal; Session: "Session1"
```

- Note : Actions like “Set Field”, “Get Field”, “Get All Fields” can work on some specific Terminal types only.

## 15) Services

### Objective

- Step by step demonstration of Services command and its various facets, like:
  - a) Getting the status of a service.
  - b) Starting/Stopping a service
- Note: Starting/Stopping a service requires “Elevated Privileges” / “Administrator Rights”, the same should be held by our windows login credentials, so that the software using our credentials can start/stop the service. Alternatively, we can launch the AAE client using “Run as an Administrator” option or go to “Tools -> Options” in the client as below



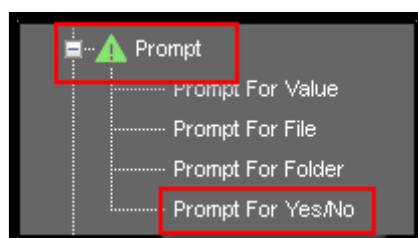
- And check mark “Runtime Settings - > Run Task as an Administrator” as seen above.

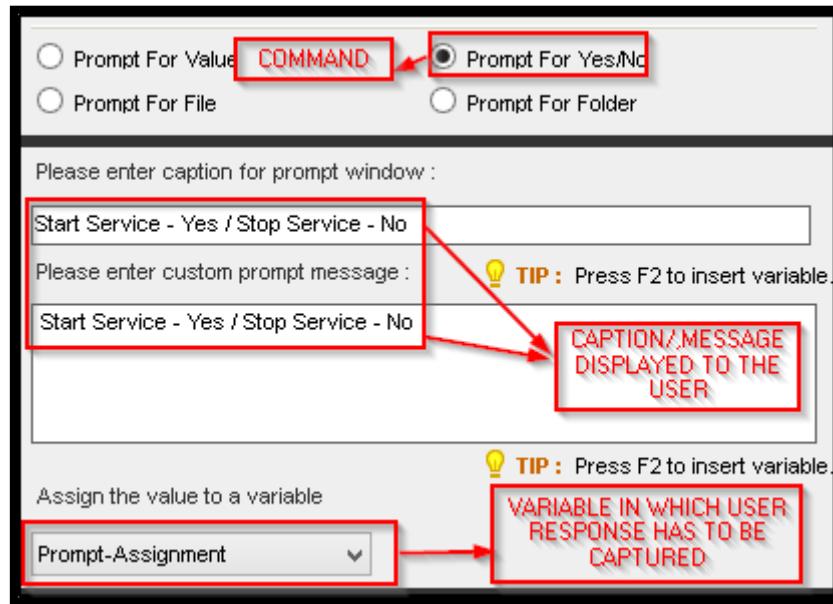
## Example: Services

- As usual we will first proceed to open the Automation Anywhere Enterprise client after ascertaining that the Automation Anywhere Control Room Service is up and running.
- We open the Task Editor to create a new task for this activity.
- As the objective, here, is to start/stop a service, let us first decide which service we want to be able to start/stop, idea is not to crash the system by altering the state of any service which is required for its smooth working.
- We click on start->run->type “services.msc” which takes us into the services section where we have all the services and their current run status getting reflected as below

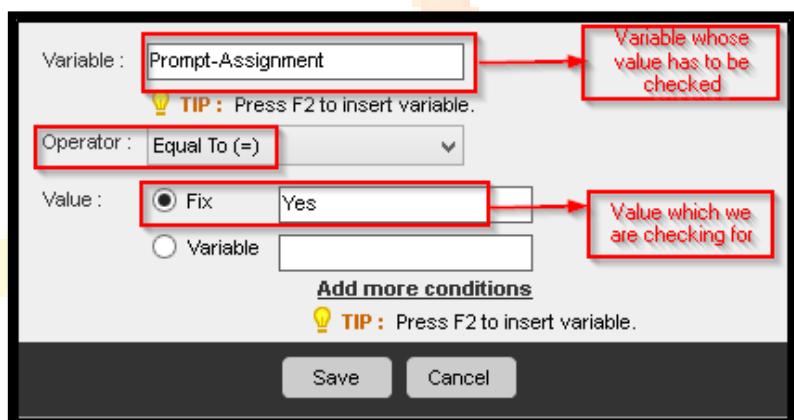
Automation Anywhere Auto Login Service	Provides support for Auto Login functionality in Automation Anywhere	
Automation Anywhere Enterprise Auto Login Service	Provides support for Auto Login functionality in Automation Anywhere Enterprise Client	Running
Automation Anywhere Enterprise Client Service	Provides Automation Anywhere Client functionality	Running
Automation Anywhere Enterprise Scheduler Service	Provides Automation Anywhere Enterprise Scheduler functionality	Running
Automation Anywhere Small Business Auto Login Service	Provides support for Auto Login functionality in Automation Anywhere Small Business Client	
Automation Anywhere Small Business Server Service	Provides Automation Anywhere Small Business Server functionality	
Automation Anywhere Web Socket Server Service	Provides Automation Anywhere Web Socket Server functionality	Running
Background Intelligent Transfer Service	Transfers files in the background using idle network bandwidth. If the service is disabled, then an... Windows infrastructure service that controls which background tasks can run on the system.	Running
Background Tasks Infrastructure Service		Running
Base Filtering Engine	The Base Filtering Engine (BFE) is a service that manages firewall and Internet Protocol security (I... Running	

- We will attempt to start/stop the “Automation Anywhere Enterprise Scheduler Service” using our “Services” command , based on user inputs.
- We will begin by asking the user if he wants to Start/Stop the service , for this we will bring in an “Prompt – Prompt for Yes/No” command as seen below:

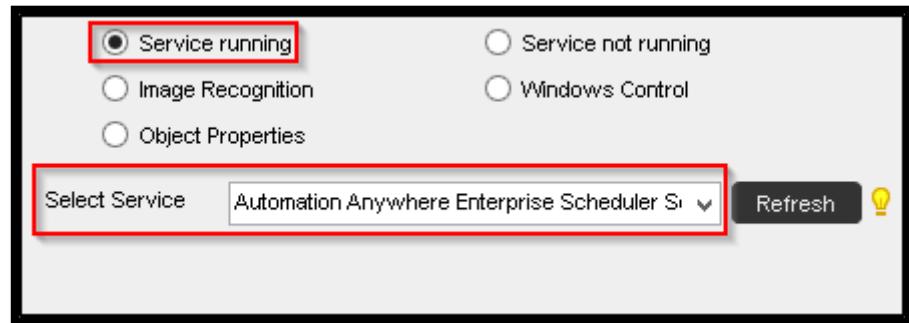




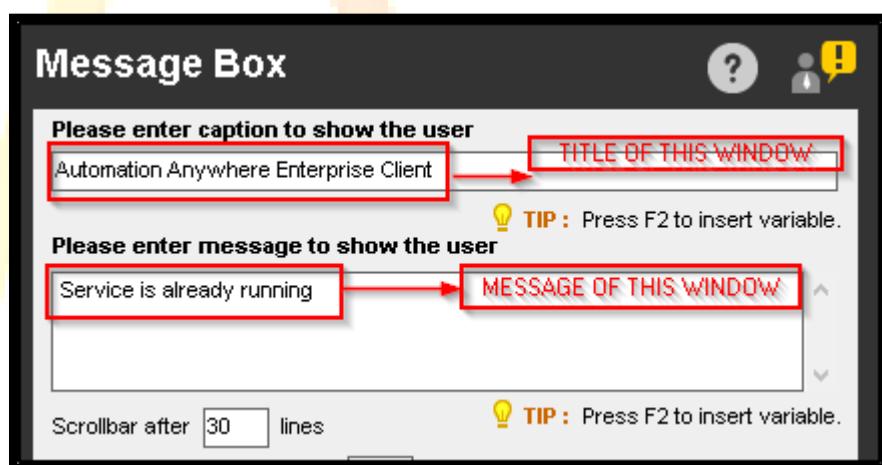
- After catching the user response , we need to check what was it that the user wanted , start or stop the service , for this we will bring in an “If/Else – Variable” command and make the changes as below :



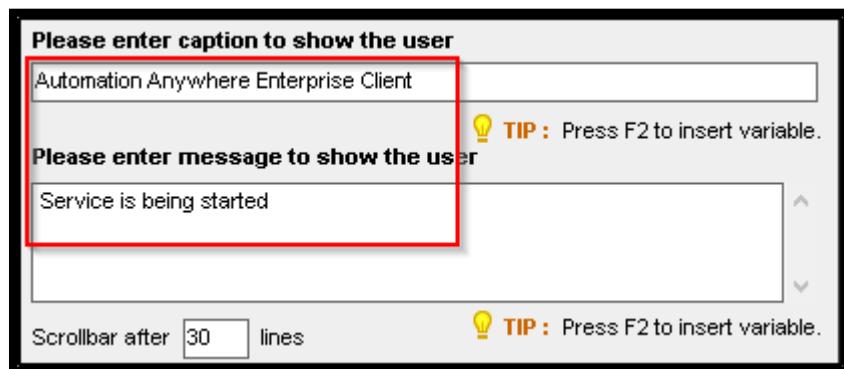
- The above line will give us an “If – End if “ block . Within the If block created above we will bring in an “If/Else – Service Running” block and check if the “Scheduler Service” is running as below :



- If the above service is running we will throw a message box on the screen , telling that the service is running successfully as below :



- Then we will drag an “If/Else –Else” to the right .
- Next , we will drag a Message Box following the “Else” , within this Message Box we will display a message for the scenario that the “Service was not running ” and “We are starting the service” as below



- Following this Message Box , we will drag in an “Services – Start Service” command , which will start the service



- Our code looks like as below, now:

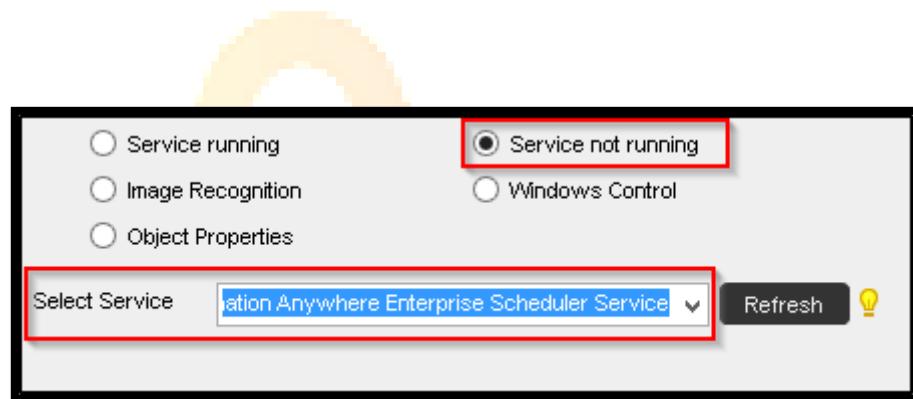
```

1  ! Prompt For Yes/No: "Start Service - Yes / Stop Service - No" for File Assign to variable "$Prompt-Assignment$"
2  IF $Prompt-Assignment$ Equal To (=) "Yes" Then
3    Comment: Please enter the conditional commands here.
4    IF Service Running ("Automation Anywhere Enterprise Scheduler Service") Then
5      Comment: Please enter the conditional commands here.
6      Message Box: "Service is already running"
7    ELSE
8      Message Box: "Service is being started"
9      Start Service: "Automation Anywhere Enterprise Scheduler Service"
10   End If

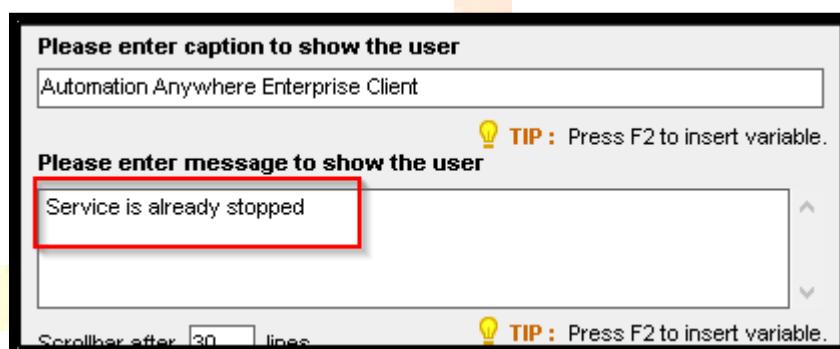
```

- So far, we have put in the code if the user wanted the service to be started, next we will drag in an “Else” command, for the scenario where the user did not want the service to be started.

- Next we will put in the code for the logic where the user did not want the service to be started.
- As, done last time, we want to first check if first the “Service is not running”, for that we drag in an “If/Else – Service Not Running” command as below, and check if our service is already stopped.



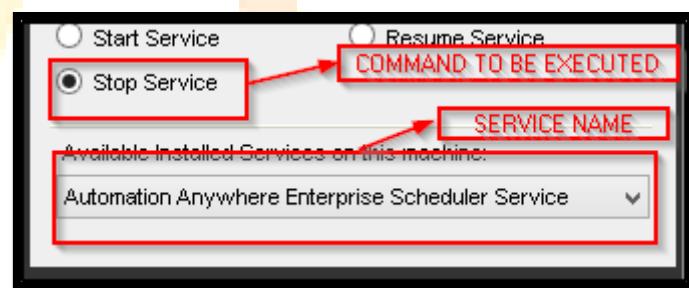
- If the service is already stopped, we bring in a Message Box and display the message “Service is already stopped” as below.



- We next bring in an “If/Else – Else” command, for the possible scenario where the service is running and needs to be stopped
- This we will follow with a “Message Box” command as below, informing all that the service is being stopped.



- Next we will bring in an “Services – Stop Service” command as below to stop the service



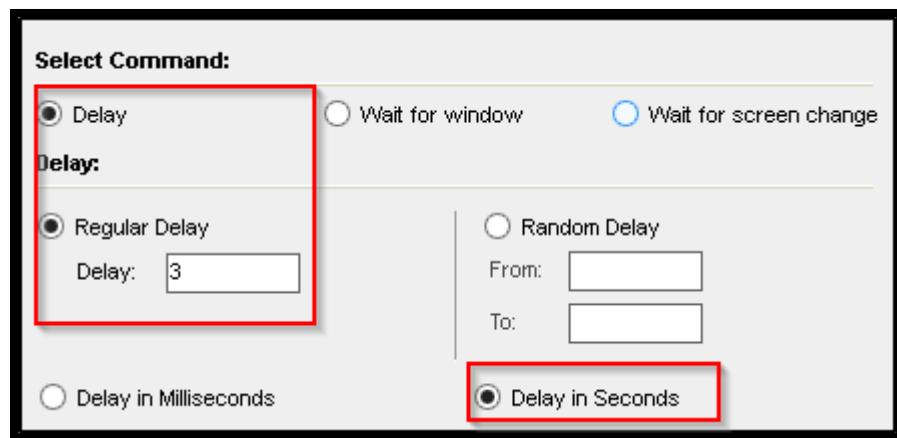
- Our Script looks like as below

```

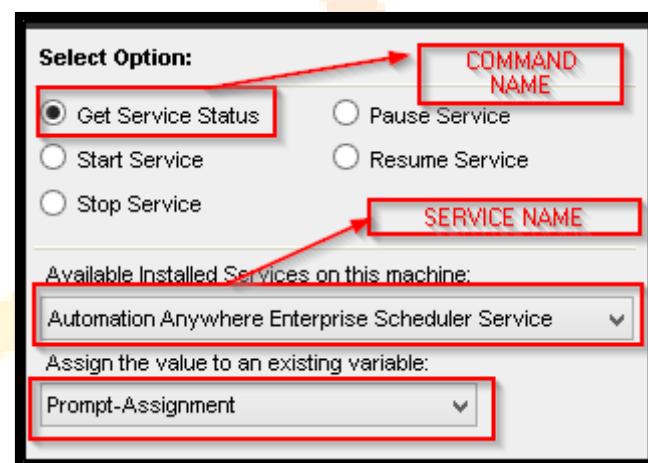
1   Prompt For Yes/No: "Start Service - Yes / Stop Service - No" for File Assign to variable "$Prompt-Assignment$"
2   If $Prompt-Assignment$ Equal To (=) "Yes" Then
3    Comment: Please enter the conditional commands here.
4     If Service Running ("Automation Anywhere Enterprise Scheduler Service") Then
5      Comment: Please enter the conditional commands here.
6      Message Box: "Service is already running"
7    Else
8      Message Box: "Service is being started"
9    Start Service: "Automation Anywhere Enterprise Scheduler Service"
10   End If
11  Else
12     If Service Not Running ("Automation Anywhere Enterprise Scheduler Service") Then
13      Comment: Please enter the conditional commands here.
14      Message Box: "Service is already stopped"
15    Else
16      Message Box: "Service is being stopped"
17    Stop Service: "Automation Anywhere Enterprise Scheduler Service"
18  End If
19 End If

```

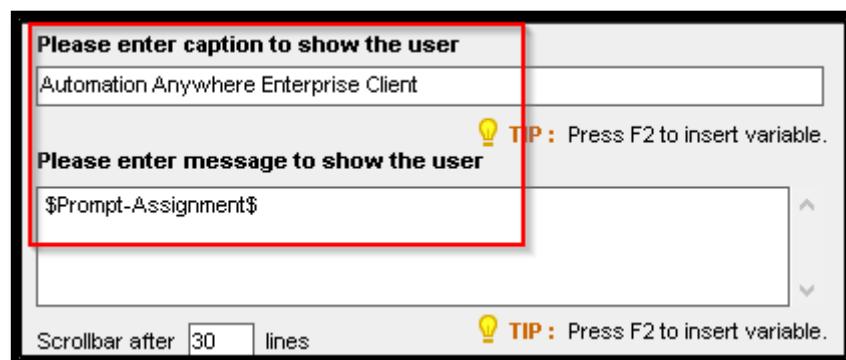
- We will next bring in a Delay command , so that the service in perspective fully starts or stops before we can get its status and display it to the user as below



- To get the status of the service (Started / Stopped / Paused / Resumed) we will use the “Services – Get Service Status” command as below



- The output of the above command is stored in prompt assignment variable , which we will then display via a Message Box as below



- We then “Save” And “Run” the script and we should be able to “Start” or “Stop” our service as per our specifications.

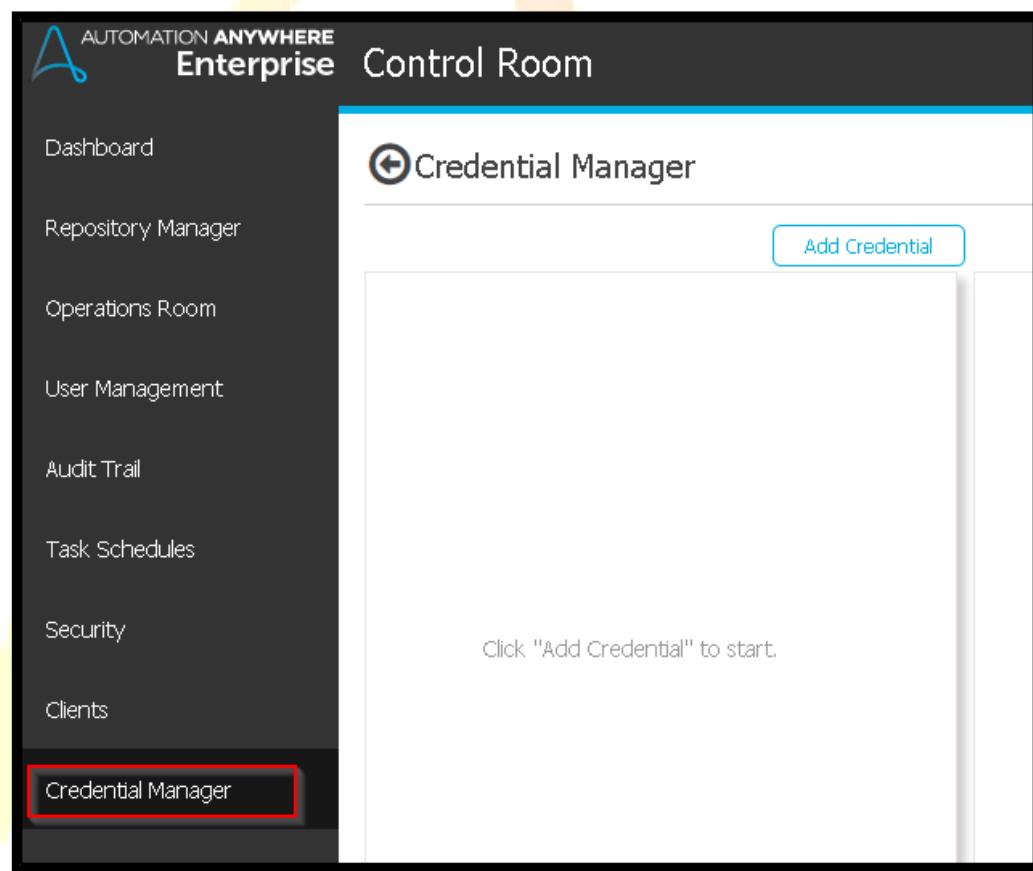
## 16) Credential Manager

### Objective

- Step by step demonstration of Credential Manager and its various facets, like:
  - a) Using the Credential Variables inside a TaskBot.

### Phase 1 - Creating Credential Variables

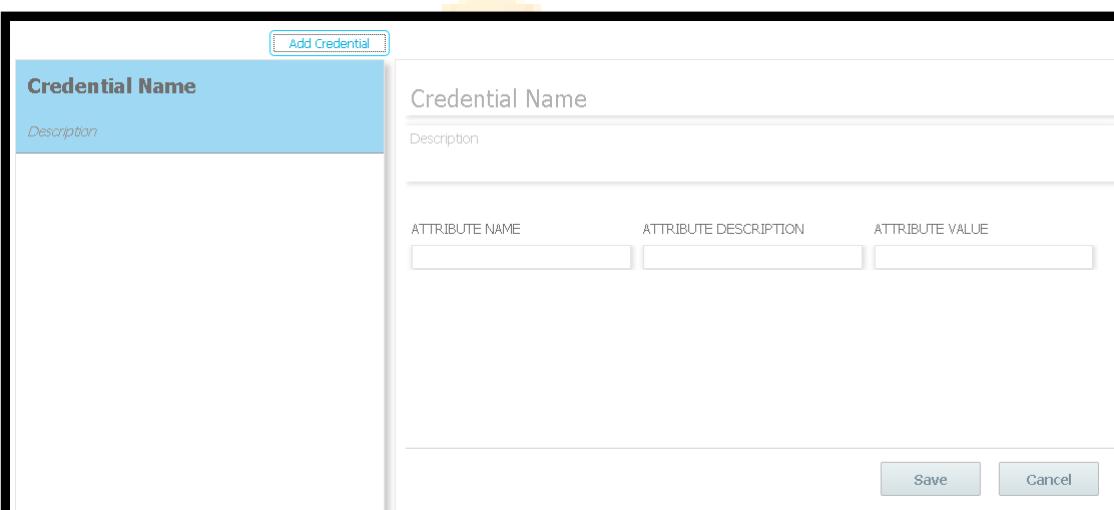
- Open the Automation Anywhere Enterprise Control Room, and login as the Super admin user.
- Choose the Credential Manager tab.



- Further within the Credential Manager tab, we will click on the “” button.

- Store the credentials of the FTP server here.

- Once we click on the “” button , we get to see the following screen as below



**Credential Name**

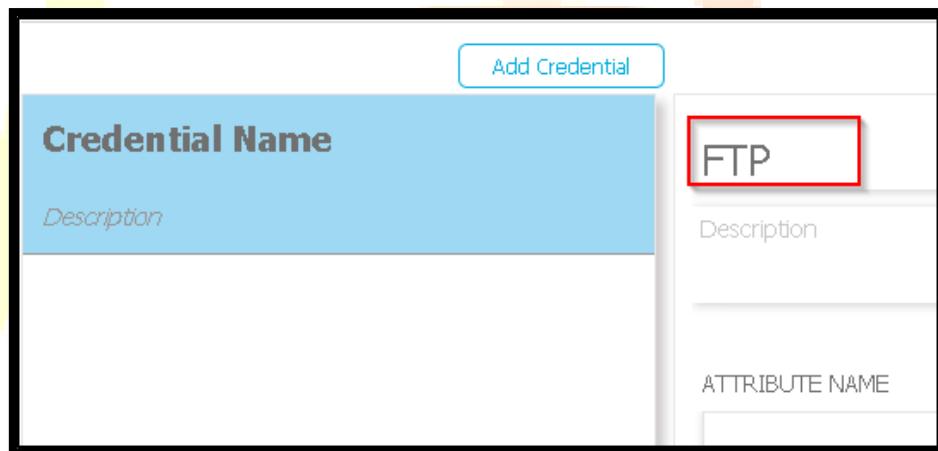
*Description*

**Add Credential**

ATTRIBUTE NAME	ATTRIBUTE DESCRIPTION	ATTRIBUTE VALUE
<input type="text"/>	<input type="text"/>	<input type="text"/>

**Save**    **Cancel**

- Give a credential name



**Credential Name**

*Description*

**Add Credential**

ATTRIBUTE NAME
<input type="text" value="FTP"/>

**Save**    **Cancel**

- Give a description to our Credential.

Add Credential

**Credential Name**

Description

FTP

FTP SERVER CONNECTION DETAILS

- Save credential now, and add values to its attributes later, or go ahead and add the attributes and their values also and save it later.

FTP

FTP SERVER CONNECTION DETAILS

ATTRIBUTE NAME	ATTRIBUTE DESCRIPTION	ATTRIBUTE VALUE
<input type="text"/>	<input type="text"/>	<input type="text"/>

Save      Cancel

⊕ Credential Manager

Add Credential

<b>FTP</b>	
FTP SERVER CONNECTION DETAILS	

FTP SERVER CONNECTION DETAILS	ATTRIBUTE NAME	ATTRIBUTE DESCRIPTION	ATTRIBUTE VALUE
-------------------------------	----------------	-----------------------	-----------------

- Click on “Edit” as below, to be able to add the Attributes to this credential and assign values to the same.

**FTP**

*FTP SERVER CONNECTION DETAILS*

ATTRIBUTE NAME	ATTRIBUTE DESCRIPTION	ATTRIBUTE VALUE

**Edit**

- Add first attribute
  - Attribute Name: Server
  - Attribute Description: Server Name
  - Attribute Value: IP Address of the FTP Server, which is encrypted.

**FTP**

*FTP SERVER CONNECTION DETAILS*

ATTRIBUTE NAME	ATTRIBUTE DESCRIPTION	ATTRIBUTE VALUE
Server	Server Name	*****

- We proceed to add the second attribute
  - a) Attribute Name: User
  - b) Attribute Description: Username
  - c) Attribute Value: Username Value, which is encrypted.

**FTP**

FTP SERVER CONNECTION DETAILS

ATTRIBUTE NAME	ATTRIBUTE DESCRIPTION	ATTRIBUTE VALUE
Server	Server Name	*****
User	UserName	*****

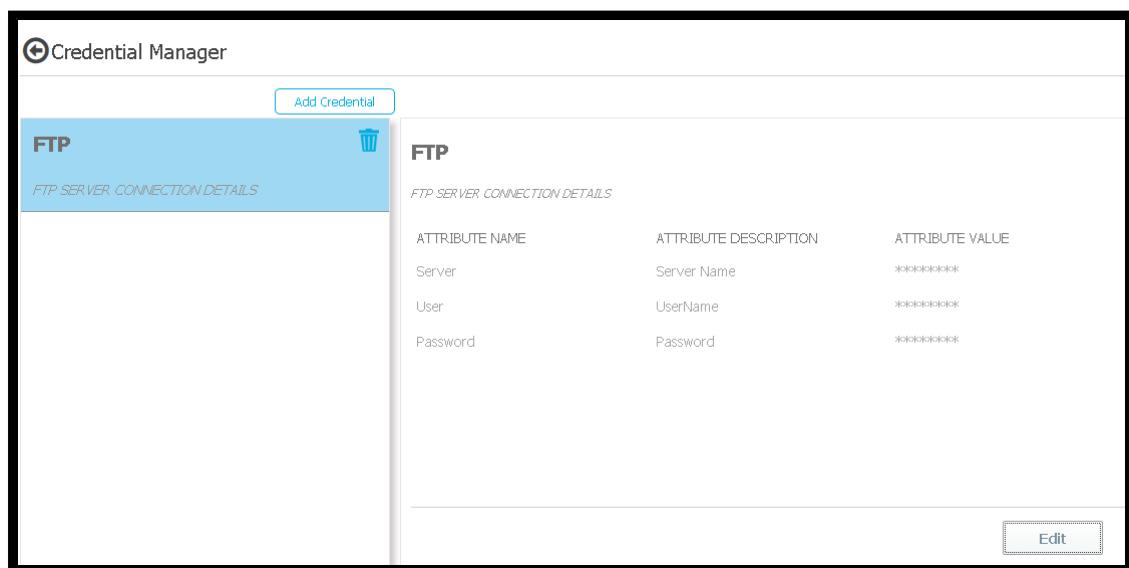
- Add third attribute
  - a) Attribute Name: Password
  - b) Attribute Description: Password
  - c) Attribute Value: Password Value, which is encrypted.

**FTP**

FTP SERVER CONNECTION DETAILS

ATTRIBUTE NAME	ATTRIBUTE DESCRIPTION	ATTRIBUTE VALUE
Server	Server Name	*****
User	UserName	*****
Password	Password	*****

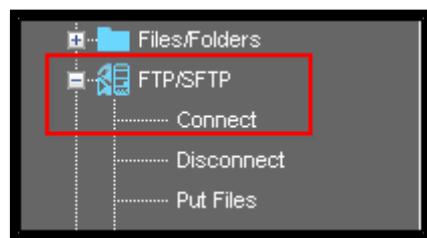
**Save**      **Cancel**



- Credentials have been successfully created and stored in the Web Control Room .

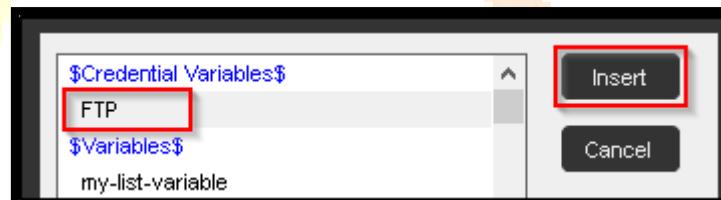
## Phase 2 - Using Credential Variables

- Login into our Automation Anywhere client.
- Click on New, choose “Task Editor”
- To connect to the FTP site using credentials stored in credential manager, choose “FTP/SFTP -> Connect”

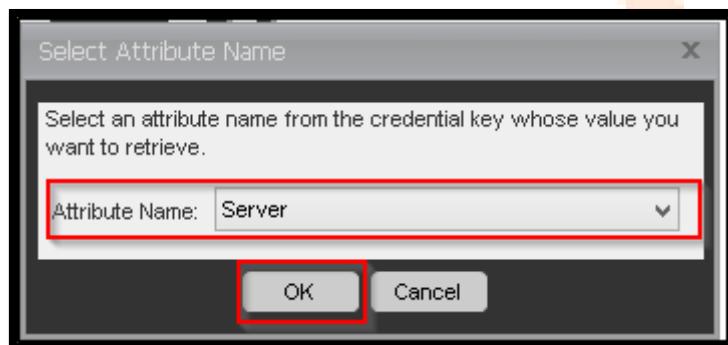


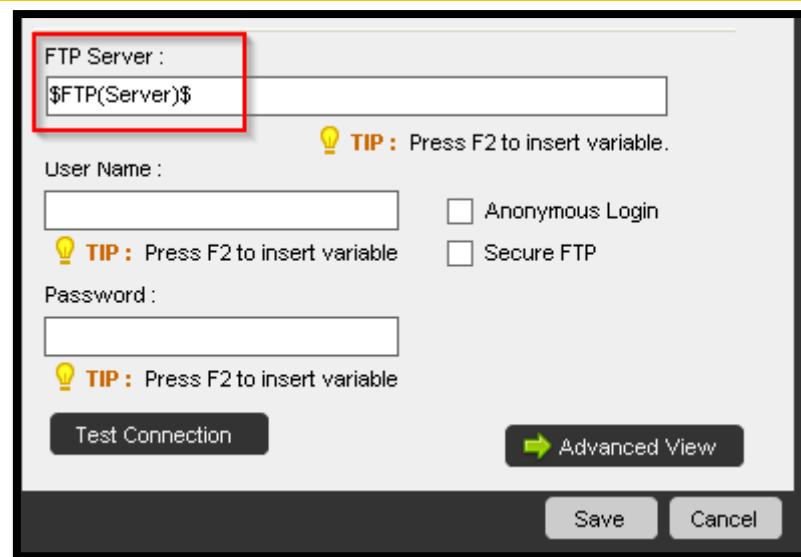


- Inside “FTP Server” textbox, press F2 , which opens the following dialog

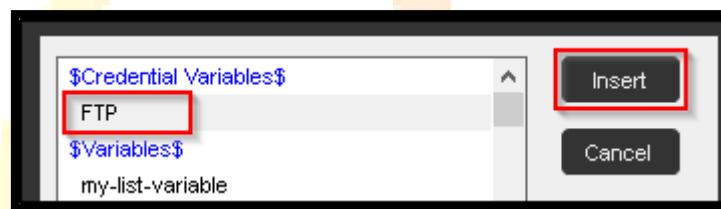


- Choose credential variable “FTP” and click on “Insert” which opens the following dialog, choose “Server”

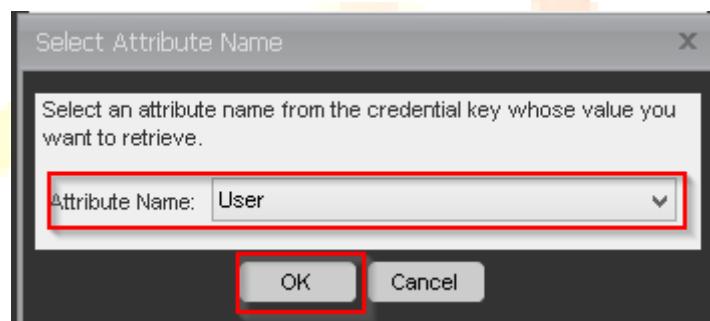


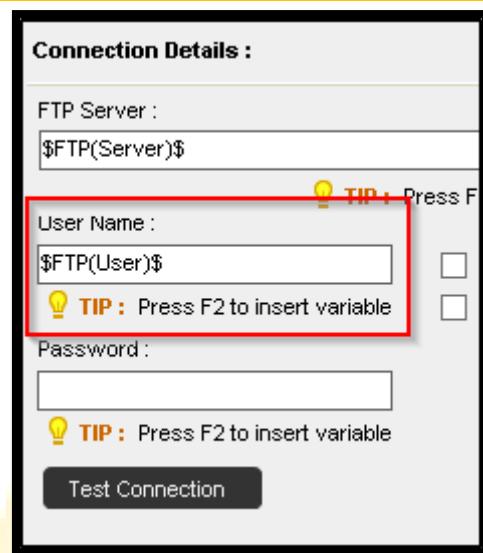


- Click inside the “User Name” text box and press F2, which opens the following dialog



- Choose credential variable “FTP” and click on “Insert” which opens the following dialog, choose “User”

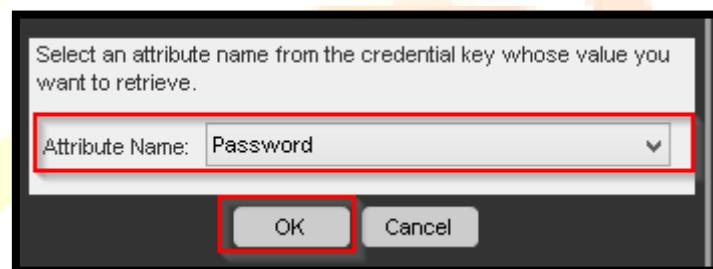


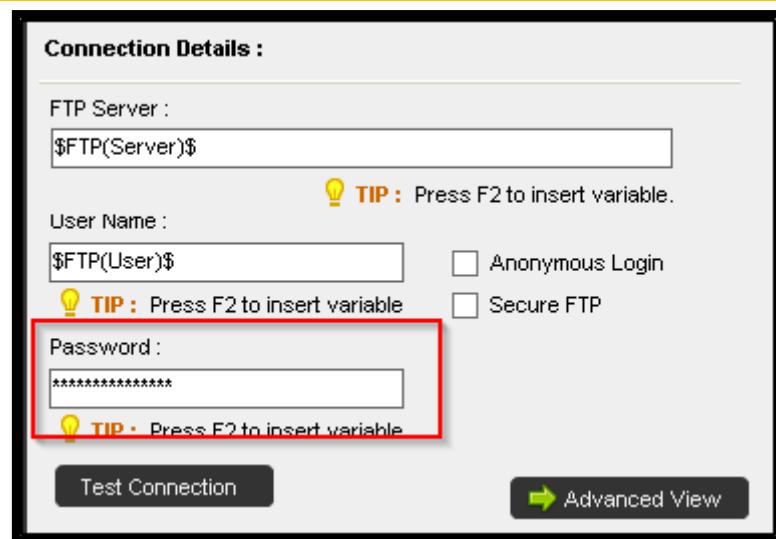


- Click inside the “User Name” text box and press F2

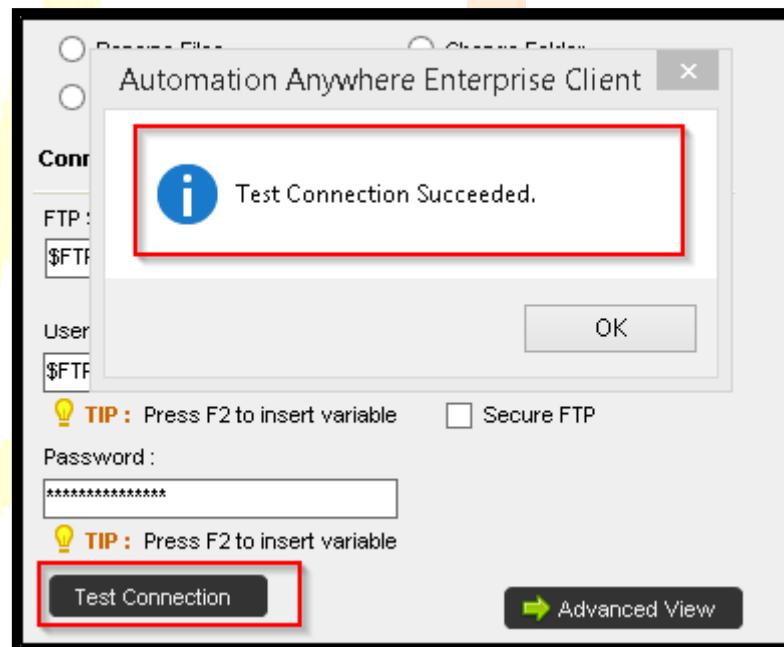


- Choose credential variable “FTP” and click on “Insert”, choose “Password”





- Click on “Test Connection” to check whether the connection has been established or not.



- Thus, a successful execution of the task.