

**Automation Anywhere Practical's**

# REST Web Service Command

*Available in Enterprise and Small Business editions 9.0 onward.*

**Change parameters of country for a REST Web Service and verify.**

1. Open REST Web Service command.
2. Enter complete URI for any web service which is up and running and you want to test.
3. Using different methods for REST service, like GET and PUT perform the necessary actions.
4. Save the response to different variables where necessary and save each command.

To enhance the test, as shown below, we have also included checkpoints wherever necessary to verify the actions that are performed.

*REST Web Service: Method: "PUT"; Request:*

*"http://192.168.2.23:9090/RestOrderService/OrderService.svc/UpdateOrder"; Save Response To:  
"\$Prompt-Assignment\$"*

*REST Web Service: Method: "GET"; Request: "10248"; Save Response To: "\$Prompt-Assignment\$"*

*Message Box: "\$Prompt-Assignment\$"*

*CheckPoint: \$Prompt-Assignment\$ Includes "US"*

*REST Web Service: Method: "PUT"; Request:*

*"http://192.168.2.23:9090/RestOrderService/OrderService.svc/UpdateOrder"; Save Response To:  
"\$Prompt-Assignment\$"*

*REST Web Service: Method: "GET"; Request: "10248"; Save Response To: "\$Prompt-Assignment\$"*

*Message Box: "\$Prompt-Assignment\$"*

*CheckPoint: \$Prompt-Assignment\$ Includes "India"*

*REST Web Service: Method: "POST"; Request:*

*"http://192.168.2.23:9090/RestOrderService/OrderService.svc/PlaceOrder"; Save Response To:  
"\$Prompt-Assignment\$"*

*Message Box: "\$Prompt-Assignment\$"*

*CheckPoint: \$Prompt-Assignment\$ Includes "true"*

*REST Web Service: Method: "GET"; Request:*

*"http://192.168.2.23:9090/RestOrderService/OrderService.svc/GetOrderDetails/1010"; Save Response  
To: "\$check\$"*

*Message Box: "\$check\$"*

*CheckPoint: \$check\$ Includes "India"*

*REST Web Service: Method: "PUT"; Request:*

*"http://192.168.2.23:9090/RestOrderService/OrderService.svc/UpdateOrder"; Save Response To:  
"\$Prompt-Assignment\$"*

*REST Web Service: Method: "GET"; Request: "10248"; Save Response To: "\$Prompt-Assignment\$"*

*Message Box: "\$Prompt-Assignment\$"*

*CheckPoint: \$Prompt-Assignment\$ Includes "US"*

## Object Cloning Command

*Available with Enterprise edition only*

**Use Case Scenario:** To search a topic, say "Introduction to Automation Anywhere" on the Automation Anywhere support site, you can utilize the 'Object Cloning' command to perform this task.

1. Click on 'New' button in the client to invoke the 'Create New Task' window.
2. Select 'Task Editor' to launch it.
3. From the Commands section, either double click or drag and drop the 'Object Cloning' command.
4. In the 'Select Window' section, from the drop down, select appropriate window; 'Help Desk of Automation Anywhere - Internet Explorer' in this case.

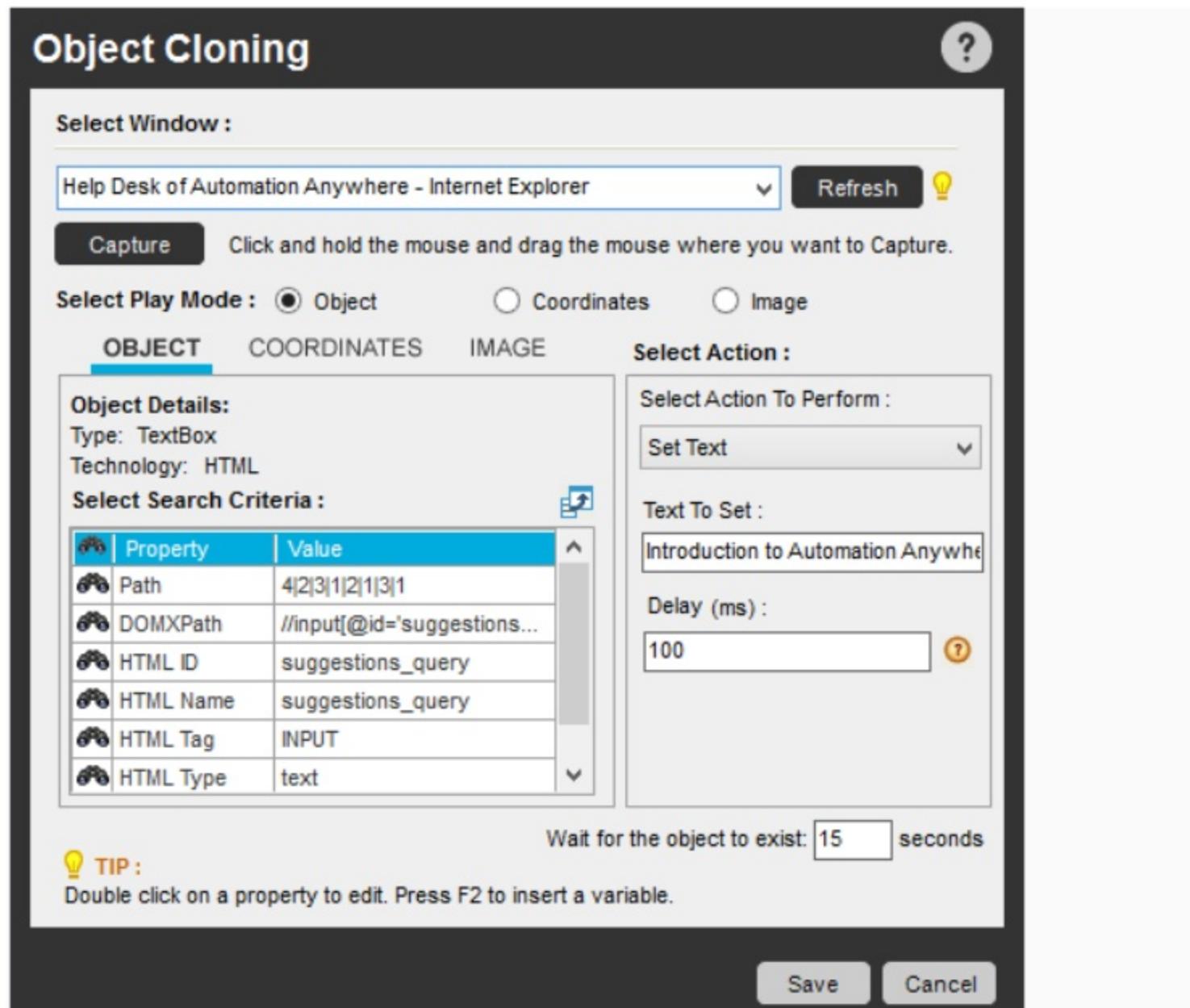
 *Note: The window has to be open so as to be visible in the list. If open, yet unavailable, click 'Refresh'.*

5. Click and hold 'Capture', and navigate the mouse to the search box in the support site that launches.

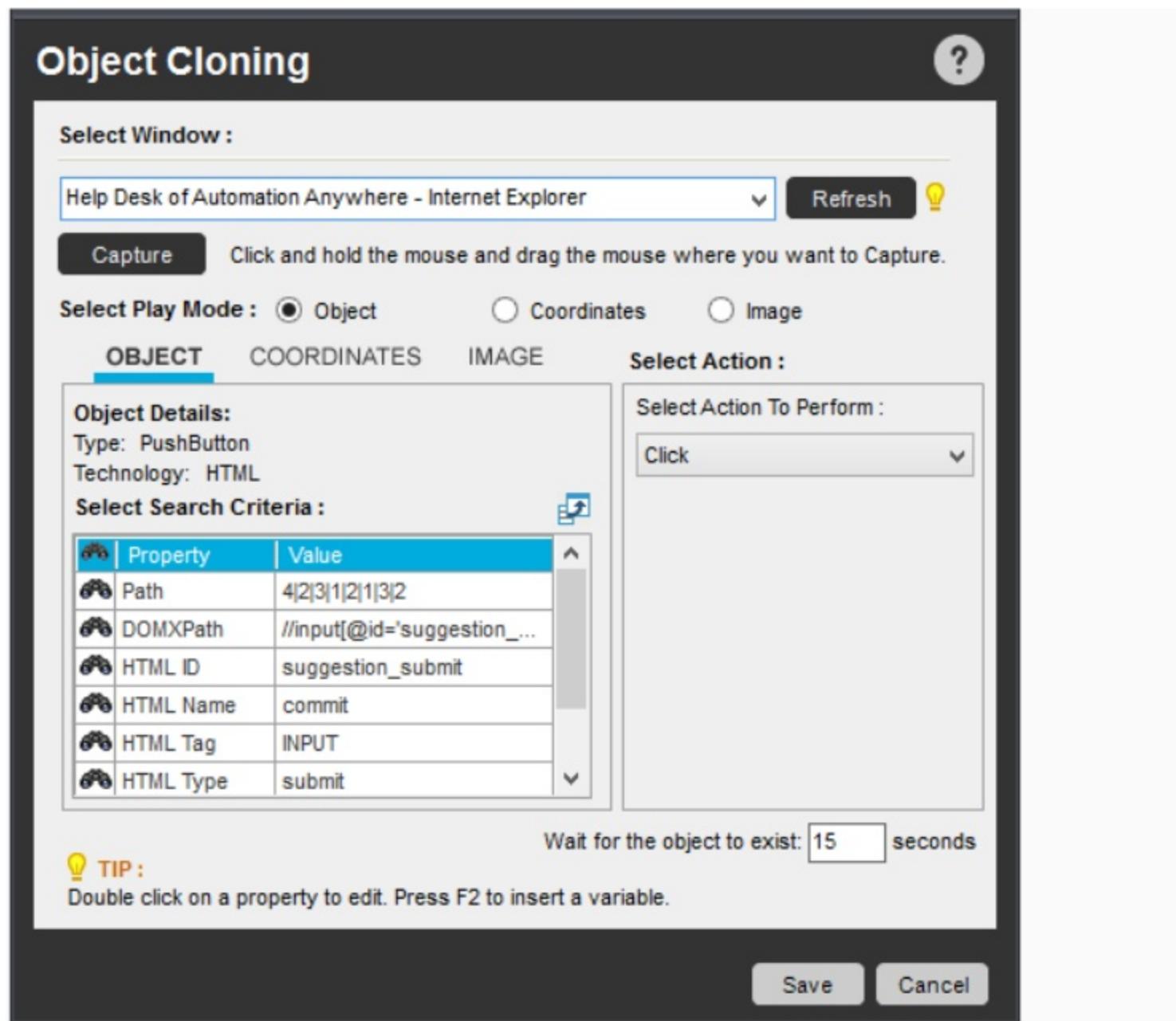
 *Note: The selected object borders will be highlighted in red.*

- The object properties are populated in the 'Object' tab of the 'Object Cloning' window.
6. From the 'Select Action To Perform' drop down list, choose 'Set Text' option.

7. Key in the search keywords; for instance "Introduction to Automation Anywhere" in the 'Text to Set' option.
8. Save the Command.



- The next step involves clicking the search button to navigate to appropriate pages.
9. Follow steps 3 and 4. Click and hold the capture button. Navigate your mouse towards 'Search' button.
  - The 'Search' button properties are populated in the 'Object' tab.
  10. In the 'Select Action' section choose 'Left Click' from the list.
  11. Save the command.



- Now you need to navigate to the appropriate topic.
12. Follow Steps 3 and 4. Click and hold the capture button. Navigate the mouse to the listed topics:

## Enter a Keyword to Find Resources

 Introduction to Automation Anywhere

Search

### Filter your search

Documentation  Examples  FAQs



#### Introduction to Automation Anywhere

Introduction to Automation Anywhere Automation Anywhere allows you to create automated tasks with ease.  
[Development Client Documentation / Getting Started with AA »](#)



#### Introduction to Process inVision

Introduction to Automation Anywhere Process inVision Automation Anywhere Process inVision allows you to the screen. Process inVision captures all the a...  
[Process inVision Documentation / Process inVision »](#)



#### Getting Started

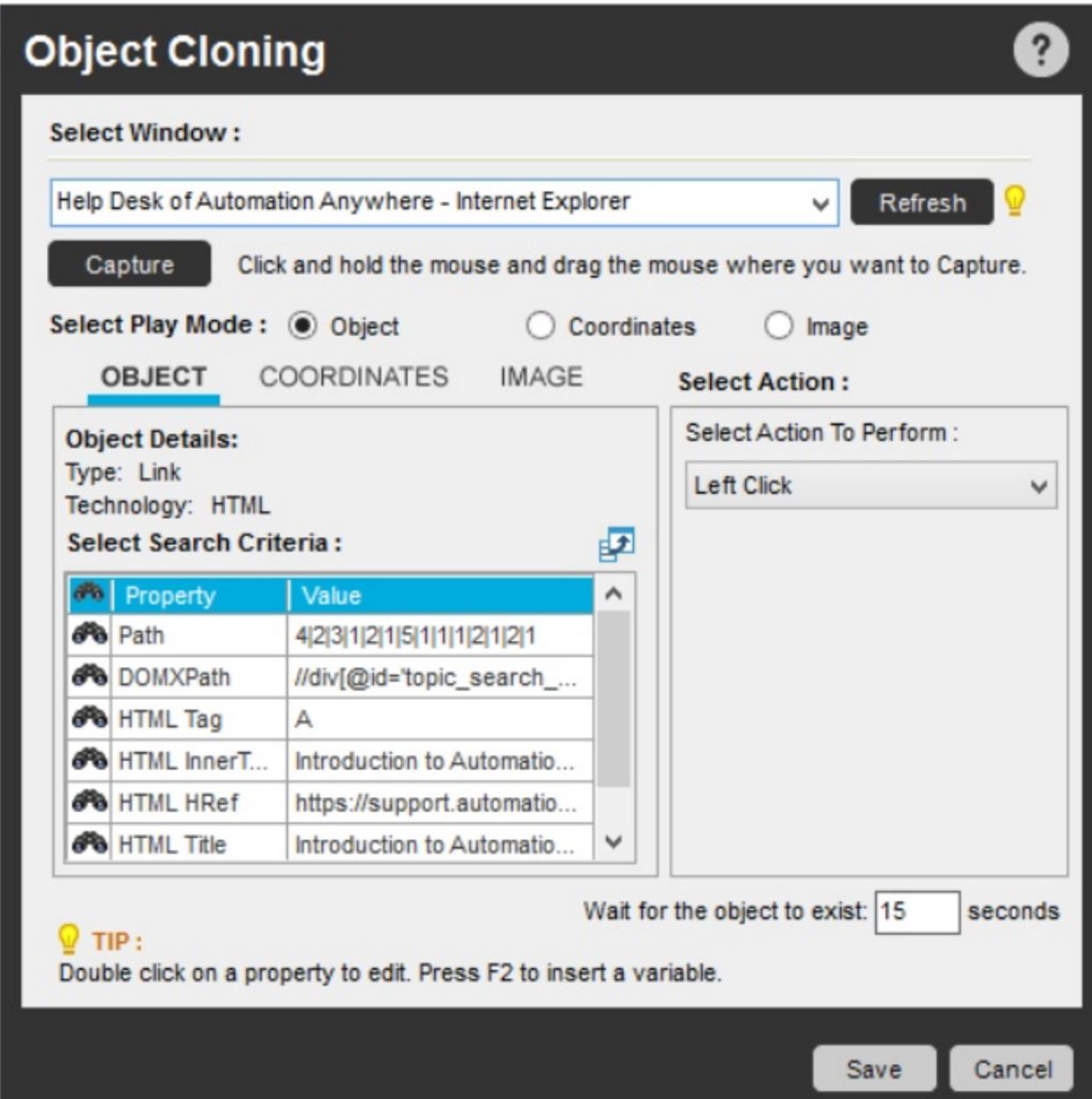
Getting Started Getting started with Automation Anywhere (AA for short!) is easy. Using the topics in this secti that are useful and expandable to larger, ...  
[Development Client Documentation / Getting Started with AA »](#)

13. Select the topic that meets your parameters.

- The selected Object properties are populated in the 'Object' tab.

14. Go to 'Select Action' and choose 'Left Click' from the list.

15. Save the command.



16. Save the task.

You are now ready to launch your search for topics on the Automation Anywhere Support site!



*Tip: Change the search parameters as and when required by editing the task.*

## XML Command

*Feature also available in Standard edition version 8.2 or higher.*

**Example 1: Get XML data by defining nodes and saving the XML data at the required location.**

1. Start XML Session using an existing file, for eg., "C:\XML\Bookstore.xml". Keep the Session as Default.

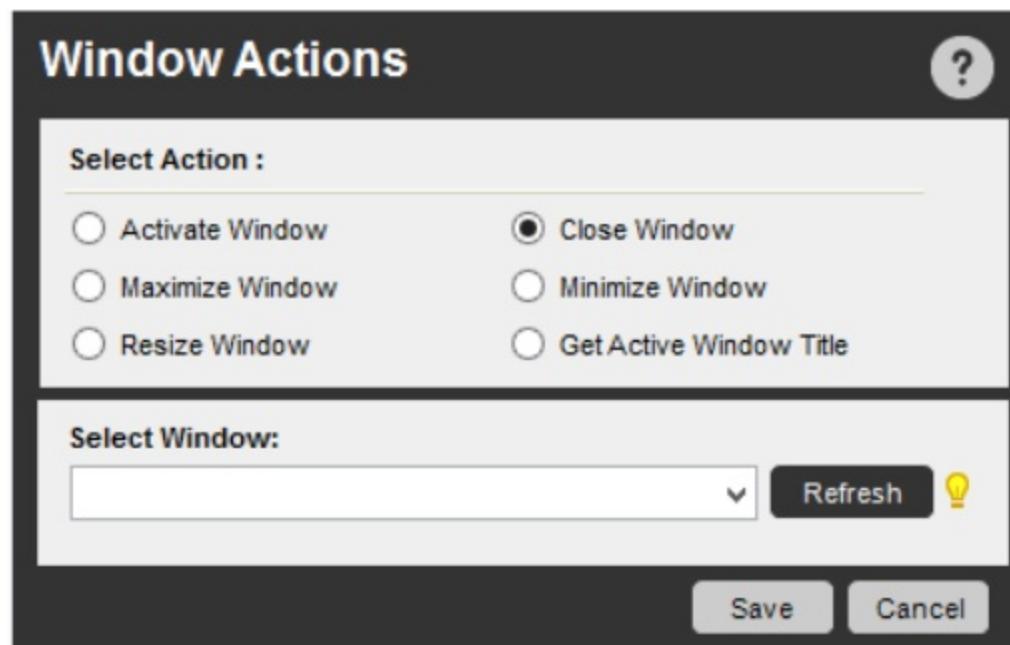
2. Update the Node value to "2010" for a XPath location "//book[3]/year".
3. Insert Node name and Node value as "newNode" and "newVal" respectively for XPath location "//book[3]/year".
4. You can execute a XPath Function "sum(//book/price)" and assign the returned value to a variable.
5. To get text value from multiple nodes given XPath expression "//book/title", go to Get Node(s) > Multiple Nodes.
6. Start Loop for 'Each node in an XML datasheet'
7. Insert a comment and prompt a message box at this juncture.
8. End Loop
9. Save the session data assigned to the variable and write the Session Data into a new file at the required location. Overwrite the file if it exists.
10. End the session by selecting End XML session

## Window Actions Command

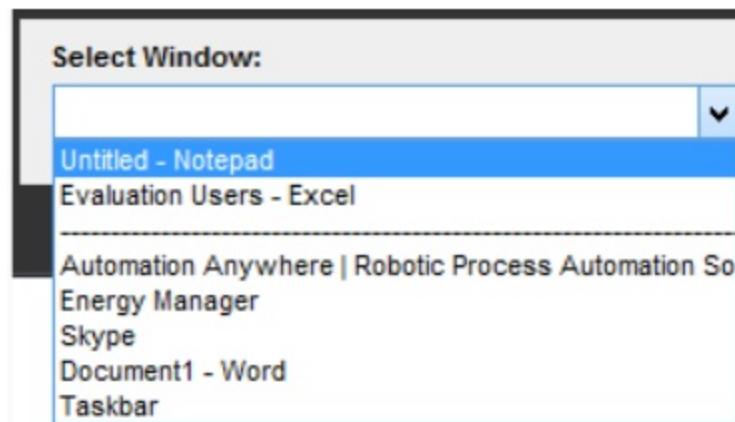
### ***Use Case Scenario 1***

Close the window 'Untitled - Notepad' in Microsoft Windows Notepad.

1. Start Microsoft Notepad. Notice that the title bar says 'Untitled - Notepad.'
2. In Automation Anywhere Client, open the Task Editor.
3. In the Task Editor, in the Commands list, double-click Window Actions. A sub-menu appears.
4. Double-click Close Window. The Window Actions dialog appears.



5. In the Select Window drop-down, select 'Untitled - Notepad.'

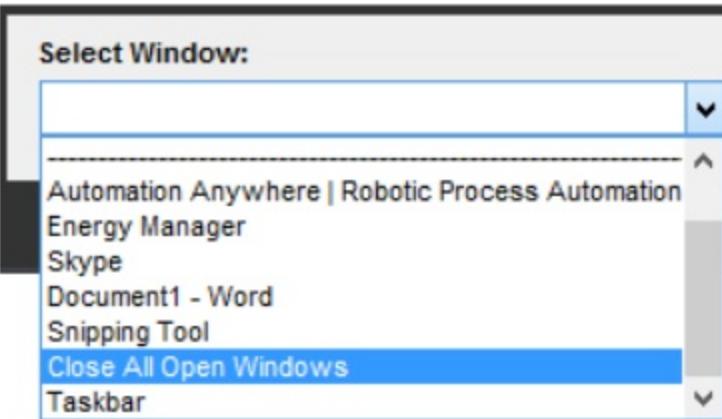


6. Click Save.

#### ***Use Case Scenario 2***

Close all open windows.

1. In Automation Anywhere Client, open the Task Editor.
2. In the Task Editor, in the Commands list, double-click Window Actions. A sub-menu appears.
3. Double-click Close Window. The Window Actions dialog appears.
4. From the Select Window drop-down, select 'Close All Open Windows.'

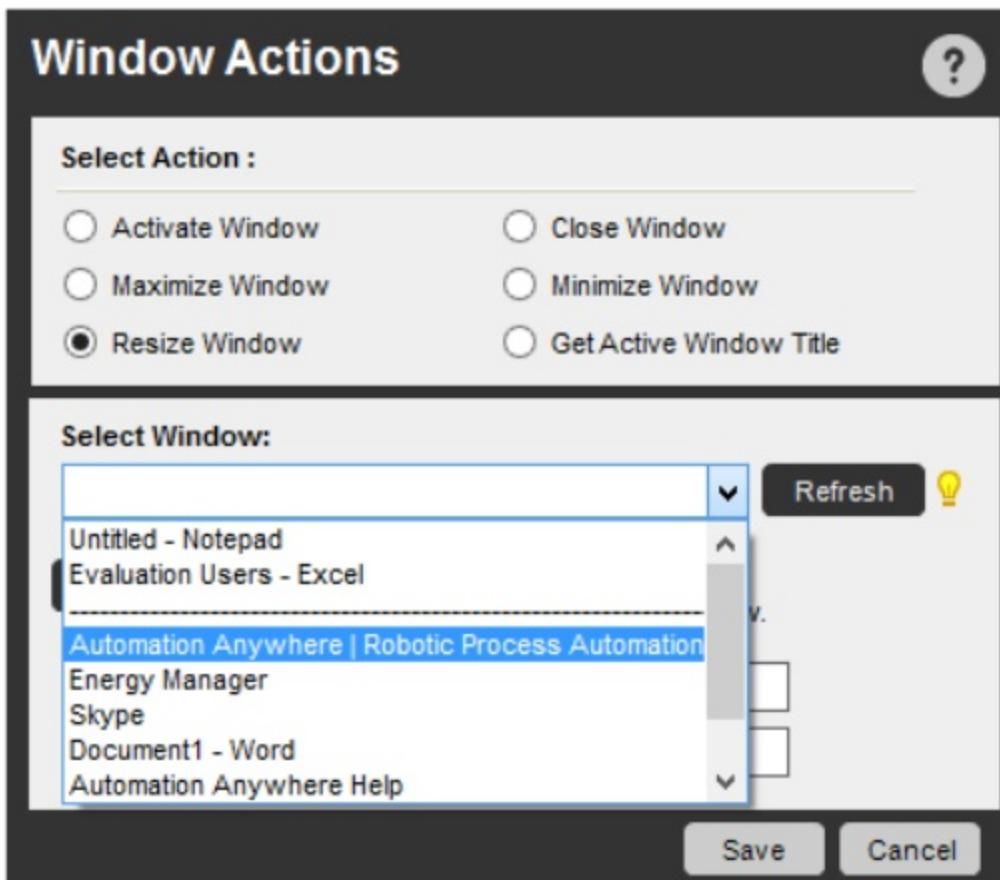


5. Click Save.

### **Use Case Scenario 3**

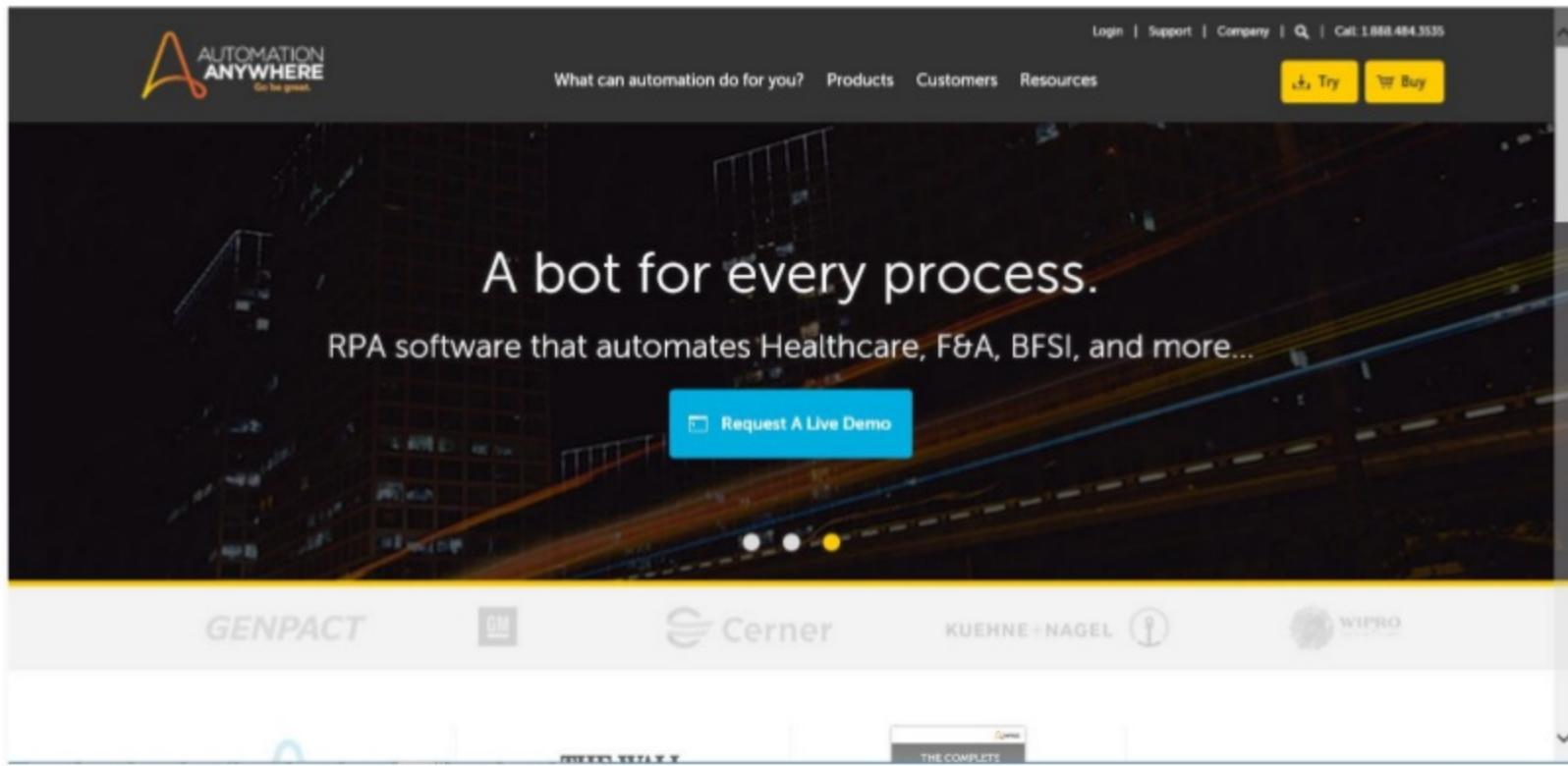
Re-size a browser window

1. Open a Web page in a browser.
2. In Automation Anywhere Client, open the Task Editor.
3. In the Task Editor, in the Commands list, double-click Window Actions. A sub-menu appears.
4. Double-click Re-size Window. The Window Actions dialog appears.

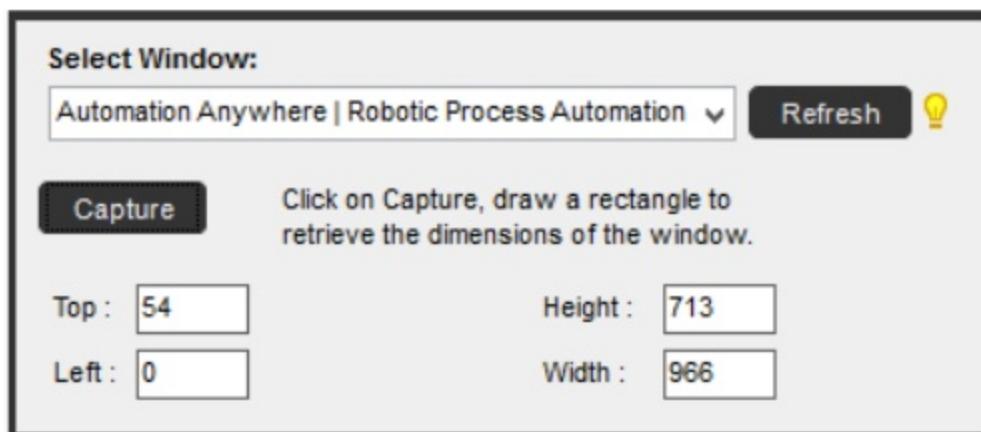


5. From the Select Window drop-down, select the browser window title that you want to re-size.

6. Click Capture, and draw a rectangle of the desired size.



7. This will retrieve the coordinates for the re-sized window.

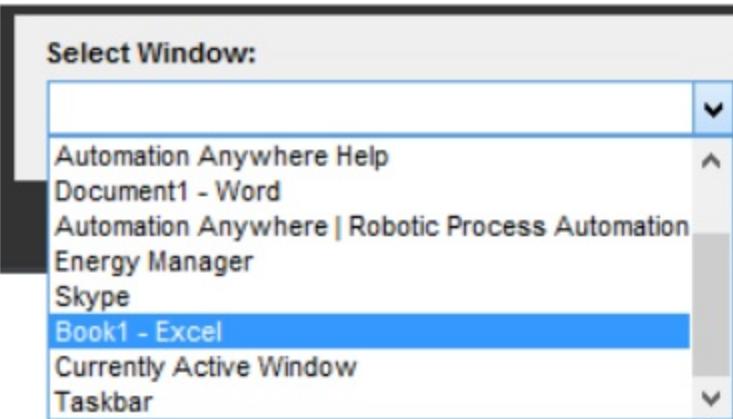


8. Click Save.

#### **Use Case Scenario 4**

Maximize a Microsoft Excel file named 'Microsoft Excel-Book1.'

1. Open the file Book1.xls in Microsoft Excel.
2. In Automation Anywhere Client, open the Task Editor.
3. In the Task Editor, in the Commands list, double-click Window Actions. A sub-menu appears.
4. Double-click Maximize Window. The Window Actions dialog appears.
5. From the Select Window drop-down, select the window titled 'Microsoft Excel-Book1.xls.'

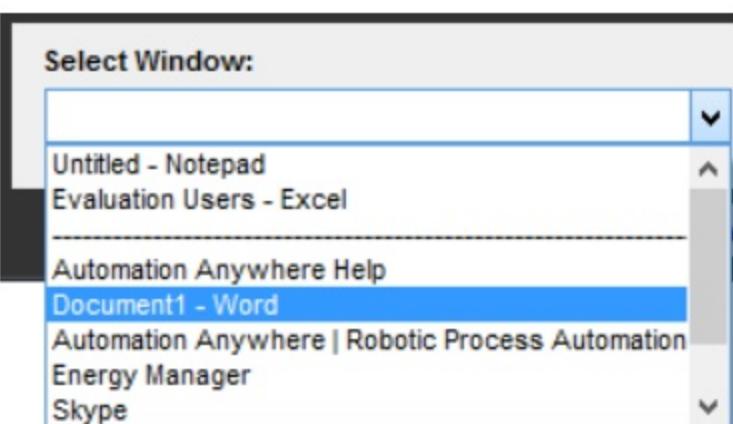


6. Click Save.

#### ***Use Case Scenario 5***

Activate a Microsoft Word document titled 'Document1.'

1. In Microsoft Word, open a new document - Word automatically names it Document 1.
2. In Automation Anywhere Client, open the Task Editor.
3. In the Task Editor, in the Commands list, double-click Window Actions. A sub-menu appears.
4. Double-click Activate Window. The Window Actions dialog appears.
5. From the Select Window drop-down, select 'Document1-Microsoft Word.'



6. Click Save.

## **SOAP Web Service Command**

*Available in Enterprise and Small Business editions only.*

#### **Example 1: Creating a Web service.**

1. Open Web Service command.

2. Enter complete URI for any web service which is up and running.
  - Example:<http://www.xignite.com/xGlobalQuotes.asmx?WSDL>
1. Click the Test Output button to view the Web Service Output data.
2. Save the data as an XML file or assign it to a variable.
3. Save and exit the Web Service command.

## Web Recorder Command

### **Example 1: Extract tabular data from a Web page.**

1. Open the Task Editor.
2. Double-click the Web Recorder command and select Extract Table.
3. Specify a URL, e.g., <http://www.yourserver.com>.
4. Launch the URL.
5. Click Capture and click the table you want to extract.
6. Specify the path of the CSV file where you want the table data to be saved when the task runs.
7. Save the command.
8. After you call the Extract Table command, Automation Anywhere Client stores all the tabular data in a system variable called `$Table Column(index)$`. You can use this variable to get the rows and columns of the table that you extracted.
9. Double-click the Loop command.
10. Select 'Each row in an Internet Explorer Table'.
11. To access each row and column in the `$Table Column$` system variable provided by Automation Anywhere Client, use 'Loop for Each row in an Internet Explorer table' to loop through all the rows of the table. To access each column, use `$Table Column(index)$`; for example, to get the first column, use `$Dataset Column(1)$`, and so on.
12. You can use Insert Keystrokes commands with the `$Table Column$` variable to type the results into the Excel document.

### **Example 2: Click a link that changes its location in Internet Explorer.**

1. Open the Task Editor.
2. Double-click the Web Recorder command, and select Open Browser.
3. Specify the URL (e.g., <http://www.yourserver.com>).
4. Save the command.
5. Double-click the Web Recorder command again and select Manage Controls.
6. Specify the URL where the link is present; in this case, the same page you specified in Open Browser.
7. Click Launch. Click Capture and leave the mouse on the link you want to click.
8. The link information will be captured.
9. Save the command.
10. Save the task.
11. When you run the task, it will open the URL and click the link you specified.

### **Example 3: Download an http file.**

1. Open the Task Editor.
2. Double-click the Web Recorder, and select Download File.
3. Specify the download file URL (e.g., <http://www.yourserver.com/myfile.exe>).
4. Save the command.
5. Specify a local filename to store on your machine.
6. Save the command.

### **Example 4: Check all links in a Web page.**

1. Open the Task Editor.
2. Double-click the Web Recorder command and select Find Broken Links.

3. Specify the URL (e.g., <http://www.yourwebpage.com>).
4. Select Test Specified Webpage Only.
5. Specify the local path for the CSV file to store the results of the command.
6. Save the command.
7. When you run the command, it will test all links in the Web page you specified, and store the results in the CSV file you named.

#### **Example 5: Check all links in a website.**

1. Open the Task Editor.
2. Double-click the Web Recorder command, and select Find Broken Links.
3. Specify the website address (e.g, <http://www.yourwebsite.com>).
4. Select Test Entire Website.
5. Specify a local file path for the CSV file to store the results of the command.
6. Save the command.
7. When you run the command, it will test all the links in the website you specified, and the results will be stored in the CSV file you named.

#### **Example 6: Always click the third link in the CNN Headlines.**

1. Click Tools > Web Recorder.
2. Type <http://www.cnn.com> and click Start
3. Click the third link, then click Stop recording.
4. Open the task in the Task Editor.
5. Edit the Click command.
6. Un-check the Name check box, and check the Index check box. Now, every time Automation Anywhere Client reaches this command it will search the link by its index (and not the name or URL it points to). Thus, even if the name of the link changes, it will click the third link.

#### **Example 7: Extract data from a website and paste it into a blank Word document.**

1. Open the Task Editor.
2. Double-click the Web Recorder command.
3. Double-click Extract Data.
4. Type a website URL and click Launch (e.g., <http://www.automationanywhere.com/products/automation-software.htm>).
5. Once the site is open, you are redirected to the Extract Data dialog box.
6. Click Capture.
7. The mouse appears as yellow background text indicating Please click on the data you want to extract.'
8. Move the mouse over the text you want to copy, and click anywhere in the text. (If the text background of the mouse changes to red, it means the text is not available for copying.)
9. Assign the copied text value to a variable from the list; e.g., \$my-variable\$.
10. Click Save.
11. Make sure a blank Word document is open.
12. Double-click Insert Keystrokes.
13. Select the window title from the list.
14. Insert the variable from the list on the right side of the dialog box, i.e., \$my-variable\$.
15. Click Save.



*Tip : You can specify keywords that determine the range of your text. For example, from the text 'Your reference no. is 245698 for today,' if you want to copy only the number and avoid the extra text, you can specify 'is' as the 'Before Keyword' and 'for' as the 'After Keyword.'*

## Wait Command

Example 1: Wait for a window titled 'Untitled - Notepad' to open.

Open the Task Editor.

Double-click the Delay/Wait command.

Select Wait For.

Select Wait For Window To Open, and type the window title: 'Untitled - Notepad'.

Type how long you want Automation Anywhere Client to wait for the condition to be true, i.e., for Notepad to open.

You can specify the task to continue or stop if the Notepad window doesn't open with the specified time.

### Using Wildcards (\*)

1. You can use wildcards (\*) in the following options: Wait For Window To Open, Wait For Window To Close, and Wait for Screen Change (relative to Window).
  - For example, if you want to perform an operation in any window with the word 'Untitled' anywhere in the title, you can insert a \* wildcard character before and after 'Untitled'. In this case, you can specify Wait For Window To Open \*Untitled\*.
2. Automation Anywhere Client will first search for the exact window named 'Untitled,' and if it does not find it, it will look for windows with 'Untitled' anywhere in the title.
3. You can insert the wildcard character (\*) at the beginning or end of the window title.

### Example 2: Wait for a submitted query to return successfully on the Web.

1. Open the Task Editor.
2. Click the Delay/Wait command.
3. Double-click Wait For Screen Change.
4. Keep the Web page open to the page where the result of the query will be displayed.
5. Select the relevant option based on whether you want to capture the image relative to the screen, or a particular window. If you select Window, you will be required to select a window from the list.
6. Click Capture and capture a rectangular selection around the area that will change when the query returns.
7. Specify 1 second in the box titled 'How long you want to wait before comparing the screen.'
8. Specify the maximum number of seconds you want the task to wait, and whether you want Automation Anywhere Client to stop the task or proceed to next command.

9. Save the command.

## Variable Operation Command

**Example 1: Use Variable Operation for an array variable for assignment and reinitialize operations.**

1. Open the Task Editor.

2. See below task with all details mentioned as comments for execution of task.

1 Comment: We use an array variable to read from existing excel file and then reuse it using data from another csv file extracted from website and then insert to database.

2 Connect to "\$ConnectStr\$" Session:'Default'

3 Comment: ArrayRows is a system defined variable used to retrieve total rows in the user defined array.

4 Comment: ArrayExl - is a user defined array variable with source as Read from Excel file . Here in our example we use the list of the books and games of the kids having title and cost, which is initialized in Variable Manager.

5 Variable Operation: \$ArrayRows(\$ArrayExl\$)\$ To \$Row\$

6 Start Loop "\$Row\$" Times

7 Comment: To skip the header title of the csv extracted from the website use below If condition

8 If \$Counter\$ Equal To (=) "1" Then

9 Continue

10 End If

11 Execute SQL Statement: 'Insert into Ebaykids (BookTitle,cost) values

("\$ArrayExl(\$Counter\$,1\$)",'\$ArrayExl(\$Counter\$,2\$)')' Session: 'Default'

12 End Loop

13 Open "<http://www.ebay.com/>"

14 Comment: Extracting the books on the philosophy into a csv file using Web Recorder

15 Set text 'philosophy books' into \_nkW in the webpage 'Electronics, Cars, Fashion, Collectibles, Coupons and More Online Shopping | eBay'

16 Click on 'Search' Command Button in the webpage 'Electronics, Cars, Fashion, Collectibles, Coupons and More Online Shopping | eBay'

17 Loop While Web Control Exists (Other HTML Control : Next ) in the webpage 'philosophy books | eBay'

18 Extract Multiple Data from Webpage to \$filePath\$ from the web page: philosophy books | eBay

19 Click on 'Next' Other HTML Control in the webpage 'philosophy books | eBay'

20 End Loop

21 Extract Multiple Data from Webpage to \$filePath\$ from the web page: philosophy books | eBay

22 Comment: Reinitialising the Array-read from excel file (ArrayExl) with the csv file extracted from the website.

23 Variable Operation: Reinitialize \$ArrayExl\$

24 Variable Operation: \$ArrayRows(\$ArrayExl\$)\$ To \$Row\$

```

25 Start Loop "$Row$" Times
26 If $Counter$ Equal To (=) "1" Then
27     Comment: To skip the header title of the csv extracted from the website
28     Continue
29 End If
30 Execute SQL Statement: 'Insert into Ebayphilosophy (BookTitle,cost) values
    ("$ArrayExl($Counter$,1)$",'$ArrayExl($Counter$,2)$')' Session: 'Default'
31 End Loop

```

**Example 2: Use different list type of variable and perform Assign and Reinitialize Operations on them. Here we are using list variable to then send emails.**

1. Open the Task Editor.
2. See below task with all details mentioned as comments for execution of task.

```

1 Comment: This Loop will execute for 5 times as List read from text file variable has 5 entries.
2 Comment: Create Variable List Type : Source = Read from text file having this RDlsttxt = 11,
21,31,41,51
3 Start Loop "List Variable $RDlsttxt$"
4 Comment: Message box will show the addition of list variable value with other variable
value. Here $row$=10
5 Variable Operation: $RDlsttxt$ + $row$ To $Prompt-Assignment$
6 Message Box: "$Prompt-Assignment$"
7 End Loop
8 Comment: Reinitialize Variable List Type : Source = Read from text file having this
RDlsttxt support@automationanywhere.com, testing@gmail.com, sales@automationanywhere.com, buynow@automationanywhere.com
9 Variable Operation: Reinitialize $RDlsttxt$
10 Start Loop "List Variable $RDlsttxt$"
11 Comment: This Loop will execute for 4 times as 'List read from text file variable has 4 entries.
12 Comment: Below command will the send emails to the address ids mentioned in the above
Read from text file -List variable.
13 Send Email: Subject "Automation Anywhere finished executing the task." with Attachment(s).
14 Message Box: "Email to '$RDlsttxt$' has been sent sucessfully."
15 End Loop
16 Comment: Create Variable List Type: Source is Value.
17 Comment: In the below Prompt command select more than one file
18 Prompt For File: "Please browse and select multiple files:" for File Assign to variable "$list_val$"
19 Variable Operation: sales@testinganywhere.com To $list_val(2)$
20 Start Loop "List Variable $list_val$"
21 Comment: This loop will execute no. of files selected in Prompt command.
22 Message Box: "$list_val$"
23 End Loop

```

### **Example 3: Use yesterdays date as a variable.**

1. Open the Variable Manager from Tools > Variable Manager.
2. Click Add new variable.
3. Set the variable type as Value, type a name for the variable (e.g., yesterday), and enter 1 as a value. (This will be the temporary value.)
4. Open the Task Editor.
5. Double-click Variable Operations.
6. On the left side of the = sign, select the variable you created from the drop down list.
7. On the right side of the = sign, press F2 and select the system variable date and click insert.
8. Type -1 after \$Date\$.
9. Click Save.

### **Example 4: Convert the value of a variable to thousands.**

1. Open the Task Editor.
2. Double-click Variable Operations.
3. For select variable, select the variable whose value you want to convert into thousands.
4. On the right side of the = sign, press F2 and insert the variable whose value you want to convert into thousands.
5. Type / 1000 after the variable.
6. Click Save.

### **Example 5: Increment the value of a variable by 1 without using \$Counter\$.**

1. Open the Task Editor.
2. Double-click Variable Operations.
3. For select variable, select the variable whose value you want to increment by 1.

4. On the right side of the = sign, press F2 and insert the variable whose value you want to increment by 1.
5. Type +1 after the variable.
6. Click Save.

## Using Web Recorder

### Example 1: Extract data from Website using pattern based data extraction.

1. Click New > Web Recorder in the main window.
2. Enter website URL for the website that you wish to extract data based on pattern . eg., an URL of a book listing site.
3. Click Start to begin recording. Please wait until Internet Explorer launches the URL completely.
4. Maximize window.
5. Click Extract Data > Pattern based data
6. Click Add to add the fields that you wish to capture from the selected web page eg., titles of books, author names, publisher, book icons, etc.
7. Click Next.
8. Check *The data spans across multiple pages* box to select similar data from subsequent pages.
9. Click Capture button to select the link that takes you to the next page.
10. Click Next.
11. Click Preview data to preview the data captured.
12. Specify the folder to save images into and specify a file ie., csv to save the extracted data into.
13. Images extracted from multiple pages will be saved in different folders.

### Example 2: Extract tabular data from a Web page.

1. Click New > Web Recorder in the main window.

2. Enter website URL <http://www.automationanywhere.com>. You can add the URL to Favourites for easy access again.
3. Click Start to begin recording. Please wait until Internet Explorer launches the URL completely.
4. Maximize window.
5. Click on Purchase > Volume Pricing in the webpage.
6. Select the [web control](#) you require in the Web recorder window that opens. Click Save.
7. Click on Extract Table button in the Web Recorder toolbar.
8. Move the yellow prompt bar provided to the table you wish to select, i.e, Feature table.



*Note: The yellow prompt bar will change to red if the content you click on is not a table.*

9. A preview of the extracted table will open for your immediate viewing. Save the file as a .csv file.
10. Use the Append to an already existing csv file option, if a CSV file has already been defined .
11. Click OK.
12. Stop recording the task.
13. Save the task.



*Note: When you run this task, it will extract the table and append it into the csv.file if specified.*

## System Command

### Example 1: Lock the computer.

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click System. A submenu appears.
3. Double-click Lock Computer. The System dialog appears. Note the green symbol indicating that the Shutdown command can help save energy and spare the environment.

4. Click Save.

### Example 2: Shut down the computer.

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click System. A submenu appears.
3. Double-click Shutdown. The System dialog appears. Note the green symbol, indicating that the Shutdown command can help save energy and spare the environment.
4. Click Save.

### Example 3: Restart the computer.

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click System. A submenu appears.
3. Double-click Restart.
4. Click Save.

 *Note: Always put System commands last in tasks, since actions entered after the System command will not be executed.*

 *Tip : You can have Automation Anywhere Client perform an auto-login, so that when your computer restarts, Automation Anywhere Client will enter your username and password. Learn more about [auto-login](#) .*

## Terminal Emulator Command -Examples

*Available in Enterprise and Small Business editions only.*

### Example 1: Obtain ipconfig of remote terminal using default prompt actions.

1. Open the Task Editor.
2. Double-click the Terminal Emulator command. Select the **Connect** option.

3. Specify the *Connection Type*, e.g., TCP/IP
4. Specify *Host Name* of the remote terminal.i.e., Admin.
5. Select the *Terminal Type* from the drop down list.
6. The *Port* is set to 23.
7. Insert the *Default Prompt command* of the terminal ie.,C:\Documents and Settings\Admin> cannot be extracted.)
8. Check *Wait For Default Prompt* if you to issue a delay. Specify the *Wait Time in milliseconds*.
9. Check the *Show Terminal Window* if you want to view the action as it happens.
10. Click Save.
11. Select the **Send Text** feature.
12. In the *Send Text* type ipconfig.You can also press F2 to insert a variable. If you wish to encrypt the text, select the *Encrypt Text* check box.
13. Select an option if *Wait for Default* is desired.
14. Click Save.
15. Select the **Get Text** feature.
16. Select *All lines* form *Get Text* option.
17. Insert a variable in *Assign Text To Variable* text box i.e., \$Clipboard\$. You can also press F2 to insert a variable. Click Save.
18. Click Save.
19. Select Message Box from the Commands list..
20. Enter a caption title for the *Message box*. You can also press F2 to insert a variable.
21. Insert variable \$Clipboard\$ in Please *enter message to show the user* text box. Specify scrollbar options for message box if desired.
22. Click Save.
23. Insert a disconnect command from the terminal selecting the **Disconnect** feature .

24. When you run this task the ipconfig of the Admin terminal will be viewed as a message on your computer screen.

## String Operation Command

### Example 1: Extract the price of a product from a product literature document.

1. Open the Task Editor.
2. Double-click the Open Program/File command.
3. Specify the location and name of the product literature document in the Program/File Path.
4. Click Launch Program/File.
5. Click Save.
6. Double-click the Insert Keystrokes command.
7. Select the window in which the document is open.
8. Specify [CTRL DOWN]a[CTRL UP][CTRL DOWN]c[CTRL UP] as the keystrokes.
9. Double-click the String Operation command.
10. Press F2 and select Clipboard variable from the list, as the source string.
11. Specify the Before keyword, i.e., the word before the product price, e.g., \$.
12. Specify the occurrence of the Before keyword in the string.
13. Specify the After keyword, i.e., the word after the product price, e.g., A.
14. Specify the occurrence of the After keyword in the string.
15. Select a variable from the list to assign the extracted price to.
16. Click Save.

### Example 2: Extract an account number from a text file.

1. Open the Task Editor.
2. Double-click the Open Program/File command.

3. Specify the location and name of the text file in Program/File Path.
4. Click Save.
5. Double-click the Insert Keystrokes command.
6. Select the window in which the text file is open.
7. Specify [CTRL DOWN]a[CTRL UP][CTRL DOWN]c[CTRL UP] as the keystrokes.
8. Double-click the String Operation command.
9. Press F2 and select Clipboard variable from the list, as the source string.
10. Specify the Before keyword, i.e., the word before the account number, e.g., No.
11. Specify the occurrence of the Before keyword in the string.
12. Specify the length of the account number in Number of characters to be extracted, e.g., 8.
13. Select a variable from the list to assign the extracted account number to.
14. Click Save.

## SNMP Command Command

### Example 1: Send a SNMP Get message to agent to access its computer information.

1. Open the Task Editor.
2. Double-click the SNMP Get command.
3. Specify the name or IP address of Agent ie., newpc or 132.18.4.12
4. Specify the Community that the Agent belongs to.
5. Specify Object ID.
6. Select the required Data Type format options, ie., Ascii or Hex.
7. Specify a path for the log file.
8. You can assign the result value to a variable.
9. In Advanced View, specify the SNMP version V1 or V2C.

10. Specify the Timeout required in milliseconds ie., 1000
11. Specify the Remote port.
12. Specify the number of retry times for the message.
13. Click Save.

#### **Example 2: Send Trap message to SNMP Manager.**

1. Open the Task Editor.
2. Double-click the Send Trap command.
3. Specify the name or IP address of Agent ie., newpc or 132.18.4.12
4. Specify the Community that the Agent belongs to.
5. Specify Object ID.
6. Select required Trap Type from the list options.
7. In Advanced View, specify the SNMP version V1 or V2C.
8. Specify the Timeout required in milliseconds .ie.,1000
9. Specify the Remote port.
10. Specify the number of retry times for the message.
11. Click Save.

## **Send Email Command**

#### **Example 1: Send an email to a list of contacts, and attach a document with details about a new product.**

1. Create a List variable for your list of contacts. You can enter all of the list values in the Variable Manager, or you can read the data from a text file.
2. Open the Task Editor.
3. Insert a Loop for List variable action from the Loop command. Select the list variable that you created in step 1.

4. Double-click the Send Email command.
5. Specify the appropriate From address.
6. In the To text box, type the list variable you created in step 1.
7. Specify the path and name of the document you want to attach.
8. Type the email message you want to send.
9. You can use the following keywords to send specific information about the task: <taskname>, <desc>, <repeat>, <status>, <lastruntime>.
10. Save the command.
11. Place the Send Email command within the Start and End loop.
12. When you run the task, it loops through the list and sends an individual email to each contact, with the attached document.

## Service Command

**Example 1: Start a user-defined or Windows service, if it is not already running, before running an application that requires the service to be running.**

1. Open the Task Editor.
2. Double-click the Get Service Status command.
3. Select the service from the list of available installed services.
4. Select the variable from the list to assign the value to.
5. Click Save.
6. Insert an IF condition that checks whether the value of the variable selected in step 4 is not equal to Started. In the IF command, select the option Variable. Click Edit. Type the variable in the first text box, select the operator Not equal To (<>), and in the Fix option, type Started.
7. Double-click the Start Service command and insert the same actions within the If-Endif commands.
8. Select the service from the list of available installed services.
9. Click Save.

10. Double-click the Open Program/File command.
11. Specify the name of the application you want to run.
12. Click Save.

#### **Example 2: Pause a service before performing an action, then resume it.**

1. Open the Task Editor.
2. Double-click the Pause Service command.
3. Select the service from the list of available installed services.
4. Click Save.
5. Insert the commands for the actions you want to perform.
6. Double-click the Resume Service command.
7. Select the service from the list of available installed services.
8. Click Save.

## **Screen Capture Command**

#### **Example 1: Capture a screen of an active application.**

1. Open the application that you wish to capture.
2. Open the Task Editor in Automation Anywhere Client.
3. Double-click the Screen Capture command.
4. Choose 'Capture Window' option.
5. Select the application window from the drop-down list.
6. Provide the path where you want to save the captured image of the application screen.
7. Click on 'Overwrite File' to overwrite the output if desired.
8. Click on Save.
9. Save the task.

# SAP-Integration Command

*Available with SAP License for Enterprise edition only.*

**Example 1: To Create Sales Order in SAP using the related BAPIs and SAP Integration command actions.**

1. Open the Task Editor.
2. See the below example with detailed steps and Comments explain the task steps.

```
1 Log to File: >> 10 Sales Order Creation Test with Individual command options << in  
"d:\saptest.txt"  
2 Log to File: >> Start Connect in "d:\saptest.txt"  
3 Comment: -----Connect to SAP system -----  
4 SAP Integration: Connect to SAP system 'm2bsrv04v1' as user 'STUDENT001'; Session: 'sales'  
5 Log to File: >> End Connect in "d:\saptest.txt"  
6 Log to File: >> Start Create in "d:\saptest.txt"  
7 Comment: ----- Initialise BAPI 'BAPI_SALESORDER_CREATEFROMDAT2' -----  
-----  
8 SAP Integration: Create BAPI Function 'createso'; BAPI: 'BAPI_SALESORDER_CREATEFROMDAT2';  
Session: 'sales'  
9 Log to File: >> End Create in "d:\saptest.txt"  
10 Log to File: >> Start Get in "d:\saptest.txt"  
11 Comment: ----- Initialise BAPI 'BAPI_TRANSACTION_COMMIT' -----  
12 SAP Integration: Create BAPI Function 'commit'; BAPI: 'BAPI_TRANSACTION_COMMIT'; Session:  
'sales'  
13 Comment: ----- Get 'ORDER_HEADER_IN' Structure in memory -----  
14 SAP Integration: Get Structure 'ORDER_HEADER_IN' as 'header' to memory from function  
'createso'; Session: 'sales'  
15 Comment: ----- Get 'ORDER_HEADER_INX' Structure in memory -----  
16 SAP Integration: Get Structure 'ORDER_HEADER_INX' as 'headerx' to memory from function  
'createso'; Session: 'sales'  
17 Comment: ----- Get 'ORDER_ITEMS_IN' Table in memory -----  
18 SAP Integration: Get Table 'ORDER_ITEMS_IN' as 'items' to memory from function 'createso';  
Session: 'sales';  
19 Comment: ----- Get 'ORDER_ITEMS_INX' Table in memory -----  
20 SAP Integration: Get Table 'ORDER_ITEMS_INX' as 'itemsx' to memory from function 'createso';  
Session: 'sales';  
21 Comment: ----- Get 'ORDER_SCHEDULES_IN' Table in memory -----  
22 SAP Integration: Get Table 'ORDER_SCHEDULES_IN' as 'schedules' to memory from function  
'createso'; Session: 'sales';  
23 Comment: ----- Get 'ORDER_SCHEDULES_INX' Table in memory -----  
24 SAP Integration: Get Table 'ORDER_SCHEDULES_INX' as 'schedulesx' to memory from function
```

```
'createso'; Session: 'sales';
25 Comment: ----- Get 'ORDER_PARTNERS' Table in memory -----
26 SAP Integration: Get Table 'ORDER_PARTNERS' as 'partners' to memory from function 'createso';
Session: 'sales';
27 Comment: ----- Get 'RETURN' Table to CSV -----
28 SAP Integration: Get Table 'RETURN' as 'return' to memory from function 'createso'; Session:
'sales';
29 Log to File: >> End Get in "d:\saptest.txt"
30 Comment: Start Sales Order Creation
31 Start Loop "10" Times
32 Comment: ----- Set 'ORDER_HEADER_IN' Fields -----
33 SAP Integration: Set Field 'DOC_TYPE' in structure 'header' in function 'createso'; Session:
'sales'
34 SAP Integration: Set Field 'SALES_ORG' in structure 'header' in function 'createso'; Session:
'sales'
35 SAP Integration: Set Field 'DISTR_CHAN' in structure 'header' in function 'createso'; Session:
'sales'
36 SAP Integration: Set Field 'DIVISION' in structure 'header' in function 'createso'; Session: 'sales'
37 Comment: ----- Set 'ORDER_HEADER_INX' Fields -----
38 SAP Integration: Set Field 'UPDATEFLAG' in structure 'headerx' in function 'createso'; Session:
'sales'
39 SAP Integration: Set Field 'DOC_TYPE' in structure 'headerx' in function 'createso'; Session:
'sales'
40 SAP Integration: Set Field 'SALES_ORG' in structure 'headerx' in function 'createso'; Session:
'sales'
41 SAP Integration: Set Field 'DISTR_CHAN' in structure 'headerx' in function 'createso'; Session:
'sales'
42 SAP Integration: Set Field 'DIVISION' in structure 'headerx' in function 'createso'; Session:
'sales'
43 Comment: ----- Set 'ORDER_ITEM_IN' Fields -----
44 SAP Integration: Set Field 'ITM_NUMBER' in table 'items' in function 'createso'; Session: 'sales'
45 SAP Integration: Set Field 'MATERIAL' in table 'items' in function 'createso'; Session: 'sales'
46 SAP Integration: Set Field 'ITM_NUMBER' in table 'items' in function 'createso'; Session: 'sales'
47 SAP Integration: Set Field 'MATERIAL' in table 'items' in function 'createso'; Session: 'sales'
48 Comment: ----- Set 'ORDER_ITEM_INX' Fields -----
49 SAP Integration: Set Field 'UPDATEFLAG' in table 'itemsx' in function 'createso'; Session: 'sales'
50 SAP Integration: Set Field 'ITM_NUMBER' in table 'itemsx' in function 'createso'; Session:
'sales'
51 SAP Integration: Set Field 'MATERIAL' in table 'itemsx' in function 'createso'; Session: 'sales'
52 Comment: ----- Set 'ORDER_PARTNERS' Fields -----
53 SAP Integration: Set Field 'PARTN_ROLE' in table 'partners' in function 'createso'; Session:
'sales'
54 SAP Integration: Set Field 'PARTN_NUMB' in table 'partners' in function 'createso'; Session:
'sales'
55 Comment: ----- Set 'ORDER_SCHEDULES_IN' Fields -----
56 SAP Integration: Set Field 'ITM_NUMBER' in table 'schedules' in function 'createso'; Session:
```

```

'sales'
57 SAP Integration: Set Field 'SCHED_LINE' in table 'schedules' in function 'createso'; Session: 'sales'
58 SAP Integration: Set Field 'REQ_QTY' in table 'schedules' in function 'createso'; Session: 'sales'
59 SAP Integration: Set Field 'ITM_NUMBER' in table 'schedules' in function 'createso'; Session: 'sales'
60 SAP Integration: Set Field 'SCHED_LINE' in table 'schedules' in function 'createso'; Session: 'sales'
61 SAP Integration: Set Field 'REQ_QTY' in table 'schedules' in function 'createso'; Session: 'sales'
62 Comment: ----- Set 'ORDER_SCHEDULES_INX' Fields -----
63 SAP Integration: Set Field 'UPDATEFLAG' in table 'schedulesx' in function 'createso'; Session: 'sales'
64 SAP Integration: Set Field 'ITM_NUMBER' in table 'schedulesx' in function 'createso'; Session: 'sales'
65 SAP Integration: Set Field 'SCHED_LINE' in table 'schedulesx' in function 'createso'; Session: 'sales'
66 SAP Integration: Set Field 'REQ_QTY' in table 'schedulesx' in function 'createso'; Session: 'sales'
67 Comment: ----- Execute BAPI 'BAPI_SALESORDER_CREATEFROMDAT2' -----
-----
68 SAP Integration: Invoke BAPI Function 'createso'; Session: 'sales'
69 Comment: ----- Get Status Message -----
70 SAP Integration: Get Field 'SALESDOCUMENT' to variable 'SalesOrderNumber' from function 'createso'; Session: 'sales'
71 Comment: ----- Get Return Table -----
72 SAP Integration: Get Table 'RETURN' to csv from function 'createso'; Session: 'sales';
73 Comment: ----- Log to file -----
74 Log to File: >> Sales Order '$SalesOrderNumber$' is created successfully. in "d:\saptest.txt"
75 End Loop
76 Comment: ----- Execute 'BAPI_TRANSACTION_COMMIT' -----
77 SAP Integration: Invoke BAPI Function 'commit'; Session: 'sales'
78 Open: "d:\saptest.txt "

```

### **Example 2:**

1. Open the Task Editor.
2. See the below example with detailed steps and Comments explain the task steps using only Create function, Get/Set Data and Invoke function options.

```

1 SAP Integration: Connect to SAP system 'm2bsrv04v1' as user 'STUDENT001'; Session: 'sales'
2 SAP Integration: Create BAPI Function 'CompanyList1'; BAPI: 'BAPI_COMPANY_GETDETAIL';
Session: 'Default'
3 SAP Integration: Set Field 'COMPANY_ID' in function 'CompanyList1'; Session: 'Default'
4 SAP Integration: Invoke BAPI Function 'CompanyList1'; Session: 'Default'

```

5 SAP Integration: Get Table 'COMPANY\_DETAIL' to array variable '' from function 'CompanyList1';  
Session: 'Default'

## Run Task Command

### Example 1: Create a workflow (Task Chain) that calls several smaller tasks.

1. Open the Task Editor.
2. Double-click the Run Task command.
3. Click Browse, and select the path to notepad.atmx.
4. You can select repeat and speed properties for the task.
5. Click Save.
6. Insert Run Task commands to call the remaining tasks.

 *Note: You can run as many tasks as you want from a single task. You can also chain tasks, e.g. call Task2 from Task1, and call Task3 from Task2, etc. You can even call tasks conditionally; for example you can call the Run Task command within an IF condition.*

## Run Script Command

### Example 1: Summation of all Variables using Java script.

1. Create and save a Java Script to sum all variables.
2. Create user defined variables using variable manager. For example:  
Variable as 0 and variable1, variable2, variable3, variable4, variable5 as NULL.
3. Now Using Variable operation command  

```
Assign Variable1 = Variable + 1
Variable2 = Variable1 + 1
Variable3 = Variable2 + 1
Variable4 = Variable3 + 1
Variable5 = Variable4 + 1
```
4. Click Run script command.
5. Specify the saved Java Script in Insert Script.

6. Insert assigned variables in parameters field. Insert a variable Return value field.
7. Drag and Drop a message box command, Using F2 support insert the variable which will return the sum.
8. Save the task.
9. Result: When you run this task, it will run the script and return a message box with summation of all variables.

**Example 2: Loop through all files in a folder, and pass each filename as a parameter to a VBScript that checks the size of the file and prompts the user.**

1. Double-click the Loop command.
2. Select 'Loop for each file in a folder.'
3. Specify the path of the folder from which you want to pass all the files as a parameter to the VBScript.
4. Call the VBScript within the loop and pass the following system variables as a parameter to the script: \$CurrentDirectory\$, \$FileName\$.Extention\$.
5. Save the task.

**Example 3: Create Run time database to insert records into database (.mdb) file from an Excel operation or a Database task.**

1. Create and save a script in VBScript to create a database (.mdb) file.
  2. Edit the required Excel operation or Database task.
  3. Select the task action where you wish to create a database(.mdb) file.
  4. Click Run script command.
  5. Specify the saved VBscript in Select Script. Click Save.
  6. You can then specify Loop and Database operations to read required Excel data and insert the data into the database file(.mdb)
  7. Save the task.
- When you run this task, it will run the script and create a database file(.mdb) after which data will be inserted from Excel into the database(.mdb) file.

FILTERS  Mouse Moves  Keystrokes  Mouse Clicks  Delays  Other Windows All Find Te

```

1 Comment: =====Task for Database and Excel Automation=====
2 Comment:
3 Comment: =====Create Database and Table =====
4 IF If File Does Not Exist ("C:\DATA\Product.mdb") Then
5   Run Script "C:\DATA\Create Database.vbs"
6 End If
7 Comment: Read Data from Excel.
8 Excel: Open Spreadsheet "C:\DATA\Data.xls". ActiveSheet: "Default". Session: Default
9 Excel: Get multiple cells from cell "A2" to cell "D7" Session: Default
10 Start Loop "Each row in an Excel dataset of Session: Default"
11 Comment: =====Insert Data into Database=====
12 $ Variable Operation: $Excel Column(1$) To $Var1$
13 $ Variable Operation: $Excel Column(2$) To $Var2$
14 $ Variable Operation: $Excel Column(3$) To $Var3$
15 $ Variable Operation: $Excel Column(4$) To $Var4$
16 $ Variable Operation: $Excel Column(5$) To $Var5$
17 Connect to "Provider=MSDASQL.1;Persist Security Info=False;Extended Properties=C:\DATA\Product.mdb"
18 Execute SQL Statement: "Insert into Product Values $Var1$ $Var2$ $Var3$ $Var4$ $Var5$" Session:'Default'
19 Disconnect from database Session:'Default'
20 Delay: (50 ms)
21 End Loop

```

```

dim provider : provider = "microsoft.jet.oledb.4.0"
dim db : db = "c:\data\Product.mdb"
dim ds : ds = "provider=" & provider & "; data source=" & db
dim catalog : set catalog = createobject("adox.catalog")

catalog.create ds

const adInteger = 3 'Integer
const adVarChar = 202 'Variable Character
dim new_table : set new_table = createobject("adox.table")

new_table.Name = "Product"
new_table.columns.append "F_Name", adVarChar, 30
new_table.columns.append "L_Name", adVarChar, 30
new_table.columns.append "P_Name", adVarChar, 30
new_table.columns.append "Subject", adVarChar, 30
new_table.columns.append "Vnotes", adVarChar, 30

catalog.Tables.Append new_table
set new_table = nothing
set catalog = Nothing

```

#### **Example 4: Add data into an XML file from an Excel operation.**

1. Create and save a script in VBScript to add data into XML file.
2. Edit the required Excel operation.
3. Specify Loop and Database operations to get required Excel data.
4. Select the task action where you wish to add Excel data into XML file.
5. Click Run script command.
6. Specify the saved VBscript in Select Script.
7. Insert variables in Parameters.
8. Click Save.
9. Save the task.
- When you run this task, it will run the script and add Excel data into the specified XML file.

The screenshot shows the Task Editor interface with a script history on the left and a code editor on the right.

**Script History (Left):**

```

1 Comment: ======XML and Excel Operation=====
2 Comment:
3 Comment: ======Get Data from Excel ======
4 Excel: Open Spreadsheet "C:\DATA\DataSet.xls", ActiveSheet: "Default", Session: Default
5 Excel: Get multiple cells from cell "A2" to cell "D7" Session: Default
6 Start Loop "Each row in an Excel dataset of Session: Default"
7 $ Variable Operation: $Excel Column(1)$ To $Var1$
8 $ Variable Operation: $Excel Column(2)$ To $Var2$
9 $ Variable Operation: $Excel Column(3)$ To $Var3$
10 $ Variable Operation: $Excel Column(4)$ To $Var4$
11 $ Variable Operation: $Excel Column(5)$ To $Var5$
12 Comment: ======Run Script Add Data in XML=====
13 Run Script "C:\DATA\Add Data in XML File.vbs $Var1$ $Var2$ $Var3$ $Var4$ $Var5$"
14 Delay: (50 ms)
15 End Loop

```

**Code Editor (Right):**

```

Set objXML, binFileExist
Set objXMLv
Dim intID

Set objXML = CreateObject("Microsoft.XMLDOM")
objXML.async = False

binFileExist = objXML.load("C:\DATA\Config.xml")
If binFileExist = False Then
    objXML.appendChild(objXML.createProcessingInstruction("xml", "version=""1.0"""))
    objXML.appendChild(objXML.createElement("Guests"))
    intID = 1
Else
    intID = objXML.documentElement.childNodes(
        objXML.documentElement.childNodes.length - 1).childNodes(0).text + 1
End If

Set objXMLv = objXML.createElement("visitor")

objXMLv.appendChild(objXML.createElement("ID"))
objXMLv.appendChild(objXML.createElement("FirstName"))
objXMLv.appendChild(objXML.createElement("LastName"))
objXMLv.appendChild(objXML.createElement("Name"))
objXMLv.appendChild(objXML.createElement("Subject"))
objXMLv.appendChild(objXML.createElement("Note"))

objXMLv.childNodes(0).text = intID
objXMLv.childNodes(1).text = WScript.Arguments.Item(0)
objXMLv.childNodes(2).text = WScript.Arguments.Item(1)
objXMLv.childNodes(3).text = WScript.Arguments.Item(2)
objXMLv.childNodes(4).text = WScript.Arguments.Item(3)
objXMLv.childNodes(5).text = WScript.Arguments.Item(4)

objXML.documentElement.appendChild(objXMLv.cloneNode(True))
objXML.save("C:\DATA\Config.xml")

Set objXMLv = Nothing
Set objXML = Nothing

```

## Read CSV\_Text Command

**Example 1:** Read values from a tab-delimited CSV file and transfer the values to another application.

1. Open the Task Editor.
2. Double-click the Read from CSV/Text command.
3. Specify the name of the CSV file you want to read.
4. Select Tab as the delimiter, and check the Header check box if your CSV file has headers.
5. Click Save.
6. To access the values returned by the command, use the \$Filedata Column\$ system variable. Use the 'Loop for Each row in a CSV/Text file' option to loop through the rows of the result. To access each column, use \$Filedata Column(index)\$; for example, to get the first column, use \$Filedata Column(1)\$.
7. Within the loop, insert Manage Windows Controls commands to set the value of the controls in the target application.

**Example 2:** Getting values under a specified Column Title.

1. In the Task Editor double click 'Read from CSV/Text' command.

2. Click  and select the CSV File that hypothetically contains table with the heading titles in the first row (refer image):
3. Tick the Contains Header check-box.
4. Save the command.
5. Double click on Loop > Each Row in A CSV/Text File Command.
6. Save the command; it should then populate the Task Editor with the Start Loop, Comment and End Loop lines.
7. Inside the loop, double click the Message Box Command
8. Press F2 and select the \$Filedata Column\$ variable
  - A dialog box will pop-up prompting you to enter the Column Header
10. Enter Name in that text-box and click OK
  - You will notice that the Message Box Command will display \$Filedata Column(Name)\$
11. Save the Command. Save and Run the task; you will get the values under the Name Column one by one.

## Registry Command

**Example 1: Create a new registry key, before creating a key check out whether it is already created or not.**

1. Open the Test Editor.
2. Insert Checkpoint - Registry, select "Registry key exists" option and provide a registry key to check if the registry key exists or not.
3. Double-click the Registry command.
4. Enter the registry key to be created.
5. Enter a value name of the key to be created.
6. Select a type of a value.
7. Enter appropriate data for the value according to the selected value data base.

8. Click Save.

9. Save the test.

### **Example 2: Get data of a value from registry**

1. Open the Test Editor.

2. Double-click the Get Value Data command.

3. Specify the registry key for which data needs to be fetched.

4. Assign the value to a variable from the list.

5. Click Save.

6. Save the test.

### **Example 3: Export the registry**

1. Open the Test Editor.

2. Double-click the Export command.

3. Select the export range.

4. Provide the path where you want to save the exported registry file.

5. Click Save.

6. Save the test.

## **Prompt Command**

### **Example 1: Prompt the user for Start and End dates and generate a sales report for the date range.**

1. Open the Task Editor.

2. Double-click the Prompt command and double-click Prompt For Value.

3. Select Don't Write In Any Window from the drop-down list.

4. Type a message to prompt the user for the Start date.

5. Assign the value the user enters to a variable, by selecting the variable from the variable list.
6. Similarly, assign the End date entered by the user to a variable.
7. You can use the two variables in place of start and end dates in the task that generates the sales report.
8. When the task runs, the two variables will be assigned the values entered by the user.

### **Example 2: Create a task that prompts the user for data and automates the installation of an application.**

1. Click Record in Automation Anywhere Client.
2. Perform the steps to install the application.
3. If you plan to run the task on different computers with different resolutions, we suggest using the keyboard instead of the mouse while recording the installation process.
4. After you finish recording and save the task, select it in the Task List and click Edit.
5. Review your keystroke actions, such as the actions where you specified an installation folder, etc.
6. Right-click an action and select Convert to Prompt (or select it and click the Actions button).
7. This converts your recorded keystrokes to a Prompt command that prompts the user when the task runs.

In this manner, you can create a generic task to automate the application and not depend on the values you entered while recording.

### **Example 3: Prompt for a file that will be used in a loop to enter data.**

1. Open the Task Editor.
2. Double-click the Prompt command and double-click Prompt for File.
3. Type the caption text.
4. Enter a custom message for the user to specify the file.
5. Assign the value to a list variable from the list.
6. Expand the Loop command and double-click the list.

7. Select the list variable you assigned to the file.
8. Double-click the Insert Keystrokes command.
9. Select the window title from the list of files where the data will be entered.
10. Type the data you want to insert into the file.
11. Click Save.

 *Note: This is a procedure for a single file. You can do the same for multiple files; you will have to check 'Enable this task to run with similar window titles/files' under the General Tab, in the Properties menu.*

#### **Example 4: Prompt for the folder that contains a setup file (setup.exe).**

1. Open the Task Editor.
2. Double click the Prompt command and double-click Prompt for Folder.
3. Type a caption.
4. Type a message for the user to select the folder.
5. Assign the value to a value variable from the list.
6. Click Save.
7. Double-click the Open Program/File command.
8. In the Program/File path, type the variable to which you assigned the value of the folder, followed by '\setup.exe'.
9. Click Save.

#### **Example 5: Prompt the user for confirmation before deleting a record from a database.**

1. Open the Task Editor.
2. Insert the command to connect to a database.
3. Double-click the Prompt command and double-click Prompt For Yes/No.
4. Type a message for the user to select the folder.

5. Type a caption.
6. Type a custom message asking the user to select the desired option (Yes/No).
7. Assign the value to a value variable from the list.
8. Click Save.
9. Double-click the Loop command and double-click Condition.
10. In the If dialog, choose File exists.
11. Insert an If - Variable condition to check whether the variable to which the value assigned has the value Yes.
12. Click Save.
13. Double-click the Database command and double-click Insert/Update/Delete.
14. In the Database dialog, in the Enter Insert/Update/Delete Statement text box, type the delete statement to delete a record from a particular table.
15. Click Save.
16. In the Program/File path, type the variable to which the value of the folder was assigned, followed by \setup.exe.  
Double-click the Database command and double-click Disconnect.
17. Click Save.

## Printer Settings Command

### **Example 1: Change the default printer.**

1. Open the Task Editor.
2. Double-click the Set Default Printer command.
3. Select a printer from the list of installed printers.
4. Click Save.

### **Example 2: Remove a printer.**

1. Open the Task Editor.

2. Double-click the Remove Printer command.
3. Select a printer from the list of installed printers.
4. Click Run script command.
5. Click Save.

## Play Sound Command

### Example 1: Insert a Play Sound command to beep when the task finishes.

1. Open the task in the Task Editor.
2. Double-click the Play Sound command and select Play Beep option.
3. Click Save.
4. Move the Play Beep action to the last position in the task.

### Example 2: Play a recorded file.

1. Open the task in the Task Editor.
2. Double-click the Play Sound command, select Play Media File option and specify a file to play.
3. Select the media file to play, and click Save.

## PGP Command

*Available in Enterprise edition only.*

### Example 1: Create folders and encrypt and decrypt file using Passphrase.

1. Open the Task Editor.
2. See the below example with detailed steps and Comments explain the task steps.

```
1 If Folder Does Not Exist ("c:\pgp") Then  
2   Comment: Please enter the conditional commands here.  
3   Create Folder "c:\pgp"  
4 End If  
5 If File Does Not Exist ("c:\pgp\pgp.txt") Then  
6   Comment: Please enter the conditional commands here.  
7   Create File "c:\pgp\pgp.txt"
```

```
8 End If
9 Open File "C:\pgp\pgp.txt"
10 Keystrokes: [SHIFT DOWN]welcome to security[SHIFT UP]command "pgp.txt - Notepad" with
delay: 5 ms
11 Keystrokes: [CTRL DOWN]s[CTRL UP] in "pgp.txt - Notepad" with delay: 5 ms
12 Close Window: "pgp.txt - Notepad"
13 If Folder Does Not Exist ("c:\pgp\encrypt") Then
14   Comment: Please enter the conditional commands here.
15   Create Folder "c:\pgp\encrypt"
16 End If
17 PGP: Encrypt Files using Passphrase; Source: "C:\pgp\pgp.txt"; Destination: "C:\pgp\encrypt"
18 If Folder Does Not Exist ("c:\pgp\decrypt") Then
19   Comment: Please enter the conditional commands here.
20   Create Folder "C:\pgp\decrypt"
21 End If
22 PGP: Decrypt Files using Passphrase; Source: "C:\pgp\encrypt\"; Destination: "C:\pgp\decrypt\"
```

### **Example 2: Create folders and encrypt and decrypt file using Keys after creating the key files first.**

1. Open the Task Editor.
2. See the below example with detailed steps and Comments explain the task steps.

```
1 If Folder Does Not Exist ("c:\pgp") Then
2   Comment: Please enter the conditional commands here.
3   Create Folder "c:\pgp"
4 End If
5 If File Does Not Exist ("c:\pgp\pgp.txt") Then
6   Comment: Please enter the conditional commands here.
7   Create File "c:\pgp\pgp.txt"
8 End If
9 Open File "C:\pgp\pgp.txt"
10 Keystrokes: [SHIFT DOWN]welcome to creation of key files [SHIFT UP]bc in "pgp.txt - Notepad"
with delay: 5 ms
11 Keystrokes: [CTRL DOWN]s[CTRL UP] in "pgp.txt - Notepad" with delay: 5 ms
12 PGP: Create Keys; Public key: "C:\pgp\public.pkr"; Private key: "C:\pgp\private.skr"
13 Close Window: "pgp.txt - Notepad"
14 If Folder Does Not Exist ("c:\pgp\encrypt") Then
15   Comment: Please enter the conditional commands here.
16   Create Folder "c:\pgp\encrypt"
17 End If
18 PGP: Encrypt Files using Public Key; Source: "C:\pgp\pgp.txt"; Destination: "C:\pgp\encrypt"
19 If Folder Does Not Exist ("c:\pgp\decrypt") Then
20   Comment: Please enter the conditional commands here.
```

```
21 Create Folder "C:\pgp\decrypt"
22 End If
23 PGP: Decrypt Files using Public Key; Source: "C:\pgp\encrypt\"; Destination: "C:\pgp\decrypt\"
```

## PDF Integration Command

*Available in Enterprise edition only.*

**Example 1: Extract form fields from a downloaded and pre filled PDFs and split them, then merger and encrypt them.**

Open the Task Editor.

See the below example with detailed steps and Comments explain the task steps.

```
1 Download file '$FolderPath$fw4.pdf' from 'http://www.irs.gov/pub/irs-pdf/fw4.pdf'
2 Comment: After download, fill in details of the forms. In this example two pre-filled forms are
taken.
3 Comment: Below loop creates folders for the output files
4 If Folder Does Not Exist ("$OutPutFolderPath$") Then
5   Create Folder "$OutPutFolderPath$"
6   Create Folder "$OutPutFolderPath$$EmployeeData$"
7   Create Folder "$OutPutFolderPath$$DeductionSheets$"
8 End If
9 Comment: Below Loop will save images into $OutPutFolderPath$
10 Start Loop "Each File In Folder $FolderPath$"
11   PDF Integration: Convert "$FolderPath$$FileName$.pdf" to images as
"$PDFFileName$_$PDFTitle$" under folder "$OutPutFolderPath$$EmployeeData$"
12   PDF Integration: Convert "$FolderPath$$FileName$.pdf" to images as
"$PDFFileName$_$PDFTitle$" under folder "$OutPutFolderPath$$DeductionSheets$"
13 End Loop
14 Comment: Creation of one csv file as below to store the extract form fields.
15 Log to File: "First Name","Last Name","Home Address","City,State,Zip","Security
Number","Employer","Section 10" in "$OutPutFolderPath$$CsvFilePath$"
16 Comment: Below loop will process each file from from $FolderPath$ and extract form fields to
$OutPutFolderPath$\PDFOutput.csv
17 Start Loop "Each File In Folder $FolderPath$"
18   PDF Integration: Extract Form Fields from "C:\Sample PDF\fw41.pdf"
19   Log to File:
"$FirstName$","$LastName$","$HomeAddress$","$CityStateZip$","$SecurityNo$","$EmployerNam
e$","$AmountForSection10$" in "$OutPutFolderPath$$CsvFilePath$"
20 End Loop
21 Comment: Below loop will process each file from $FolderPath$ and will split into two parts that
```

is employee data and deduction sheet.

22 Start Loop "Each File In Folder \$FolderPath\$"

23 PDF Integration: Split "\$FolderPath\$\$FileName\$.pdf" into files as "\$FileName\$" under folder "\$OutPutFolderPath\$\\$EmployeeData\$"

24 PDF Integration: Split "\$FolderPath\$\$FileName\$.pdf" into files as "\$FileName\$" under folder "\$OutPutFolderPath\$\\$DeductionSheets\$"

25 End Loop

26 Comment: Merge all the documents into single file and save under \$OutPutFolderPath\$

27 PDF Integration: Merge Documents into "\$OutPutFolderPath\$\AllFormsDetail.pdf"

28 Comment: Encrypt document and save under \$OutPutFolderPath\$

29 PDF Integration: Encrypt Document "\$OutPutFolderPath\$\AllFormsDetail.pdf"; Save it as "\$OutPutFolderPath\$\EncryptedAllFormsDetail.pdf"

## Open Program\_File Command

### Example 1: Open Notepad.exe.

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click Open Program/File. The Open Program/File dialog appears.
3. In the Program/File Path text box, type 'notepad.exe'.
4. Click Save.
5. Optionally, if you want to open a file in Notepad called mynotes.txt, then type the path and filename in the Parameters text box, for example, C:\words\correspondence\mynotes.txt.

### Example 2: Run a batch file from Automation Anywhere Client.

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click Open Program/File. Automation Anywhere Client opens the Open Program/File dialog.
3. In the Program/File Path text box, type the path and name of the batch file; e.g., C:\My-Batch-File.bat.

4. Click Save.

### Example 3: Open a Microsoft Word document.

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click Open Program/File. Automation Anywhere Client opens the Open Program/File dialog.
3. In the Program/File Path text box, type the path and name of the batch file; e.g., C:\My Documents\smithletter.doc.
4. Click Save.



*Tip: If you use the Record feature in Automation Anywhere Client to create a task that opens an application or a file, you can make the task run faster by opening the task in the Task Editor and replacing the multiple keystrokes that you recorded with single Open Program/File commands, as described in the examples above.*

## Object Recorder Command

### Example : To record the setup process of Launch-n-Go setup.

Open the Launch-n-Go setup file.

Click on New > Object Recorder.

Select the Launch-n-Go setup file from the Select Window option. Check the **Use Intermediate delay** box.

Click Record.

Follow the instructions of the Launch-n-Go setup by using the appropriate control property options that the Object Recorder provides during each step.

When you finish with the setup file, select Save in the Object Recorder dialog box.

Type in a name for the Task and click Save in the Save Task window.

When you run this task, it will automatically execute the Launch-n-Go setup in the background.

You can also set the **SnapPoints** to view your actions, step by step, using the Visualize feature.

# OCR Command

*Available in Enterprise and Small Business editions only.*

## Example 1: Capture the text from a remote desktop software window

1. Open the Task Editor.
2. Double-click the OCR command.
3. Double click Capture Window.
4. Select the application window from the drop-down list.
5. Click View Captured Text. This will generate the image preview on the right side, together with the captured text.
6. If the captured text needs filtering, check the "Filter the captured text" check-box and specify the Before and After keywords.
7. Assign the value to a variable from the list.
8. Click Save.
9. Save the task.

## Using Wildcards (\*)

- You can specify wildcards (\*) in a window title.
- For example, use wildcards to perform an operation in any window that has the word "Untitled" anywhere in the title. You can place the wildcard character before and/or after "Untitled" to capture variations such as "Window 1: Untitled" or "Untitled - Text Document", etc.
- To capture all variations of "Untitled," simply specify \*Untitled\*.
- Automation Anywhere Small Business Client will first search for a window titled exactly "Untitled" and if it does not find it, it will look for windows with the word Untitled anywhere in the title.

# Message Box Command

### **Example 1: Insert Message Box commands to instruct the user how to change the screensaver on a PC.**

1. Insert a Message Box command that says 'Click on Start button and then click on Control Panel.'
2. Insert a Delay of 2 seconds.
3. Insert another Message Box command that says 'Double-click on the Display.'
4. Insert another Message Box command that says 'Select the Screen Saver Tab.'
5. Insert another Message Box command that says 'Select your Screen Saver and click on Apply, then click OK.'

### **Example 2: Create an automated interactive task that tells the user how to perform the next action.**

1. Open the Task Editor.
2. Create a task and insert a Message Box command at appropriate locations to tell the user which actions to take.
3. Insert a Pause command after the Message Box command. Sometimes users may hit OK without reading the complete message. This ensures that the task does not proceed but pauses until the user manually resumes it.

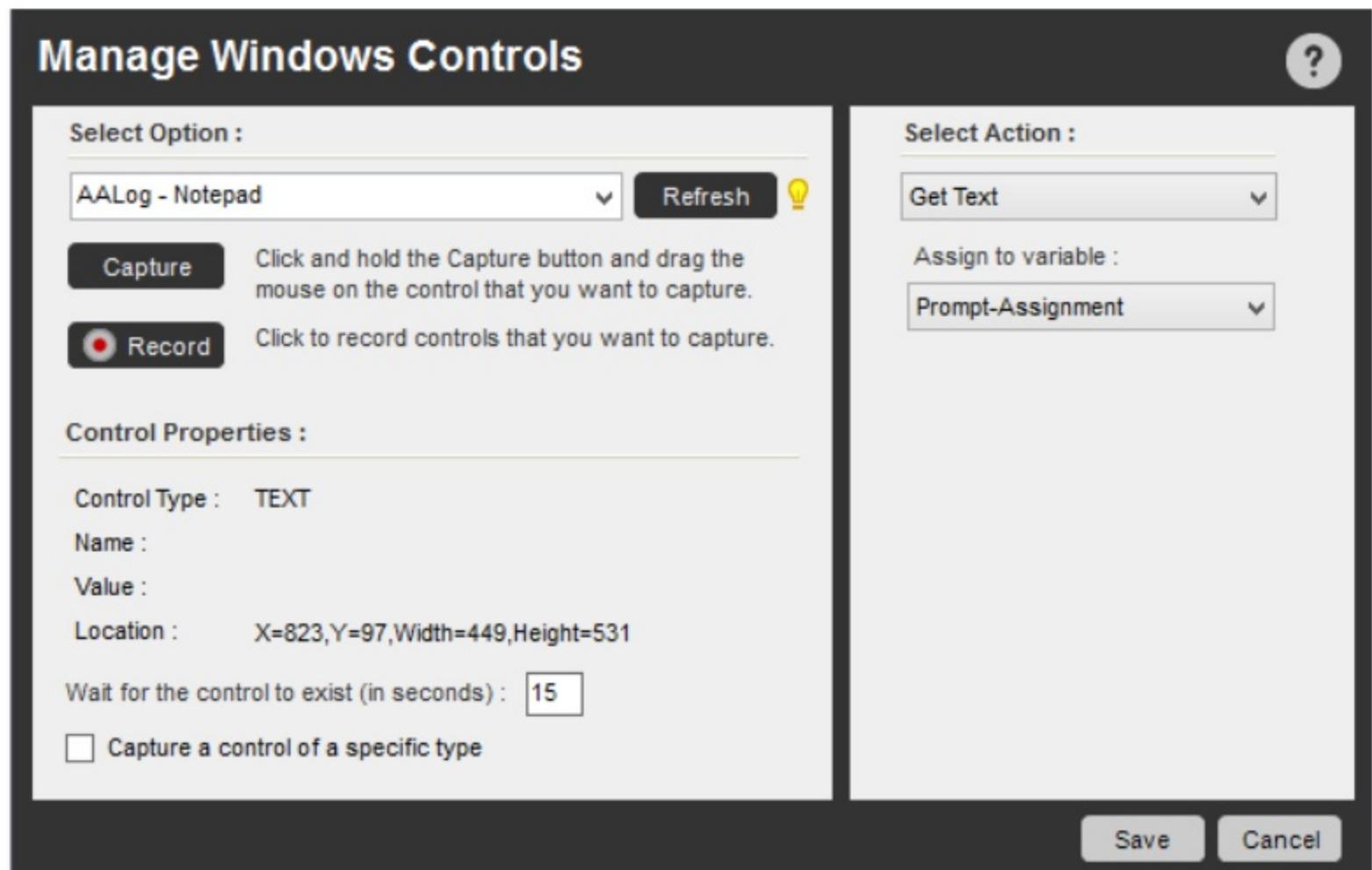
## **Manage Window Controls Command**

*Feature also available in Standard edition version 8.2 or higher.*

### **Example 1: Extract a PC's event log.**

1. Open the event log whose contents you want Automation Anywhere Client to extract.
2. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
3. In the Task Editor, in the Commands list, double-click Manage Windows Controls. The Manage Windows Controls dialog appears.
4. From the Select Window drop-down list, select the window that contains the log file data you want to capture.

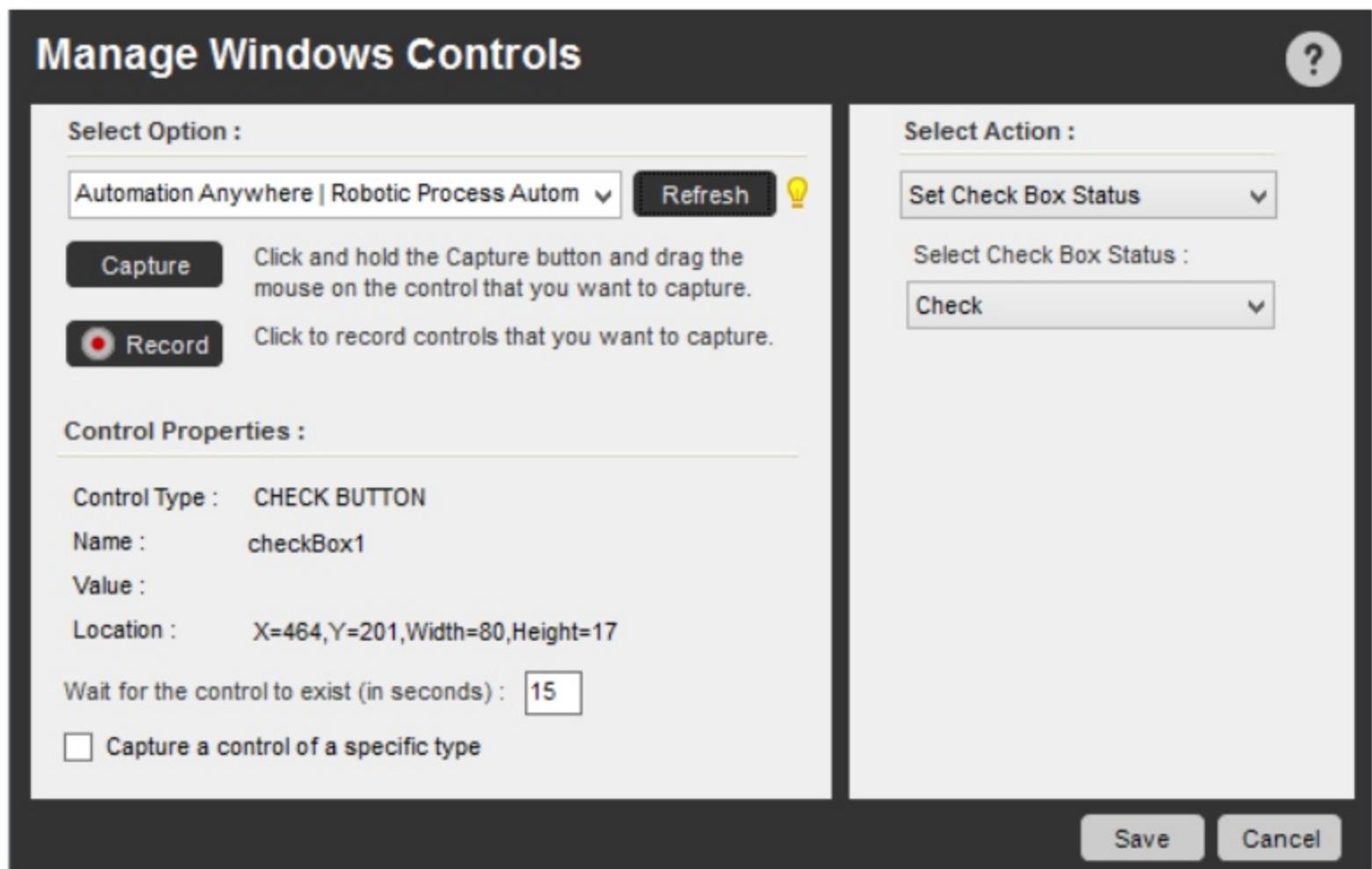
5. Click and hold Capture and select the text in the log file.
6. In the Manage Windows Controls dialog, select Get Text in the right pane under Select Action.
7. In the Assign to variable drop-down, select the variable to which you want to assign the log file. (To learn more about variables, see Using Variables.)
8. Click Save.



#### Example 2: Get (or set) the status of a check box.

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click Manage Windows Control. The Manage Windows Controls dialog appears.
3. Click and hold Capture and drag the mouse to the checkbox whose status you want to capture.
4. From the Select Window drop-down, select the title of the window that contains the checkbox whose status you want to get or set.

5. To get the status of the checkbox:
  - In the Manage Windows Controls dialog, select Get Check Box Status in the right pane.
  - In the Assign to variable drop-down, choose a variable to assign the checkbox status to. (For detailed information on variables, see Using Variables.)
6. To set the status of the checkbox:
  - In the Manage Windows Controls dialog, select Set Check Box Status in the right pane.
  - In the Select Check Box Status select Check, Uncheck or Toggle for the appropriate action you want to perform.
7. Click Save.



#### Example 3: Enter data from a comma-separated (CSV) data file into an application.

1. Open the application where you want to enter data.
2. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.

3. In the Task Editor, in the Commands list, double-click Excel. A submenu appears.
4. Double-click the Open Spreadsheet command. The Excel dialog appears.
5. In the Excel dialog, fill in the Spreadsheet Path and Specific Sheet Name text boxes, then click Save.
6. In the Task Editor, in the Commands list, double-click Excel > Get Cells. The Excel dialog appears.
7. In the Excel dialog, select Get Multiple Cells.
8. Fill in the From Specific Cell and To Specific Cell text boxes: type the coordinates of the first and last cells of the data you want to capture.
9. Click Save.
10. In the Task Editor, double-click Loop. A submenu appears.
11. Double-click Each Row In An Excel Dataset. The Loop dialog appears.
12. Type a session name, and click Save.
13. In the Task Editor, in the Commands list, double-click Manage Windows Controls. The Manage Windows Controls dialog appears.
14. From the Select Window drop-down, select the application window where you want to insert the captured CSV data.
15. Click and hold the Capture button and drag the mouse onto the control you want to select. Then select the required action to set the value of the control to that from the Excel dataset.
16. Click Save.
17. In the Task Editor, in the Commands list, double-click Loop. A submenu appears.
18. Double-click 'Each Row In An Excel Dataset.' The Loop dialog appears.
19. Select End Loop.
20. Click Save.
21. In the commands list, double-click Excel. A submenu appears.
22. Double-click Close Spreadsheet. The Excel dialog appears.

23. Click Save.

#### **Example 4: Select options to generate a report with an application.**

1. Open the application from which you want to generate the report, and open the application's report-generation options window.
2. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
3. In the Task Editor, in the Commands list, double-click Manage Windows Controls. The Manage Windows Controls dialog appears.
4. In the Select Window drop-down, select the report generation options window.
5. Click and hold Capture and drag the mouse onto the control you want to select.
6. In the Manage Windows Controls dialog, in the right pane under Select Action, select the action to set the value of the control.
7. Click Save.
8. Repeat steps 5 to 7 for each of the options required to generate the report.
9. In the Manage Windows Controls dialog, click and hold Capture and drag the mouse in the application onto the button to be clicked to save the options you entered and begin processing the report.
10. In the Manage Windows Controls dialog, click Save.

#### **Example 5: Extract a table from a Web page.**

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click Web Recorder and double-click Open Browser. The Web Recorder dialog appears.
3. In the URL to Open text box, type the URL of the page you want Automation Anywhere Client to open.
4. Click Save.

5. In the Commands list, double-click Manage Windows Controls. The Manage Windows Controls dialog appears.
6. In Internet Explorer, open the page that contains the table you want to extract.
7. In the Manage Windows Controls dialog, in the select Window drop-down, select the Internet Explorer window.
8. Click and hold Capture and drag the mouse onto the Internet Explorer status bar.
9. In the right pane of the Manage Windows Controls dialog, in the Select Action list, select the variable to which you want to assign the value of the status bar.
10. Click Save.
11. In the Commands list, double-click If/Else. A submenu appears.
12. Double-click Variable. The If dialog appears.
13. Click Edit. The If Variable dialog appears.
14. In the Variable text box, type the name of the variable you specified above.
15. In the Operator drop-down, select Equal To (=).
16. Select Fix, and in the Fix text box, type a value.
17. Click Save.
18. In the Commands list, double-click Web Recorder. A submenu appears.
19. Double-click Extract Table. The Web Recorder dialog appears.
20. In the Step 1 text box, type the URL of the Web page where you want to capture the table.
21. In the Step 3 text box, type the name of a CSV file where you want to save the table data.



*Note: The Extract Table command above should be within the If and End If actions.*

22. Click Save.
23. In the Commands list, double-click Web Recorder, and in the submenu double-click Close Browser. The Web Recorder dialog appears, with Close Browser selected.
24. Click Save.

### **Using Wildcards (\*)**

- You can use wildcards (\*) when you specify a window title in any Manage Windows Controls dialog.
- For example, if you want to perform any operation in any window that contains the word 'Untitled,' insert the '\*' wildcard character to indicate any number of characters before and/or after 'Untitled': '\*Untitled\*.'
- Automation Anywhere Client will first search for a window titled 'Untitled,' and if it does not find it, it will look for other windows whose names include the word 'Untitled.'
- You can insert the wildcard character (\*) at the beginning or end of the window title.

## **Loop While Command**

### **Example 1: Loop While for Object Property of button**

1. Open the Test Editor.
2. Double-click the Loop command and select the option Times, then type 5 in the text box.
3. Save the command. It automatically inserts an End Loop command.
4. Insert an Open Program/File command to open IEXPLORE.exe within the Start and End Loop commands.

### **Example 2: Loop through a list and insert one value at a time in an already-open Excel document.**

1. Create a list variable from Tools > Variable Manager (we'll call it \$my-list\$).
2. Specify an option to have the variable read from a text file.
3. Let's say you have filled the values of the list variable.
4. Double-click the Loop command.
5. Select List and select \$my-list\$ from the drop-down list.
6. This inserts Start and End Loop commands in the editor.
7. Insert a Keystroke action \$my-list\$[ENTER in the Excel window. It will type the next value of the list variable each time it loops.

**Example 3: Monitor an application and send email to the administrator if the application shuts down.**

1. Double-click the Loop command.
2. Select Condition.
3. Select Application Running and specify the path of the application that you want to monitor for (e.g., C:\My-Application.exe).
4. Save the command. It inserts Loop and End Loop commands.
5. Insert a Send Email command after the End Loop command.
6. When you run the test, it loops as long as the application is running. As soon as the application crashes or shuts down, it leaves the loop and sends an email alert.

**Example 4: Query for all contacts in the database that have last name beginning with 'R' and print the names in an Excel document.**

1. Using database commands, connect to a database and issue an SQL query to fetch your records. Insert a Connect command first, then an SQL Query command.
2. Double-click the Loop command.
3. Select "Each row in a SQL Query dataset."
4. To access the result returned by the SQL query, use the \$Dataset Column\$ system variable provided by Testing Anywhere Client. Use the 'Loop for Each row in a SQL query dataset' option to loop through all rows of the SQL query result. To access each column, use \$Dataset Column(index)\$; for example, to get the first column use \$Dataset Column(1)\$, and so on.
5. You can use the Insert Keystrokes commands with the \$Dataset Column\$ variable to type the results in the Excel document.
6. Disconnect from the database.

**Example 5: Print prices extracted from a pricing table on a Web page.**

1. Use the Web Recorder 'Extract Table' command to extract the tabular data from the Web page.
2. After you call the Extract Table command, Testing Anywhere Client stores all of the tabular data in a system variable called \$Table Column(index)\$ . You can use this variable to get all the rows and columns of the table you extracted.

3. Double-click the Loop command.
4. Select 'Each row in an Internet Explorer Table.'
5. To access each row and column in the \$Table Column\$ system variable, use the 'Loop for Each row in an Internet Explorer table' option to loop through the rows of the table. To access each column, use \$Table Column(index)\$; for example, to get the first column, use \$Dataset Column(1)\$, and so on.
6. You can use the Insert Keystrokes command with the \$Table Column\$ variable to type the results in the Excel document.

#### **Example 6: Insert an Exit Loop command to break out of the loop.**

1. Open the Test Editor.
2. Double-click the Loop command and select Exit Loop, and move it to the appropriate location between the Start and End Loop commands.
3. The Exit Loop command can appear only between Start Loop and End Loop statements.

#### **Example 7: Insert a Loop for each mail message in your mail server.**

1. Click Each Message in Server option.
2. Edit and fill up required details like Hostname, Username, Password, Port number and specify a folder to Save attachment in.
3. Save the form.
4. Edit the test. Insert variables to download email attributes like Subject, Message, To, Cc, Attachment.
5. This loop extracts and downloads all mail details as specified.
6. Using Nested Loop
7. Let's say you are using a nested loop as follows:
8. Start Loop "List Variable my-list-variable"
9. Variable Operation: \$my-list-variable\$ To \$Income\$
10. Start Loop "5" Times
11. Keystrokes: \$Income\$[ENTER] in "Untitled - Notepad"

12. End Loop (for "5" times)
13. End Loop (for my-list varibale)
14. You should assign the value of the list variable to a value type variable before the nested loop starts, and use the value type variable inside the nested loop.
15. You can apply the same tip when you are using Loop for each row in an Internet Explorer Table, Loop for each row in a SQL Table, or Loop for each row in an Excel dataset.

**Example 8: Insert Loop While command with 'Wait' time for a 'File to Exist' when downloading data.**

1. From the commands list, double click/drag and drop the 'Open File/Folder' command.
2. Insert an 'If file exists' condition.
3. Add delay of 50 milliseconds.
4. Insert 'Delete Files' command and specify the file that should be deleted. This is useful as it deletes files with same name downloaded earlier and allows it to be saved to the same location.
5. End the 'If' command.
6. Insert a 'Web Recorder' command with 'Manage Web Controls' selected.
7. Specify the URL. The website is launched.
8. Capture the required actions - in this case, clicking the download link to save an offline copy of the data to a csv file.
9. The required parameters are populated in the 'Web Recorder' command window.
10. Insert an 'Activate Window' command to activate the website window.
11. Insert a 'Loop While' for 'File does not exists'. This ensures that the program will keep looping till the file does not exists and will loop out as soon as it finds the required file.
12. Add 'Wait time for the condition to be true'. The time set could be 5 seconds if you feel the file can be downloaded within this time frame.
13. End the loop.
14. Open the csv file and verify your data.

**Example 9: Insert Loop While command with 'Wait' time for 'Ping Unsuccessful' when uploading files to a server.**

1. From the commands list drag and drop,double click the 'Loop While' command for 'Ping Unsuccessful' with 'Wait for condition to be true' enabled. This ensures that the loop will be executed till the ping is unsuccessful. The task exits the loop as soon as it validates a ping.
2. Insert 5 seconds as the time and 'End' the loop.
3. Connect to a server using the 'Connect' sub-command in 'FTP/SFTP'.
4. Use 'Start Loop' with 'Each file in Folder' option selected. Here, loop command is used to ensure that all files in the folder are selected for upload to a path specified in the next step.
5. In the 'FTP/SFTP' command select the option 'Put Files' and specify the path.
6. Transfer the data to a csv file using the 'Log to File' command.

7. End the loop.
8. Use the 'Disconnect' sub command of FTP/SFTP.
9. Now mail the data file to the user specified in the 'Send Email' command.

## Loop Command

### Example 1: Open five Internet Explorer windows with one click.

1. Open the Task Editor.
2. Double-click the Loop command and select the option Times, then type 5 in the text box.
3. Save the command. It automatically inserts an End Loop command.
4. Insert an Open Program/File command to open IEXPLORE.exe within the Start and End Loop commands.

### Example 2: Loop through a list and insert one value at a time in an already-open Excel document.

1. Create a list variable from Tools > Variable Manager (we'll call it \$my-list\$).
2. Specify an option to have the variable read from a text file.
3. Let's say you have filled the values of the list variable.
4. Double-click the Loop command.
5. Select List and select \$my-list\$ from the drop-down list.
6. This inserts Start and End Loop commands in the editor.
7. Insert a Keystroke action \$my-list\$[ENTER in the Excel window. It will type the next value of the list variable each time it loops.

### Example 3: Monitor an application and send email to the administrator if the application shuts down.

1. Double-click the Loop command.
2. Select Condition.
3. Select Application Running and specify the path of the application that you want to monitor for (e.g., C:\My-Application.exe).

4. Save the command. It inserts Loop and End Loop commands.
5. Insert a Send Email command after the End Loop command.
6. When you run the task, it loops as long as the application is running. As soon as the application crashes or shuts down, it leaves the loop and sends an email alert.

**Example 4: Query for all contacts in the database that have last name beginning with 'R' and print the names in an Excel document.**

1. Using database commands, connect to a database and issue an SQL query to fetch your records. Insert a Connect command first, then an SQL Query command.
2. Double-click the Loop command.
3. Select "Each row in a SQL Query dataset."
4. To access the result returned by the SQL query, use the \$Dataset Column\$ system variable provided by Automation Anywhere Client. Use the 'Loop for Each row in a SQL query dataset' option to loop through all rows of the SQL query result. To access each column, use \$Dataset Column(index)\$; for example, to get the first column use \$Dataset Column(1)\$, and so on.
5. You can use the Insert Keystrokes commands with the \$Dataset Column\$ variable to type the results in the Excel document.
6. Disconnect from the database.

**Example 5: Print prices extracted from a pricing table on a Web page.**

1. Use the Web Recorder 'Extract Table' command to extract the tabular data from the Web page.
2. After you call the Extract Table command, Automation Anywhere Client stores all of the tabular data in a system variable called \$Table Column(index)\$ . You can use this variable to get all the rows and columns of the table you extracted.
3. Double-click the Loop command.
4. Select 'Each row in an Internet Explorer Table.'
5. To access each row and column in the \$Table Column\$ system variable, use the 'Loop for Each row in an Internet Explorer table' option to loop through the rows of the table. To access each column, use \$Table Column(index)\$; for example, to get the first column, use \$Dataset Column(1)\$, and so on.

6. You can use the Insert Keystrokes command with the \$Table Column\$ variable to type the results in the Excel document.

#### **Example 6: Insert an Exit Loop command to break out of the loop.**

1. Open the Task Editor.
2. Double-click the Loop command and select Exit Loop, and move it to the appropriate location between the Start and End Loop commands.
3. The Exit Loop command can appear only between Start Loop and End Loop statements.

#### **Example 7: Insert a Loop for each mail message in your mail server.**

1. Click Each Message in Server option.
2. Edit and fill up required details like Hostname, Username, Password, Port number and specify a folder to Save attachment in.
3. Save the form.
4. Edit the task. Insert variables to download email attributes like Subject, Message, To, Cc, Attachment.
  - This loop extracts and downloads all mail details as specified.

### **Using Nested Loop**

Let's say you are using a nested loop as follows:

1. Start Loop "List Variable my-list-variable"
  2. Variable Operation: \$my-list-variable\$ To \$Income\$
  3. Start Loop "5" Times
  4. Keystrokes: \$Income\$[ENTER] in "Untitled - Notepad"
  5. End Loop
  6. End Loop
- You should assign the value of the list variable to a value type variable before the nested loop starts, and use the value type variable inside the nested loop.

- You can apply the same tip when you are using Loop for each row in an Internet Explorer Table, Loop for each row in a SQL Table, or Loop for each row in an Excel dataset.

## Log To File Command

### Log to File Command

**Example 1: Monitor the performance of a task by inserting a Log to File command in the task. When task playback reaches the Log to File command, Automation Anywhere Client creates a time-stamp in a log file.**

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools -> Task Editor.
2. In the Task Editor, in the Commands list, double-click Log to File. The Log to File dialog appears.
3. In the Log File text box, type a path and name for the log file, or click Browse to choose an existing file.
4. Optionally, in the text box, type any text that Automation Anywhere Client should add to the log file each time it creates an entry in the file.
5. Optionally, select Add Timestamp. Each time the task runs, Automation Anywhere Client will insert the current date and time.
6. After the task runs, you can open the log file (a simple text file) and check the actions that Automation Anywhere Client has played back. Log files can help you measure the performance and accuracy of your tasks.

**Example 2: Log the clipboard contents to help debug a task that copies data.**

Let's say you are using loops within your tasks, and you want to check the text in the clipboard variable during each pass of the loop.

Automation Anywhere Client provides system variables called \$Counter\$ and \$Clipboard\$. You can log these variables, for example, by including a \$Counter\$-\$Clipboard\$ variables in a Log to File command. For complete information on using variables, see the Using Variables help section.

## Internet Connection Command

**Example 1: Connect to 'myDialUp'.**

1. Open the Task Editor.

2. Double-click the Internet Connection command.
3. Select the Connect option in the 'Internet Connection' category.
4. Select your dial-up connection from the drop-down list.
5. Click Save.

## Launch Website Command

*Available in Enterprise and Small Business editions only.*

**Example 1: Open the website www.cnn.com.**

1. Open the Task Editor.
2. Double-click the Launch Website command.
3. Type 'www.cnn.com' in the 'Enter a URL and click Launch' text box.
4. Click Launch.
5. Click Refresh.
6. Select the title 'CNN.com - Microsoft Internet Explorer' in the drop-down.
7. Click Save.

## Insert Mouse Scroll Command

**Example 1: Perform a scroll after downloading a PDF file. Then extract data from PDF and save in a CSV file.**

1. Open the Task Editor.
2. To use Mouse Scroll, double-click the Insert Mouse Scroll command in the task.
3. Follow the below steps of the example. Comments explains the command steps for the task.
  - Similar test cases can be generated by providing appropriate Mouse Scroll iterations for up and down.
1. Comment: Download User Manual from website, using Web Recorder -> Download File command

2. Download file 'C:\AAUserManual.pdf' from  
<http://www.automationanywhere.com/support/downloads/AAUserManual.pdf>
3. Comment: Open file
4. Open File "C:\AAUserManual.pdf"
5. Comment: Scroll upto second page, one can choose delay or can scroll without delay  
Mouse Scroll: Direction: Down; Iterations: 15
6. Comment: Capture data from PDF file
7. App Integration: Capture text from window "AAUserManual.pdf - Adobe Reader" and assign to \$Prompt-Assignment\$
8. Comment: Extracted data and save as CSV file
9. Log to File: "\$Prompt-Assignment\$" in "C:\Features and capabilities.csv"

## Insert Mouse Move Command

**Example 1: Insert a mouse move command that moves the mouse cursor from position (200,300) to (400,500) in the active application window.**

1. In Automation Anywhere Client, click New Task, Task Editor, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click Insert Mouse Move.
3. Select Mouse Position (From).
4. Click Capture. Automation Anywhere Client switches to the active application window.
5. Left-click and drag the mouse to location (200,300) and release the mouse button.
6. In the Insert Mouse Move dialog, select Mouse Position (To).
7. Click Capture. Automation Anywhere Client switches to the active application window.
8. Left-click and drag the mouse to location (400,500) and release the mouse button.
9. In the Insert Mouse Move dialog, click Save.

## Insert Mouse Click Command

### **Example 1: Perform a mouse click at the Yahoo! sign-in dialog.**

1. Open mail.yahoo.com in your Web browser.
2. In Automation Anywhere Client, click New > Task Editor, or Tools > Task Editor.
3. In the Commands list, double-click the Insert Mouse Click command.
4. Click Refresh.
5. From the Select the Window... drop-down, select 'Sign in to Yahoo! - Microsoft Internet Explorer'.
6. Click Capture and hold the left mouse button. Automation Anywhere Client activates the 'Sign in to Yahoo! - Microsoft Internet Explorer' window.
7. Click the Yahoo! 'Sign In' button, and release the mouse button.
8. In the Insert Mouse Click dialog, click Save.

 *Note: To insert other types of clicks (double-click, right-click, etc.), click Advanced View and select the appropriate options.*

## **Insert Keystrokes Command**

### **Example 1: Insert keystrokes in a Microsoft Word document titled news.doc.**

1. Open a document named news.doc in Microsoft Word.
2. In Automation Anywhere Client, click New, and in the Create New Task dialog, click Task Editor.
3. In the Commands list, double-click Insert Keystrokes or drag and drop from the command panel.
4. Click the downward-pointing arrow to the right of the Select Window text box, and in the drop-down list select news.doc - Microsoft Word.
5. Insert the required keystrokes. For example, type Today's tasks are complete and under Special Keystrokes choose Enter.
6. Click Save.

### **Example 2: Copy a cell from an Excel document titled data.xls.**

1. Open a document named data.xls in Microsoft Excel.
2. In Automation Anywhere Client, click New, and in the Create New Task dialog, click Task Editor.
3. In the Commands list, double-click Insert Keystrokes.
4. Click the downward-pointing arrow at the right of the Select Window text box, and in the drop-down list click Microsoft Excel - data.xls.
5. Follow these steps to perform a copy operation by inserting a Ctrl+C command in the Keystrokes window:
  - a. In the Special Keystrokes section, click Ctrl Down, then type 'c' in the Keystrokes window, and under Special Keystrokes click Ctrl Up. The Keystrokes window now displays: [CTRL DOWN]c[CTRL UP].
  - b. Click Save.

### **Example 3: Insert the current date and time.**

For example, to include the date and time in filenames, so that you can identify the files that were created when you ran the task.

Automation Anywhere Client provides system variables that you can use to insert the current date and time and other useful data. The date system variables are: \$Date\$, \$Year\$, \$Month\$, \$Day\$, \$Hour\$, \$Minute\$, and \$Second\$. You can configure the \$Date\$ variable to print the date in your preferred format.

1. In the Task Editor, choose Tools > Variable Manager > System Variables > Date. Click the '...' button to the right of the Date variable, and in the Date format text box, edit the format, or click the down-arrow to the right of the Date format text box and select a standard format from the drop-down list.
2. For example, to name an Excel file MyExcelResults plus the current date and time, open the File Save dialog in Excel, then follow steps 1 and 2 above to insert the following in the Keystrokes text box: MyExcelResults-\$Year\$-\$Month\$-\$Day\$-\$Hour\$-\$Minute\$-\$Second\$.xls.
3. To configure your own date and time format, using the System variables \$Year\$, \$Month\$, \$Day\$, \$Hour\$, \$Minute\$, \$Second\$, follow these steps:
4. In the Task Editor, double-click Insert Keystrokes in the Commands list.
5. Click the down-arrow to the right of the Select Window text box, and from the drop-down list select the window where you want to insert the date and time.

6. From the Insert Variable list, select a system variable (e.g., Date) and click Insert. The variable appears in the Keystrokes text box.

**Example 4: Insert keystrokes in any currently active window.**

1. Open the window of the application where you want to enter keystrokes.
2. In the Task Editor, double-click Insert Keystrokes in the Commands list.
3. From the Select Window drop-down list, select Currently Active Window.
4. Enter keystrokes and commands in the Keystrokes text box, as described in Example 1 above.
5. Click Save. When you run the task, Automation Anywhere Client will insert your keystrokes.

## Image Recognition Command

*Feature also available in Standard edition version 8.2 or higher.*

**Example 1: Search for an image in a website and click it if found.**

1. Open the Task Editor.
2. Double-click the Launch Website command.
3. Specify the URL of the website where you want to search for the image.
4. Click Launch.
5. Click Refresh.
6. Select the window where the website has been launched.
7. Click Save.
8. Assign the value to a variable from the list.
9. Double-click the Image Recognition command.
10. Click Save.

When you run the task, it will launch the website and click Image 2 if it is found within Image 1.

### Using Wildcards (\*)

- For the options Window Exists, Window Does Not Exist, and Image Recognition, you can specify wildcards (\*) in the beginning, middle and end.
- For example, if you want to perform an operation if any window has the word 'Microsoft' in the title, you can use wildcards to indicate any string before or after 'Microsoft'.
- For example, you can specify: IF Window Exists \*Microsoft\*.
- Automation Anywhere Client will first search for the exact window title ('Microsoft'), and if it does not find it, it will look for windows with the word Microsoft anywhere in the title.

## If-Else Command

### Example 1: Close the Windows Notepad window (if open).

1. Open Windows Notepad. A new file opens automatically, titled 'Untitled Notepad.'
2. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
3. In the Task Editor, in the Commands list, double-click If/Else. A submenu appears.
4. Double-click Window Exists. The If dialog appears, with Window exists selected.
5. In the Select Title drop-down, select 'Untitled Notepad'.
6. Click Save.

 Note: You should insert the Close Window command between If and End if statements in the task.

### Example 2: Search for a record. If found, then perform a different set of actions.

1. Automation Anywhere Client can search for an item, a word, or a record in an application, and if found, perform a set of actions that you specify; and if not found, perform a different set of actions.
2. The following example shows how to search for an item, word, or record that doesn't exist, so that you see the standard application window that says 'Not Found' (or equivalent wording).
3. Open the application, and display a 'Not Found' (or equivalent) page by searching for nonexistent data.

4. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
5. In the Task Editor, in the Commands list, double-click If/Else. A submenu appears.
6. Double-click Window Exists. The If dialog appears.
7. In the Select Title drop-down, select the 'Not Found' window in the application.
8. Click Save.

 *Note: You would perform steps 1 to 6 to create a new condition within If and End If statements*

### **Example 3: Make a copy of the log file named mylog.txt, if it exists.**

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click If/Else. A submenu appears.
3. Double-click File Exists. If dialog appears with Window exists selected.
4. In the Select File text box, type (or click Browse to browse to) the path and filename of the mylog.txt file.
5. Click Save. Automation Anywhere Client inserts If File Exists and End if commands in the task.
6. In the Task Editor, in the Task Actions List, click the last action before the End If statement.
7. In the Commands list, double-click Files/Folders. A submenu appears.
8. Double-click Copy Files. The Files/Folders dialog appears.
9. In the Source File(s) and Destination Folder text boxes, type or browse to the path and filename of the file you want to copy, and the folder that you want to copy it to.
10. Click Save.

 *Note: Automation Anywhere Client has inserted the Copy Files action immediately above the End If statement. When you run the task, Automation Anywhere Client will check if the file exists, and if it does, it will copy the file to the folder you specified.*

#### **Example 4: Create a task that prompts the user for a value, and if the user types 'STOP' the task stops.**

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click Prompt. A submenu appears.
3. Double-click Prompt For Value. The Prompt dialog appears.
4. Click 'Assign the value to an existing variable.'
5. Click Save. Automation Anywhere Client inserts If File Exists and End if commands in the task.

 *Note: This example uses variables for illustration purposes only; to learn how to create and use variables, see [Using Variables](#).*

1. In the Select Variable pull-down, select a variable that you previously created (for example, \$User-Input\$).
2. Insert an IF condition that compares the value of the \$User-Input\$ variable with the text 'Stop.' In the IF command, select the option 'Variable'. Click Edit. Specify the variable \$User-Input\$ in the first text box, select Includes, and under Fix, type 'STOP'.
3. Insert a Stop task command within the IF-End if commands.
4. Set the Repeat property of the task to 'Repeat Until I Stop.'
5. This task will continue to repeat until the user enters 'STOP' when prompted.

#### **Example 5: To open a downloaded file, if it exists (if not, prompt for another file before proceeding).**

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click Prompt. A submenu appears.
3. Double-click Prompt For Value. The Prompt dialog appears.
4. Select 'If-File Exists' and type the path to the file in the Select File text box, e.g., C:\Documents and Settings\...\Desktop\Downloadfile.txt.

5. Click Save.
6. In the Commands list, double-click Open Program/File.
7. Type the program/file path.
8. Click Save.
9. In the Commands list, double-click If-Else, then click End if in the submenu.
10. Select Else and click Save.
11. Double-click Prompt for File/Folder.
12. Type a custom caption and message.
13. Under Select File/Folder, select File.
14. Assign it to a list variable from the list.
15. Click Save.
16. Double-click Open Program/File again.
17. In the program/file path, press F2 and insert the list variable that was used in Prompt for File/Folder.
18. Click Save.

**Example 6:** Create a task that checks if a specific Web server in a cluster is up and running.

1. Open the Task Editor.
2. Double-click If-Else, then EndIf in the submenu.
3. Select If-Ping-Successful and type the IP address or the hostname of the first Web server in the cluster.
4. Click Save.
5. Select Else If from the If/Else menu. Then select If-Ping-Successful and type the IP address or the hostname of the next Web server in the cluster.
6. Click Save.
7. Repeat steps 5 and 6 for all Web servers in the cluster.

8. Select Else from the If/Else menu.
9. Double-click Run Task and select a task that you want to run if all Web servers are down.
10. Click Save.

**NEW! Example 7: Create a task that sends notifications to Automation Anywhere trial users to opt for a purchase version.**

1. In Automation Anywhere client, launch the Task Editor to create a new task.
2. Create variables 'Trial\_Period', 'Trial\_Run' and 'Trial\_Saves'. These shall be used in the task for If conditional commands.
3. Now, you need to tap into customer details to find out when their evaluation package is about to expire. Use the 'Read from CSV/Text' command to achieve this.

 *Note: Here we assume that a CSV file is saved to your machine that contains the required data. Also, each column in the file should contain headers/titles that match the ones specified as variables (see image)*

4. Insert a 'Loop' command to read through each column and fetch the data to make the condition given in the next step true.
5. Insert an 'If Variable' with 'Match Any' selected in the conditions window, which is invoked on clicking 'Add more conditions'.
6. Key in the parameters such as 'Trial\_Period' Equal to (= )'14' OR 'Trial\_Run' Equal To (= ) '50 OR 'Trial\_Saves' Equal to (= ) '50' for the OR condition 'Match Any'. If any one of the conditions is satisfied for a user, s/he will be notified about the pending upgrade.

*(Image as seen in Enterprise edition 8.0)*

- Use the 'Add' button to add as many conditions as required.

 *Note: You can use only a single condition type and cannot use both i.e. Match Any + Match All.*

- The 'event data' is populated in the IF command window:
- You will have to send notifications to your evaluation customers to opt for the purchased product version, for which you can use the 'Send Email' command.
  - Insert the command between the If and End If commands.
  - Insert the End Loop at this juncture i.e. at the very end.
  - Save the task.

**NEW! Example 8: Search for books by a specific author on an e-com site using the 'AND' condition**

- In Automation Anywhere client, launch the Task Editor to create a new task.
- Create value type variables 'Books' and 'Author' to be used later into the task.
- From the commands pane, double click/drag and drop the 'Web Recorder' command and specify url for the e-commerce site you wish to navigate to.
- You can now record the steps that lead to searching books by a specific author available on the website with the help of the 'Manage Web Controls' commands.



*Tip: Since you will be using 'Manage Web Controls' command a lot in this task, you can copy and paste where required. Simply change the parameters!*

- Launch the e-com website with the help of 'Manage Web Control' command and in the 'Advanced View' choose 'Select Item By Text' and enter item name 'Books'. This action will enable you to select 'Books' from the drop-down list provided in the website.
- Now you need to assign the same to a variable 'Books' which we did in the beginning. Select the 'Manage Web Controls' command and in the 'Advanced View' choose 'Get Selected Item' and 'Assign to Variable' 'Books'.
- Next, time to key in the name of your favorite author! Use the 'Set Text' action in the 'Advanced View' of the 'Manage Web Controls' command. Key in the author name in the text box 'Text to Set'.
- The selected author should be assigned to a variable in the next step. Use the 'Get Text' action to do this from the Manage Web Controls' command.

8. You can now use conditions to scroll through the results on different pages. Insert an 'IF variable' condition. Use the 'Match All' option in the 'If Variable' window.
9. Set parameters as shown:

*(Image as seen in Enterprise edition 8.0)*

- The 'event data' is populated in the IF command window:
- 
- You can edit the parameters if you want to change those.
10. Use the 'Manage Web Controls' command to click on the 'Search' button/control. Choose 'Perform on Click' in Select action in the Advanced View.
  11. You can choose to specify the number of pages to be searched on the website for which you can include a 'Loop' command. Here, let's specify '5 Times' as the loop.
  12. Insert another 'Perform on Click' action here so that the task moves to the next page automatically till 5 have been reached.



*Note: Here, insert a delay of 5 or 10 seconds for the page to load completely before moving to the next one.*

13. End loop.
14. Save the task.

## Using Wildcards (\*)

- For the options Window Exists, Window Does Not Exist, and Image Recognition, you can specify wildcards (\*) in the beginning, middle and end.
- For example, if you want to perform an operation if any window has the word 'Microsoft' in the title, you can use wildcards to indicate any string before or after 'Microsoft'.
- For example, you can specify: IF Window Exists \*Microsoft\*.
- Automation Anywhere Client will first search for the exact window title ('Microsoft'), and if it does not find it, it will look for windows with the word Microsoft anywhere in the title.

# FTP\_SFTP Command

## Example 1: Download a file routinely from an FTP/SFTP server

1. Open the Task Editor.
2. Double-click the FTP/SFTP command. Select the Connect option.
3. Specify the server address, e.g., `ftp://yourserver.com`, and the username/password, and click Save.
4. To use the SFTP feature, check the Secure FTP box.
5. Double-click the FTP/SFTP command and select the Get File option.
6. In Remote File specify the name of the file on the FTP/SFTP server that you want to download, e.g., `Folder1/Folder2/myfile.doc`.
7. In Local file, select the folder name by clicking the '...' button.
8. If your document is HTML, plain text, or any other ASCII format, select ASCII format, otherwise select binary.
9. Click Save.
10. Double-click the FTP/SFTP command and select Disconnect.

## Example 2: Download your Website.

1. Insert an `FTP/SFTP > Connect` command.
2. Insert an `FTP/SFTP > Get Folder` command. Specify the folder you want to download from your Web server.
3. Specify the Local Folder where you want to download the FTP/SFTP folder.
4. Optionally, specify `*.htm` File Type to download only .htm files.
5. Insert an `FTP/SFTP > Disconnect` command.

## Example 3: Upload your Website.

1. Insert an `FTP/SFTP > Connect` command.

2. Insert an FTP/SFTP > Put Folder command. Specify the folder you want to upload to your Web server.
3. Specify the File Type that you want to upload, e.g., \*.htm.
4. Insert an FTP/SFTP > Disconnect command.

#### **Example 4: Periodically clean your FTP/SFTP folder.**

1. Insert an FTP/SFTP > Connect command.
2. Insert an FTP/SFTP > Delete File(s) command.
3. Specify the File Types you want to delete, e.g., abc.htm, or \*.htm, or abc\*.atmx, or \*.\*, etc.
4. Insert an FTP/SFTP > Disconnect command.

#### **Example 5: Rename files in your FTP/SFTP folder before you upload the new ones.**

1. Insert an FTP/SFTP > Connect command.
2. Insert an FTP/SFTP > Rename File(s) command.
3. Specify the file names you want to rename, e.g., abc.htm, or \*.htm, or abc\*.atmx, or \*.\*, etc.
4. Specify the filename that you want to give to the files, e.g., OLD\*.htm or OLD\*.\* , etc.
5. Insert an FTP/SFTP > Disconnect command.

## **Files Folders Command**

#### **Example 1: Copy files from one folder to another that were created within the last 180 days, and that are 50 Kb or larger.**

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click Files/Folders. A list of Files/Folders actions appears.
3. Double-click Copy Files. The Files/Folders dialog appears, with the Copy Files option selected.

4. In the Source Files text box, type the path and name of a source file to copy; or click Browse and select a file.
5. In the Destination Folder text box, type the name of a destination folder where you want to copy the file.
6. Click Advanced View. The Command Options appear on the right side of the Files/Folder dialog.
7. In the Select Size drop-down, select 'atleast', and in the KB text box type 50.
8. In the Date drop-down, select 'created date'.
9. Select 'in last' and type 180 in the 'days' text box.
10. Click Save.

### **Example 2: Rename all .dat files to .bak**

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click Files/Folders. A list of Files/Folders actions appears.
3. Double-click Rename Files. The Files/Folders dialog appears, with the Rename Files option selected.
4. In the File(s) text box, type the path to the folder that contains the .dat files you want to rename, and specify all .dat files by inserting the '\*' wildcard character. For example: C:\MyFolder\\*.dat. Or click Browse and select a folder, and add '\*.dat' to the folder path.
5. In the New File Name text box, type the filename extension for the renamed files; for example: \*.bak.
6. Click Save.

### **Example 3: Back up your work at the end of the day in a Zip file.**

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click Files/Folders. A list of Files/Folders actions appears.

3. Double-click Zip Files. The Files/Folders dialog appears, with the Zip Files option selected.
4. In the Specify File(s) to Compress text box, type the path to the folder that contains the files you want to compress. For example: C:\MyWork\. Or click Browse and choose a folder.
5. Optionally, in the File Type text box, type the filename extensions for the file types you want Automation Anywhere Client to compress. Separate the filespecs with commas; for example: \*.doc, \*.txt, \*.rtf.
6. In the Specify Destination Zip File text box, type a name for the Zip file that will hold the compressed files.
7. To choose advanced options, click Advanced View. The advanced options appear on the right side of the Files/Folder dialog. Here you can tell Automation Anywhere Client whether to overwrite or move existing files; specify a compression speed; and/or password-protect the resulting Zip file, etc.
8. Click Save.

#### **Example 4: Delete all \*.tmp files at the end of the day.**

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click Files/Folders. A list of Files/Folders actions appears.
3. Double-click Delete Files. The Files/Folders dialog appears, with the Delete Files option selected.
4. In the File(s) Name text box, type the path to the file(s) you want Automation Anywhere Client to delete. For example, to delete all temporary files with a .tmp filename extension in the C:\Temp folder, type: C:\Temp\\*.tmp
5. Click Save.

#### **Example 5: Print all of the Microsoft Word documents in the folder C:\My Documents\.**

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click Files/Folders. A list of Files/Folders actions appears.

3. Double-click Print Multiple Files. The Files/Folders dialog appears, with the Print Files in a Folder option selected.
4. In the Folder Name text box, type the path to the folder; for example: C:\My Documents\.
5. Optionally, in the File Type text box, type the filename extensions for the file types you want Automation Anywhere Client to compress. Separate the filespecs with commas; for example: \*.doc, \*.txt, \*.rtf.
6. Click Save.

#### **Example 6: Create a shortcut to open a file by clicking an icon on the desktop.**

1. In Automation Anywhere Client, open the Task Editor. To open an existing task, select the task in the Task List, and click Edit. To create a new task, click New, or choose Tools > Task Editor.
2. In the Task Editor, in the Commands list, double-click Files/Folders. A list of Files/Folders actions appears.
3. Select the Create File Shortcut option.
4. Specify the location of the file for which you want to create a shortcut. e.g. C:\Temp\DailyPlan.xls.
5. Specify the destination folder where you want to create the shortcut, e.g., C:\Documents and Settings\Administrator\Desktop\.
6. Click Save.

## **Excel Command**

#### **Example 1: Copy the values in a column in an Excel sheet to another sheet.**

1. Open the Task Editor.
2. Double-click the Open Spreadsheet command.
3. Specify a session name, e.g., 'copy cells.'
4. Click '...' and select the Excel spreadsheet from which you want to copy.
5. Click Save.
6. Double-click the Open Spreadsheet command.

7. Specify a session name, e.g., 'paste cells.'
8. Click '...' and select the Excel spreadsheet where you want to paste.
9. Click Save.
10. Double-click the Get Cells command.
11. Specify a session name, e.g., 'copy cells.'
12. Select Get Multiple Cells and insert the range, e.g., if the range is from cell C6 to C18, insert C6 in From Specific Cell and C18 in To Specific Cell.
13. Choose Loop for Each row in an Excel Dataset and set the session name as 'copy cells.'
14. Double-click the Set Cells command.
15. Set the session name as 'paste cells.'
16. In the cell value, select the variable Excel Column and select the column number as 1, since you are retrieving values for one column.
17. Click Save.
18. Double-click the Go to Cells command.
19. Set the session name as 'paste cells.'
20. Select one cell below.
21. Click Save.

#### **Example 2: To copy from Excel into Notepad.**

1. Open the Task Editor.
2. Double-click the Open Spreadsheet command.
3. Click '...' and select the Excel spreadsheet that you want to copy from.
4. Click Save.
5. Click Get Cells.
6. Select Get Multiple Cells and insert the range, e.g., if your range is from cell C6 to C18, insert C6 in From Specific Cell and C18 in To Specific Cell.

7. Start a Loop for Each row in an Excel Dataset.
8. Make sure Notepad is open.
9. Double-click the Insert Keystrokes command.
10. Select the Notepad window title from the drop-down list.
11. Select the variable Excel Column from the list on the right, and insert the column number as 1, since you are retrieving values for one column.
12. Insert a special keystroke of [ENTER] after the \$ExcelColumn(1)\$ variable.
13. Click Save.

#### **Example 3: Run an Excel macro in an Excel spreadsheet.**

1. Open the Task Editor.
2. Double-click the Open Spreadsheet command.
3. Click '...' and select the Excel spreadsheet that contains the macro.
4. Click Save.
5. Double-click the Run Excel Macro command.
6. Type the macro name, e.g., Macro1.
7. Click Save.

#### **Example 4: Delete an entire row in an Excel spreadsheet.**

1. Open the Task Editor.
2. Double-click the Open Spreadsheet command.
3. Click '...' and select the Excel spreadsheet.
4. Click Save.
5. Double-click the Delete Cells command.
6. Specify a cell in the row you want to delete.
7. Select Entire Row from the Delete options.

8. Click Save.

#### **Example 5: Getting values under a specified Column Title.**

1. In the Task Editor double click 'Open Spreadsheet' sub-command.
2. Click  and select the Excel spreadsheet that hypothetically contains table with the heading titles in the first row (refer image):

	A	B	C
1	<b>ID</b>	<b>Name</b>	<b>Designation</b>
2	123	John Smith	Product Manager
3	124	Dwane Williams	Support Lead
4	125	Jeremy McCullum	Software Engineer

3. Click Save.
4. Select Get Cells > Get All Cells option.
5. Select 'Contains Header' check-box to enable the task to fetch required data from the specified column.
6. Save the command.
7. Select Loop > Each row in an Excel dataset and save.
  - This should then populate the Task Editor with the Start Loop, Comment and End Loop lines.
8. Insert a Message Box command within this loop. Press F2 and select the \$Excel Column\$ variable
  - A dialog box will pop-up prompting you to enter the Column Header
9. Enter 'Name' in that text-box and click OK
  - You will notice that the Message Box Command will display \$Excel Column(Name)\$
10. Save the Command.Save and Run the task
  - You will get the values under the 'Name' Column, one by one.

## Error Handling Command

#### **Example 1: To continue extraction of all tables from a website after a table is not found.**

1. Open the Task Editor.
2. Double-click the Web Recorder command.
3. Double-click the Extract Table command.
4. Type the website URL and click Launch.
5. Once the site is open, you will be redirected to the Extract Table command dialog box. Click Capture.

6. The mouse will appear as a yellow background text indicating 'Please click on the table you want to extract.'
7. Move the mouse over the table you want to extract and click anywhere in the table. (If the text background of the mouse changes to red, it means the table cannot be extracted.)
8. Specify the local file path for the CSV file where you want to store the results of the command.
9. Click Save.
10. Double-click the Loop command.
11. Double-click Times.
12. For example, if you want to extract all tables with table number between 1 and 50, specify the loop to run 50 times.
13. Click Save.
14. You need to place the Extract Table command within the above loop.
15. Double-click the Error Handling command.
16. Double-click the Begin Error Handling command.
17. Select the Continue option.
18. Select the Log Data into File check box.
19. Type the local file path for the log file, and the text you want to log.
20. Click Save.
21. The Extract Table command should be within the Begin Error Handling and End Error Handling commands.
22. When you run this command, it extracts all tables with table number from 1 to 50 in the Web page you specified, and stores the results in the CSV file you specified. If a table is not found, the error is logged in the specified log file but the task continues to extract the next table.

## Email Automation Command

*Available in Enterprise and Small Business editions only.*

### **Example 1: Get All Messages from your mail server.**

Open the Task Editor.

Double-click the Email Automation command. Select the Get All Messages option.

Specify the Host Name. ie., your mail server ID.

Specify your User Name and Password.

Select server type. i.e., IMAP or POP3.

Specify the Port Number.

Specify the mail status if IMAP server type is selected. ie., All, Read or Unread.

Specify the message format. It is HTML by default.

Optionally, specify the folder where attachments to be saved. If not specified, attachments will not be downloaded.

Save the command. An Email Automation loop will be generated and you can access Email Automation variables in it.

## **Delay Command**

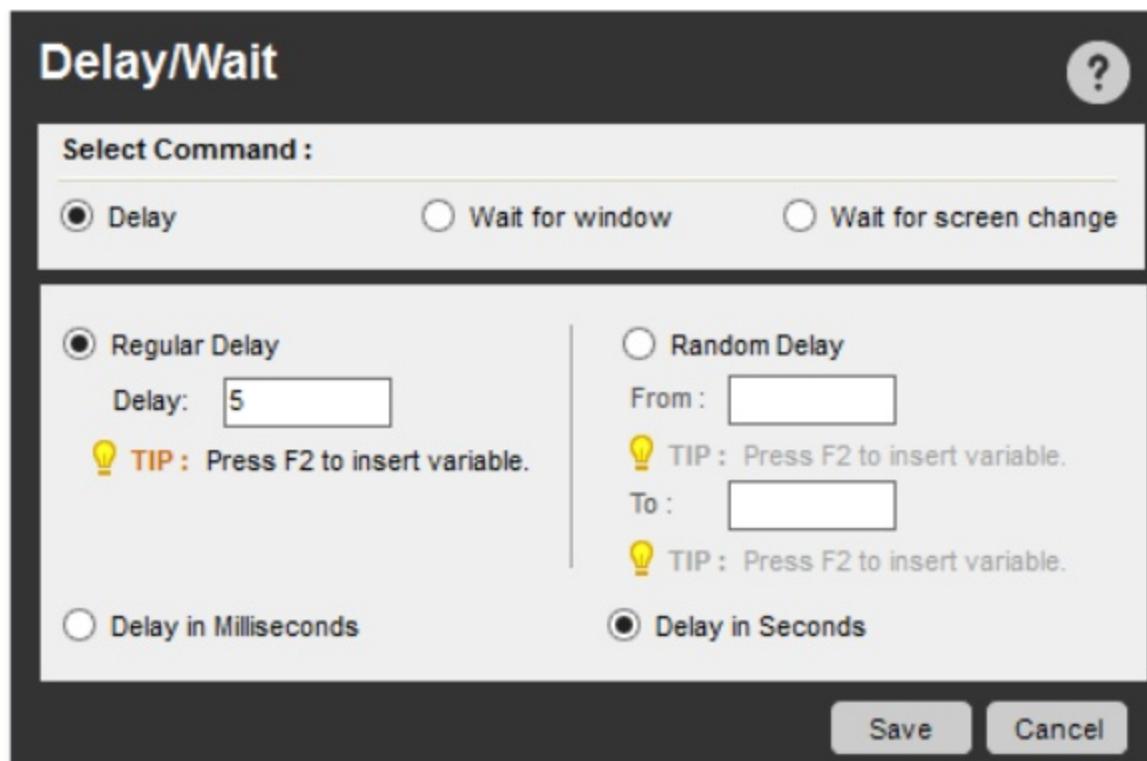
### ***Use Case Scenario 1***

Insert a delay of 5 seconds.

Open the Task Editor.

Double-click Delay.

Select Delay from the invoked window



Type '5' in the Delay text box for the 'Regular Delay' option

Select 'Delay in Seconds'.

Click Save.

### ***Use Case Scenario 2***

Generate random delays to simulate user behavior in a testing scenario.

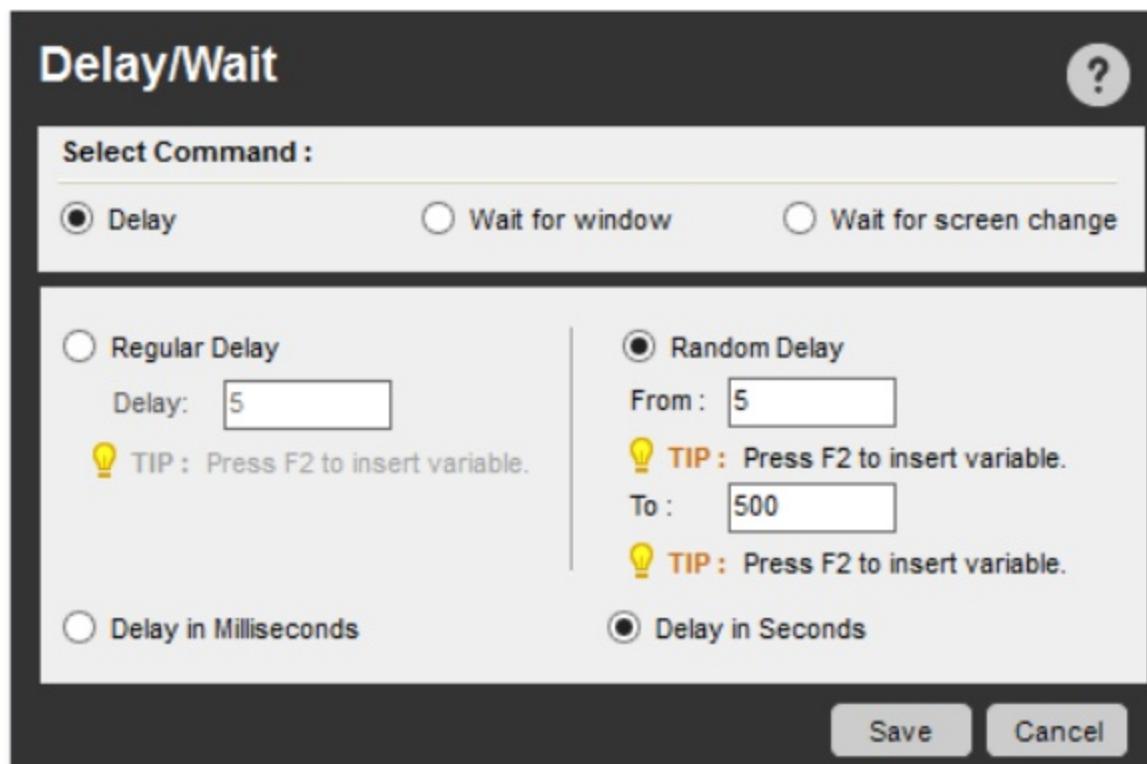
Open the Task Editor.

Double-click the Delay command.

Select Delay.

Select the 'Random Delay' option.

Type 5 in the 'From' text box and '500' in the 'To' text box.



Select 'Delay in Seconds'.

Click Save.



*Tip: When the task runs and Automation Anywhere Client comes to the random delay command, it generates a random number in the range you specified.*

## Database Command

*Available in Enterprise and Small Business editions only.*

### Use Case Scenario 1

Connect to a database and update the records based on user input.

Open the Task Editor.

Double-click the Database command.

Select Connect.

Specify the database you want to connect to.

Test the connection before you save the command.

Similarly, insert a Database -> Insert/Update/Delete command with the appropriate SQL statement.

Insert a Database -> Disconnect command.



*Tip: The Insert/Update/Delete statement can include variables, and the values of the variables can be taken from the user with the Prompt command.*

### **Use Case Scenario 2**

Query all accounts with revenue over \$1 million and print the account names to an Excel file.

Open the Task Editor.

Double-click the Database command.

Select Connect.

Specify the database you want to connect to.

Test the connection before you save the command.

Similarly, insert a Database -> Select Query command with the appropriate SQL statement to get all required accounts.

Insert a Database -> Disconnect command.

To access the result returned by the SQL Query, use the \$Dataset Column\$ system variable. Use 'Loop for Each row in a SQL query dataset' to loop through the rows of the SQL Query result. To access each column, use \$Dataset Column(index)\$; for example, to get the first column, use \$Dataset Column(1)\$.

### **Use Case Scenario 3**

Connect to a database and create tables, drop tables, or alter tables.

Open the Task Editor.

Double-click the Database command.

Select Connect.

Specify the database you want to connect to.

Test the connection before you save the command.

Similarly, insert a Database -> Insert/Update/Delete command with the appropriate SQL statement to create, drop, or alter tables, etc.

Insert a Database -> Disconnect command.

Example 4: Connect to a database and call a procedure stored in the database to update records based on user input.

Open the Task Editor.

Double-click the Database command.

Select Connect.

Specify the database you want to connect to.

Test the connection before you save the command.

Similarly, insert a Database -> Run Stored Procedure command with the procedure name and parameters, if any. The procedure you specify should have already been created in the database.

Insert a Database -> Disconnect command.



*Tip: The procedure name and parameters can have variables, and the values of the variables can be taken from the user with the Prompt command.*

### **Samples SQL Queries**

To query all columns from a table called Table1, use the query Select \* From Table1.

To query FirstName and LastName from a table called Names, use the query Select FirstName, LastName from Names.

To query FirstName and LastName from a table called Names, where FirstName is John, use the query Select FirstName, LastName from Names WHERE FirstName=John.

## **Comment Command**

### **Use Case Scenario**

Insert "Comments" in an automated task.

Open the Task Editor.

Double-click the command for Comment and insert these comments as shown below.

Follow the below steps for the example.



*Note: In the following, 'Comments' explain the command steps for 'Active Directory (AD)'.*

**Comment:** Add a New Organizational Unit under TESTING Domain. i.e. 'SalesOU'

**AD Command:** Create Object 'SalesOU' of type 'Organizational Unit' at "LDAP://sharepoint/DC=TESTING,DC=COM"

## Active Directory

Select Command:

Create User     Create Group     Create Object     Search  
 Modify User     Modify Group     Modify Object     Get Property

Enter LDAP Path:

Parent Path:  ...  
e.g. LDAP://DomainServer/CN=Users;DC=Domain,DC=Com

Login User:  Login Password:

Enter New User Details:

User Name:	<input type="text"/>	Password:	<input type="text"/>
First Name:	<input type="text"/>	Last Name: (Optional)	<input type="text"/>
Display Name: (Optional)	<input type="text"/>	Initials: (Optional)	<input type="text"/>
E-Mail: (Optional)	<input type="text"/>	Description: (Optional)	<input type="text"/>
Department: (Optional)	<input type="text"/>	Title: (Optional)	<input type="text"/>

User must change password at next logon     User cannot change password  
 Is Active     Password never expires

**TIP :** Press F2 to insert variable.

Save    Cancel

**Comment:** Rename the newly added Object. i.e. Change the name from 'SalesOU' to 'Sales Department'

**AD Command:** Rename Object 'SalesOU' To 'Sales Department' at "LDAP://sharepoint/DC=TESTING,DC=COM"

## Active Directory



### Select Command:

- Create User     Create Group     Create Object     Search  
 Modify User     Modify Group     Modify Object     Get Property

### Enter LDAP Path:

Parent Path:  ...

e.g. LDAP://DomainServer/CN=Users;DC=Domain,DC=Com

Login User:

Login Password:

### Select Object Action:

Select Action:

### Enter Modify Object Details:

Object Name:

New Name:

Object Type:  Computer

Organizational Unit

**TIP:** Press F2 to insert variable.

**Save**

**Cancel**

**Comment:** Add a Group under Sales OU. i.e. Tier1 Customer

**AD Command:** Create Group 'Tier1 Customer' at "LDAP://sharepoint/OU=SalesDepartment,DC=TESTING,DC=COM"

## Active Directory



### Select Command:

- Create User       Create Group       Create Object       Search  
 Modify User       Modify Group       Modify Object       Get Property

### Enter LDAP Path:

Parent Path:



e.g. LDAP://DomainServer/CN=Users;DC=Domain,DC=Com

Login User:

Login Password:

### Enter New Group Details:

Group Name:

Description: (Optional)

Group Scope:  Domain Local       Global       Universal

Group Type:  Security       Distribution

**TIP:** Press F2 to insert variable.

Save

Cancel

**Comment:** Add a new User into the Group. i.e. John Smith with logon name as 'S001'

**AD Command:** Create User 'S001' at "LDAP://sharepoint/OU=SalesDepartment,DC=TESTING,DC=COM"

## Active Directory



### Select Command:

- Create User     Create Group     Create Object     Search  
 Modify User     Modify Group     Modify Object     Get Property

### Enter LDAP Path:

Parent Path:

e.g. LDAP://DomainServer/CN=Users;DC=Domain,DC=Com

Login User:

Login Password:

### Enter New User Details:

User Name:

Password:

First Name:

Last Name: (Optional)

Display Name: (Optional)

Initials: (Optional)

E-Mail: (Optional)

Description: (Optional)

Department: (Optional)

Title: (Optional)

User must change password at next logon

User cannot change password

Is Active

Password never expires

**TIP:** Press F2 to insert variable.

Save

Cancel

**Comment:** Add the newly created User (S001/John Smith) under Group named 'Tier1 Customer'

**AD Command:** Add User(s) to Group 'Tier1 Customer' at "LDAP://sharepoint/OU=SalesDepartment,DC=TESTING,DC=COM"

## Clipboard Command

*Available in Enterprise and Small Business editions only.*

This command is useful when you are transferring data using Ctrl + C (copying) from one application and Ctrl + V (pasting) into another application; or when you are performing these operations within a loop. Sometimes there is empty data in an application (e.g., Excel cells), and the copied data needs to be cleared after the previous iteration.



*Tip:*

During a data transfer task, use the Clear Clipboard command before each iteration.

Before the End Loop command, insert a Clear Clipboard command.

#### ***Use Case Scenario 1***

Assign the value of a variable to the Clipboard so that the value can be directly pasted using Ctrl + V after the task completes.

Open the Task Editor.

Double-click the Clipboard command.

Double-click 'Assign To Clipboard.'

Select the variable from the drop-down list whose value you want to assign to the Clipboard.

Click Save.

At the end of the task, if you want to move some data into the Clipboard, you can use this command so that as soon as the task finishes you can simply use Ctrl + V to paste the data into the desired application (e.g., an email client).

#### ***Use Case Scenario 2***

Compare two cells of an Excel document by assigning the value of the Clipboard to a variable.

Open the Task Editor.

Double-click the Clipboard command.

Double-click Assign From Clipboard.

Select the variable Cell1 from the drop-down list.

You can assign values from the Clipboard only to Value type variables, not to List or Random type variables.

Click Save.

Similarly, assign the value of the Clipboard (after performing a Ctrl + C operation) to another variable, Cell2.

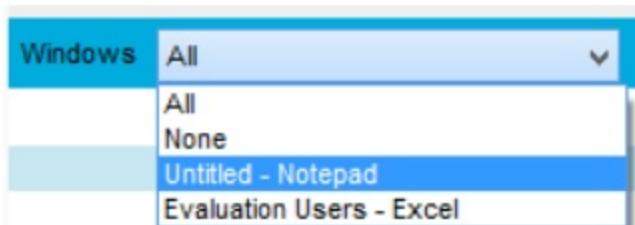
Perform the comparison operations on variables Cell1 and Cell2 as needed.

# Change Window Title Command

Set a common Window title to multiple Window actions.

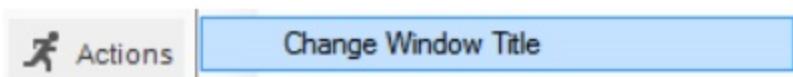
Open the Task Editor.

Select from drop down of Windows filter to select a particular Window.

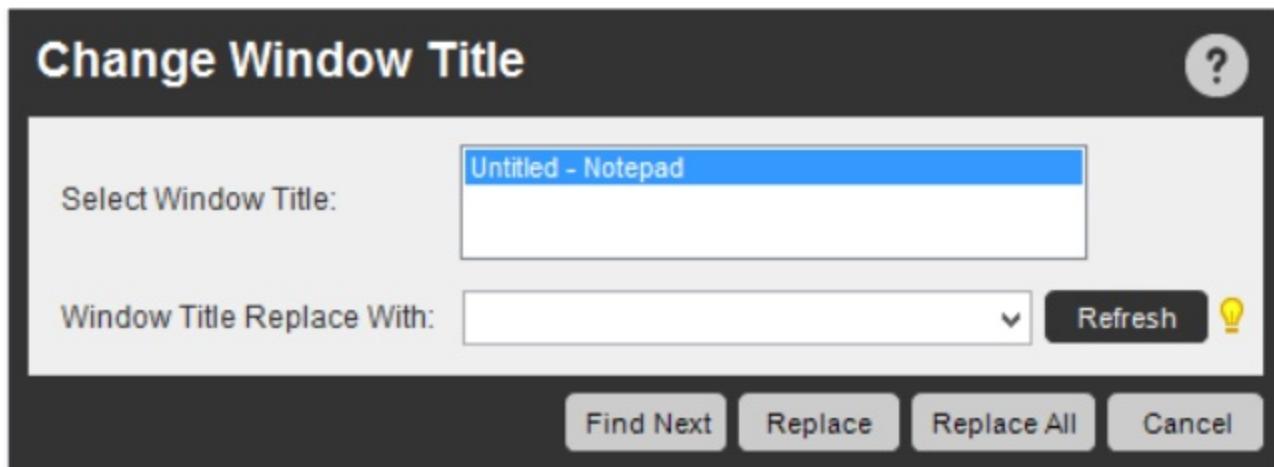


Select the specific commands which contain windows to edit their Window Titles.

Select 'Change Window Title' from Actions.



In the ensuing Window, type a Window title with which you wish to replace selected Window titles.



Use 'Find' and 'Replace' or 'Replace All' to browse and replace the selected Window title.

Click Save.

# App Integration Command

*Available in Enterprise and Small Business editions only.*

Capture the text from any control in an application and set the status of a check box depending on the captured text.

### **Use - Case Scenario**

For instance, you want to select the check box 'Send me information on product updates' on the 'Ask the Experts' page of the Automation Anywhere website, you can do the following:

Open the application (in this case Browser) from which you want to capture text data.

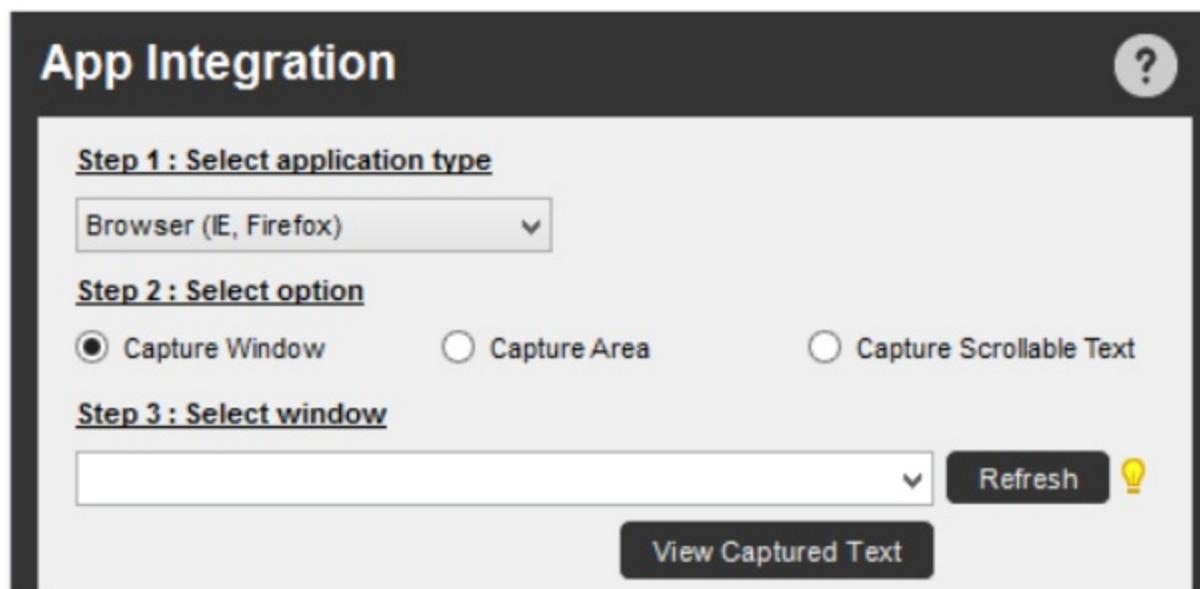
Open the Task Editor in Automation Anywhere Client.

Double-click the App Integration command.

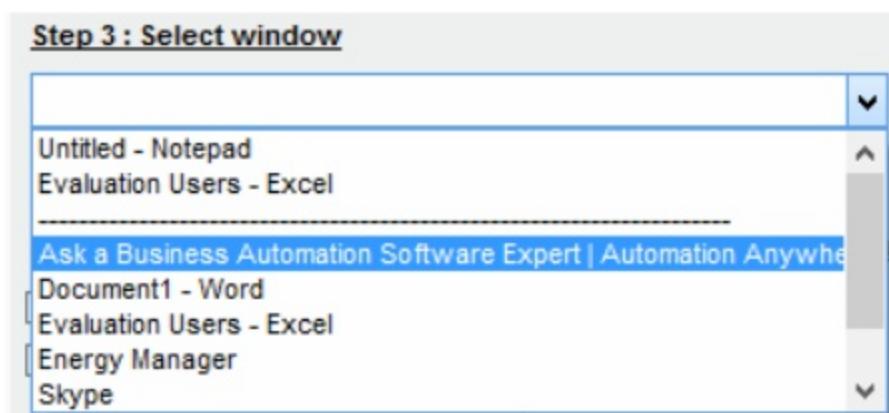
Drag and drop the Window Application option

The App Integration window is invoked.

Select the application type and options through steps 1 and 2:



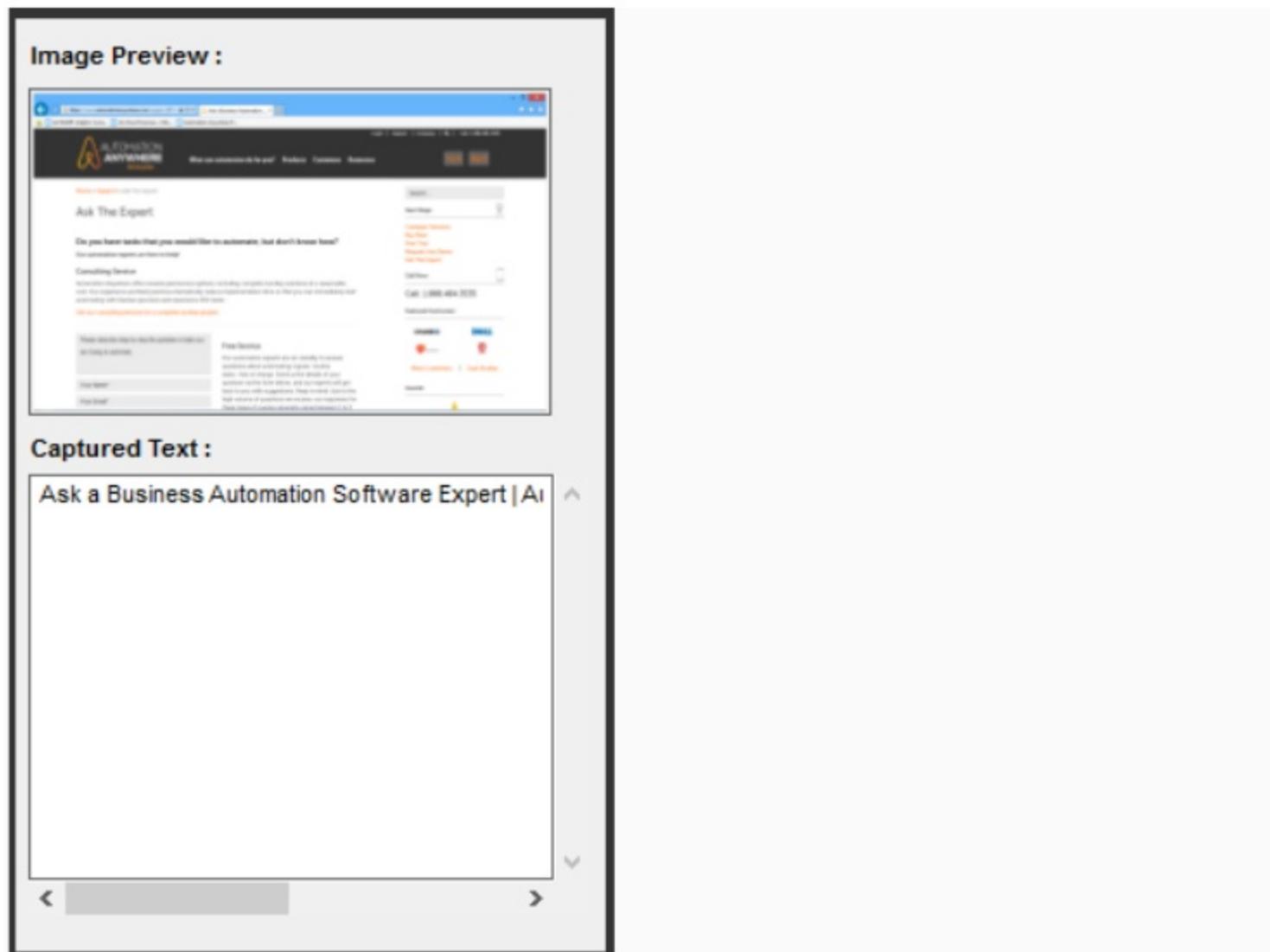
Select the required application window from the drop-down list.



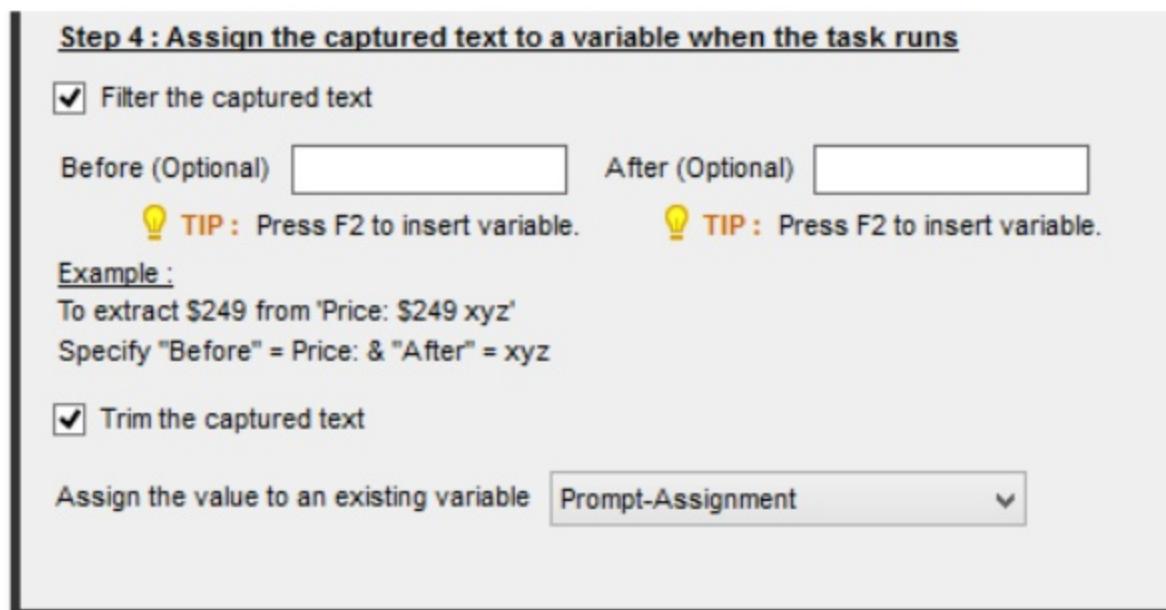
Click View Captured Text. This will generate the image preview on the right side, together with the captured text.

**View Captured Text**

The program provides a preview of the captured image and text (in this case the window title) in an extended panel:



If you need to filter the captured text, check the 'Filter the captured text' check-box and/or optionally specify 'Before' and 'After' keywords in step 4.

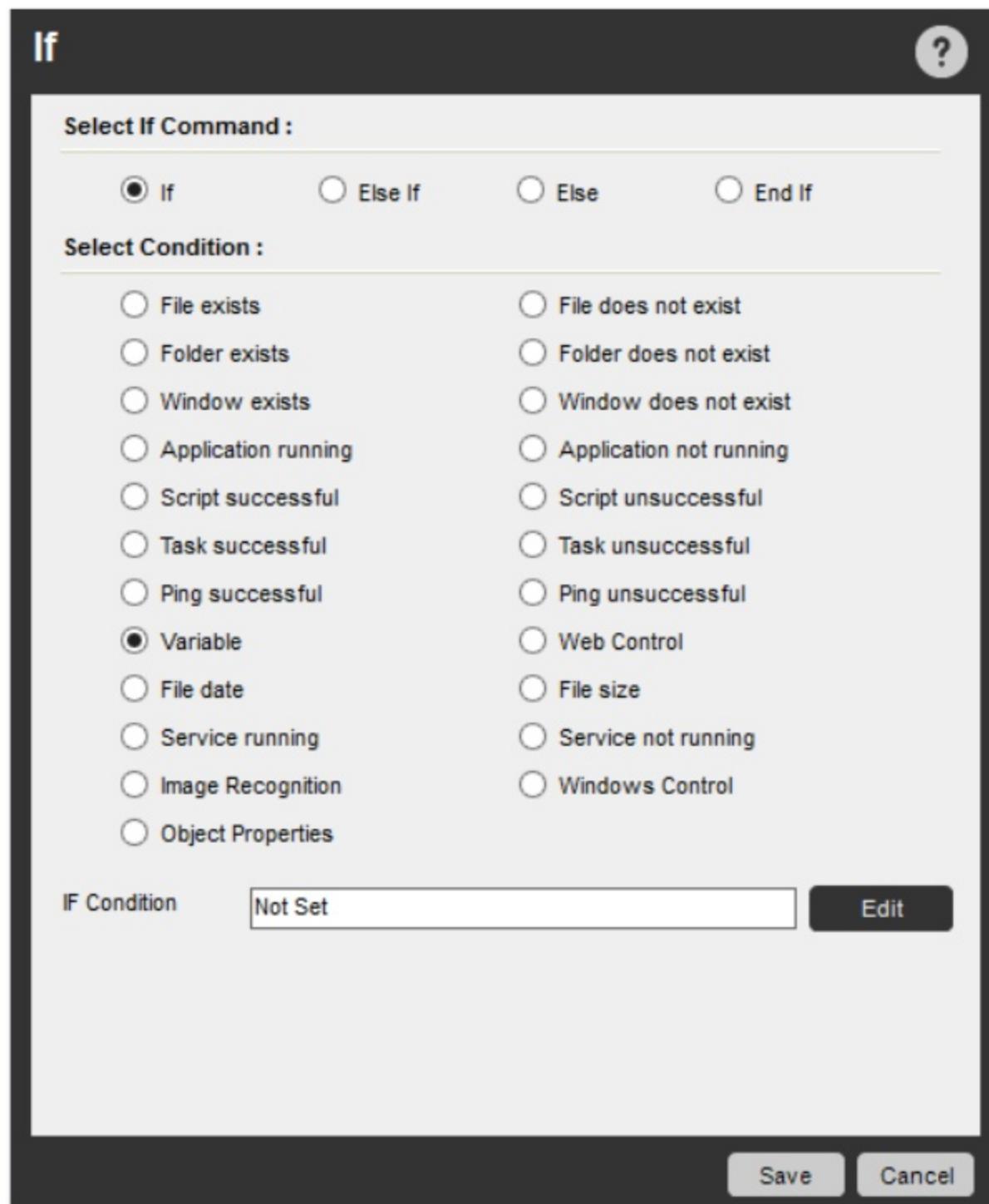


Assign the value to a variable from the list.

Click Save.

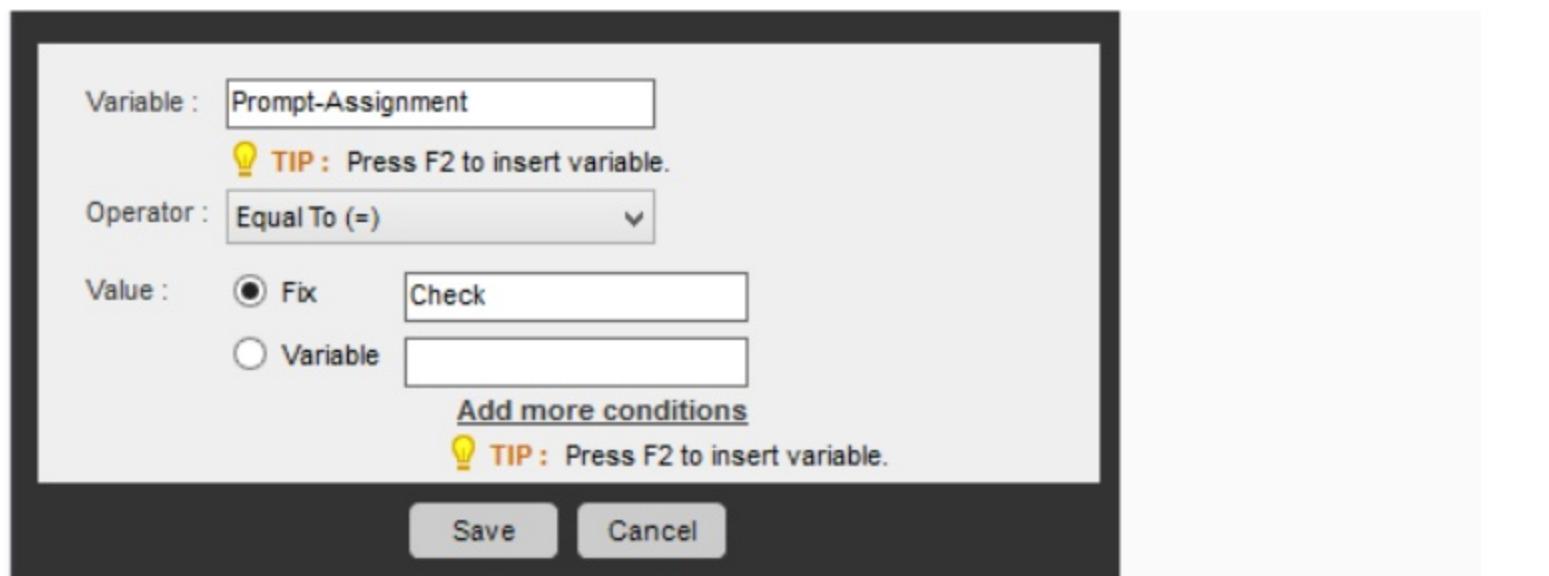
 Note: The next step involves assigning a value to the variable and then setting the status of the check - box to ' (Checked)'. For this to work, you need to use the '[If-Else](#)' and '[Manage Windows Controls](#)' commands.

Double-click the If/Else command, and drag-drop the option 'Variable'.



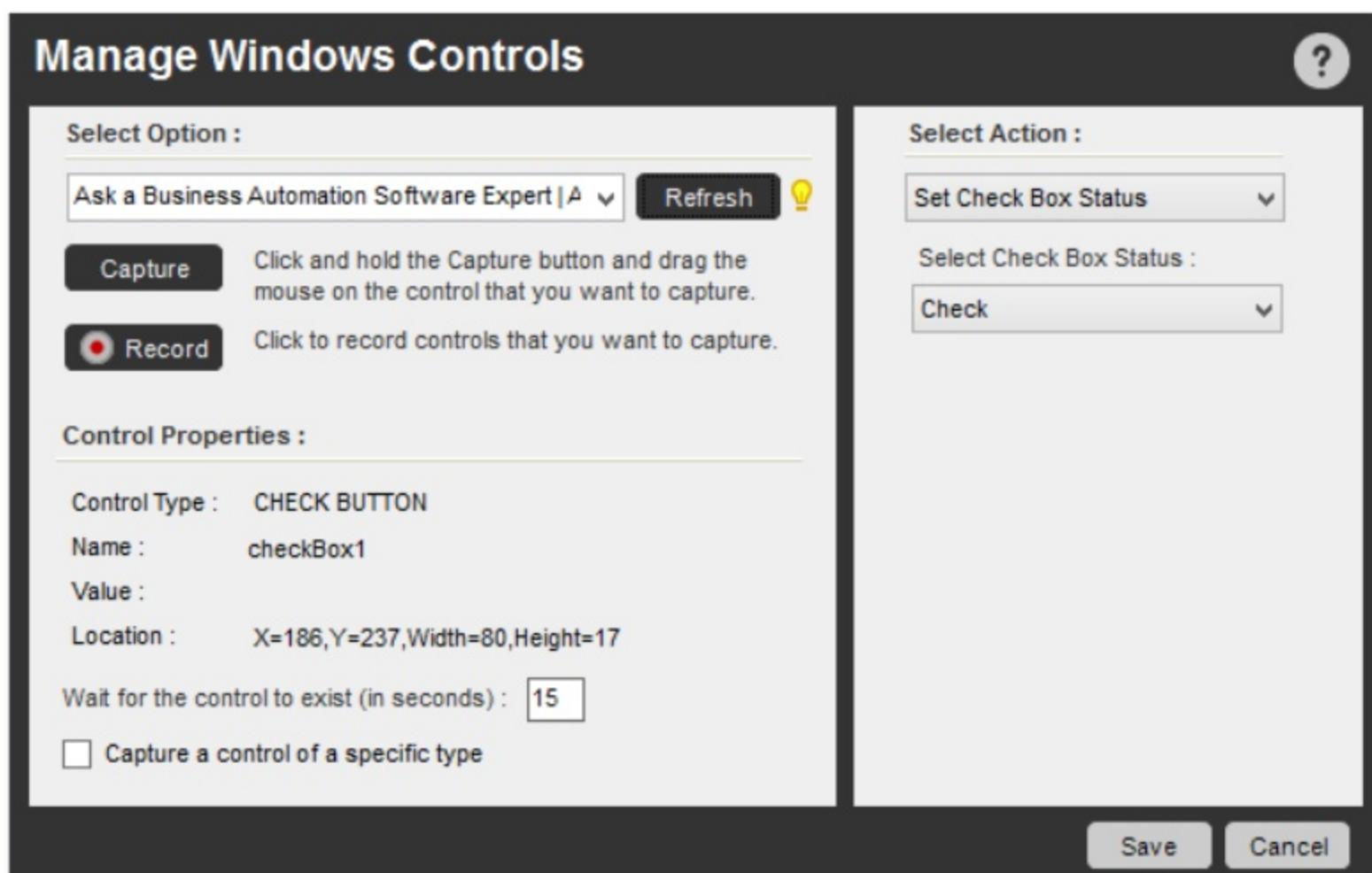
Click Edit and select the variable you specified in Step 6.

Select Equal To(= ), select Fix, and specify the value you want to check as shown:



Double-click the Manage Windows Controls command, and double-click Check Box.

Select the application window from the drop-down list. Click Capture and hold and drag the mouse to the control you wish to capture (here it is 'Control Box'). The 'Manage Windows Controls' windows displays the details as follows:



Set the 'Check Box Status' and Click Save.

Save the task.



*Note: Your task should resemble the following image:*

```
1  [App Integration] Capture text from window "Ask the Automation Expert. Automation Anywhere - The Leading Automation Software - Windows Internet Explorer" and assign to $Prompt-Assignment
2  [If] If $Prompt-Assignment$ Equal To (=) "Check" Then
3  [Comment] Comment: This condition will assign the value 'Check' to the captured control.
4  [End If]
5  [Set Status] Set Status of CheckBox in Window: "Ask the Automation Expert. Automation Anywhere - The Leading Automation Software - Windows Internet Explorer" Value: "Check"
```

## Using Wild Cards

You can specify wild cards (\*) in a window title.

For example, use this command to perform an operation in any window that has the word "Untitled" anywhere in its title.

To tell Automation Anywhere Client to perform the operation in a window with any text string before or after the word "Untitled" in the window title, simply specify \*Untitled\* for the window title.

Automation Anywhere Client will first search for the exact window ("Untitled"), and if it doesn't find it, it will look for windows with the word "Untitled" anywhere in the title.

You can put the wild card character (\*) at the beginning or end of the window title.

## Bulk Edit Command

### *Use Case Scenario 1*

Set a common delay of 5 seconds to multiple delay actions having varying delay criteria.

Open the Task Editor.

Select multiple 'Delay' actions or use the Filters to view all Delay actions in the Task.

OCR\_Win7

## Task Actions List

NORMAL VISUALIZE

FILTERS  Mouse Moves  Keystrokes  Mouse Clicks  Delays

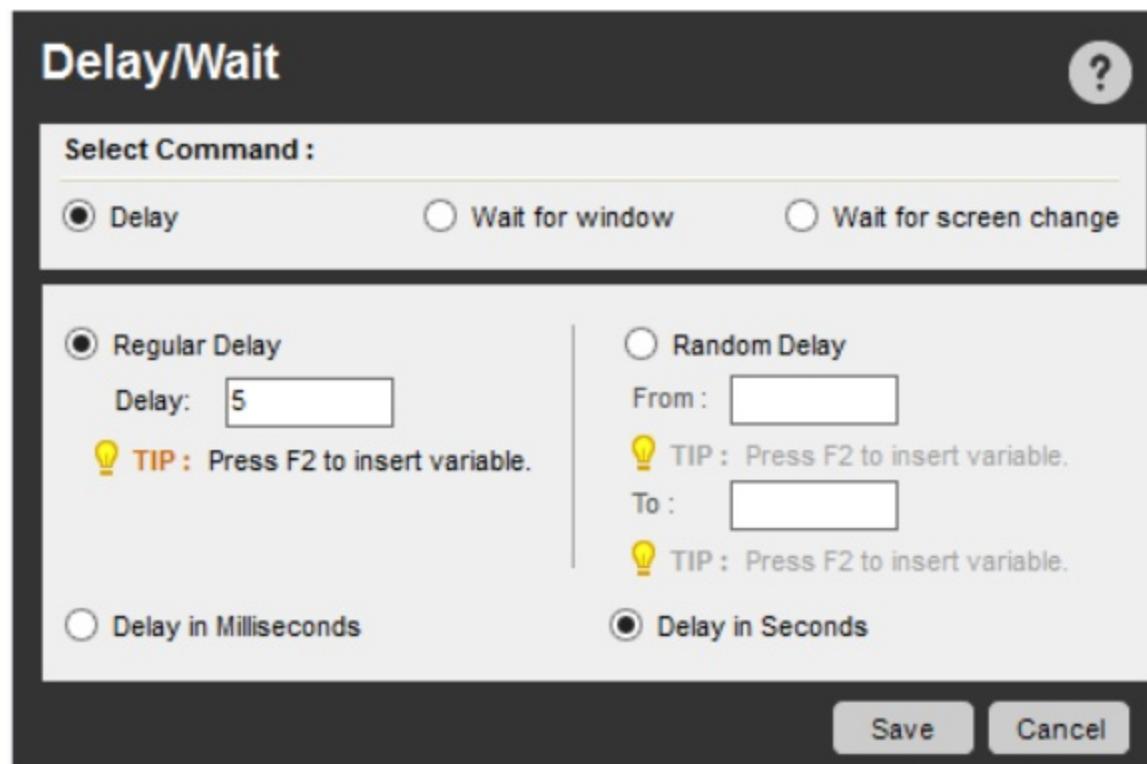
5	Delay: (1 sec)
12	Delay: (500 ms)
16	Delay: (500 ms)
24	Delay: (500 ms)
38	Delay: (500 ms)
52	Delay: (500 ms)
66	Delay: (500 ms)
80	Delay: (500 ms)
94	Delay: (500 ms)
103	Delay: (500 ms)
105	Delay: (500 ms)
109	Delay: (200 ms)

Select all 'Delay' commands

Select 'Bulk Edit' from Actions drop-down or right click and select 'Bulk Edit'.



The Delay/Wait Command window is invoked.



Type 5 in the Delay text box.

Select Delay in Seconds.

Click Save.

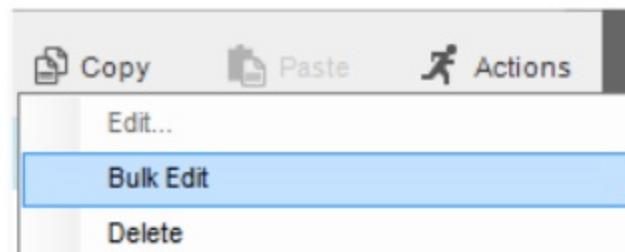
### ***Use Case Scenario 2***

Set a common random delay to multiple 'Delay' actions having varying delay criteria.

Open the Task Editor.

Select multiple 'Delay' actions in the task.

Select Bulk Edit from Actions.

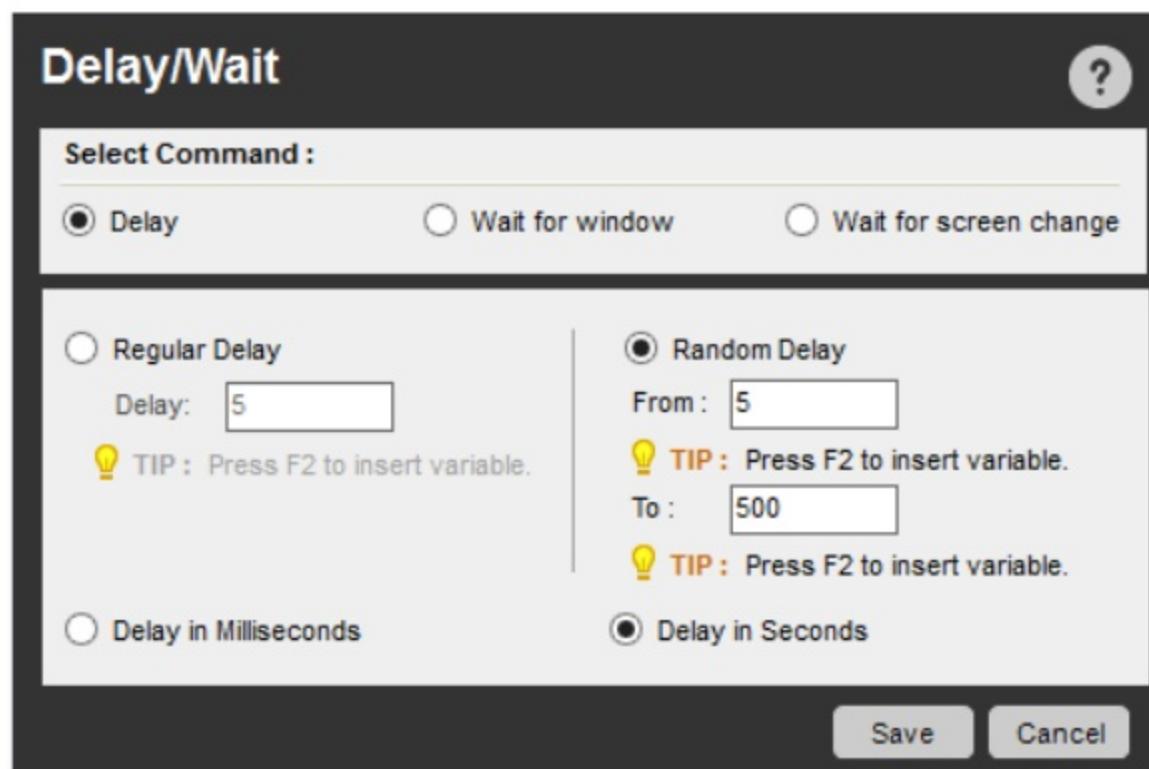


The Delay/Wait Command window is invoked.

Select Random Delay.

Type 5 in the 'From' text box.

Type 500 in the 'To' text box.



Select Delay in Seconds.

Click Save.

## Active Directory Command

*Available in Enterprise edition only.*

### Use Case Scenario

Create a new Organizational Unit and User. Add the User to a Group in Active Directory

Open the Task Editor.

Double-click/drag and drop the 'Create User' command under the Active Directory tree from the commands list.

## Active Directory



### Select Command:

- Create User     Create Group     Create Object     Search  
 Modify User     Modify Group     Modify Object     Get Property

### Enter LDAP Path:

Parent Path:  ...

e.g. LDAP://DomainServer/CN=Users;DC=Domain,DC=Com

Login User:

Login Password:

### Enter New User Details:

User Name:

Password:

First Name:

Last Name: (Optional)

Display Name: (Optional)

Initials: (Optional)

E-Mail: (Optional)

Description: (Optional)

Department: (Optional)

Title: (Optional)

User must change password at next logon

User cannot change password

Is Active

Password never expires

**TIP:** Press F2 to insert variable.

Save

Cancel

Enter details in the LDAP Path section by clicking the browse button:



Note: Ensure that the AD connection details are provided correctly for establishing a valid LDAP path.

Follow the steps for the example.



Note: Comments explain the command steps for Active Directory.

Ensure that the Organizational Unit of Sales Department does not exist in the Active Directory structure.

**Comment:** Add a New Organizational Unit under TESTING Domain. i.e. 'SalesOU'

**ActiveDirectory:** Create Object 'SalesOU' of type 'Organizational Unit' at "LDAP://sharepoint/DC=TESTING,DC=COM"

## Active Directory

Select Command:

Create User     Create Group     Create Object     Search  
 Modify User     Modify Group     Modify Object     Get Property

Enter LDAP Path:

Parent Path:  ...  
e.g. LDAP://DomainServer/CN=Users;DC=Domain,DC=Com

Login User:  Login Password:

Enter New User Details:

User Name:	<input type="text"/>	Password:	<input type="text"/>
First Name:	<input type="text"/>	Last Name: (Optional)	<input type="text"/>
Display Name: (Optional)	<input type="text"/>	Initials: (Optional)	<input type="text"/>
E-Mail: (Optional)	<input type="text"/>	Description: (Optional)	<input type="text"/>
Department: (Optional)	<input type="text"/>	Title: (Optional)	<input type="text"/>

User must change password at next logon     User cannot change password  
 Is Active     Password never expires

**TIP :** Press F2 to insert variable.

Save    Cancel

**Comment:** Rename the newly added Object. i.e. Change the name from 'SalesOU' to 'Sales Department'

**Active Directory:** Rename Object 'SalesOU' To 'Sales Department' at "LDAP://sharepoint/DC=TESTING,DC=COM"

## Active Directory



### Select Command:

- Create User     Create Group     Create Object     Search  
 Modify User     Modify Group     Modify Object     Get Property

### Enter LDAP Path:

Parent Path:  ...

e.g. LDAP://DomainServer/CN=Users;DC=Domain,DC=Com

Login User:

Login Password:

### Select Object Action:

Select Action:



### Enter Modify Object Details:

Object Name:

New Name:

Object Type:  Computer

Organizational Unit

**TIP :** Press F2 to insert variable.

**Save**

**Cancel**

**Comment:** Add a Group under Sales OU. i.e. Tier1 Customer

**Active Directory:** Create Group 'Tier1 Customer' at "LDAP://sharepoint/OU=SalesDepartment,DC=TESTING,DC=COM"

## Active Directory



### Select Command:

- Create User       Create Group       Create Object       Search  
 Modify User       Modify Group       Modify Object       Get Property

### Enter LDAP Path:

Parent Path:

e.g. LDAP://DomainServer/CN=Users;DC=Domain,DC=Com

Login User:  Login Password:

### Enter New Group Details:

Group Name:

Description: (Optional)

Group Scope:  Domain Local       Global       Universal

Group Type:  Security       Distribution



TIP: Press F2 to insert variable.

Save

Cancel

**Comment:** Add a new User into the Group. i.e. John Smith with logon name as 'S001'

**ActiveDirectory:** Create User 'S001' at "LDAP://sharepoint/OU=SalesDepartment,DC=TESTING,DC=COM"

## Active Directory



### Select Command:

- Create User       Create Group       Create Object       Search  
 Modify User       Modify Group       Modify Object       Get Property

### Enter LDAP Path:

Parent Path:  ...

e.g. LDAP://DomainServer/CN=Users;DC=Domain,DC=Com

Login User:

Login Password:

### Enter New User Details:

User Name:	<input type="text" value="JS001"/>	Password:	<input type="password" value="*****"/>
First Name:	<input type="text" value="John"/>	Last Name: (Optional)	<input type="text" value="Smith"/>
Display Name: (Optional)	<input type="text" value="JSmith"/>	Initials: (Optional)	<input type="text" value="JS"/>
E-Mail: (Optional)	<input type="text" value="automationanywhere.com"/>	Description: (Optional)	<input type="text"/>
Department: (Optional)	<input type="text" value="Sales"/>	Title: (Optional)	<input type="text" value="Sales Manager"/>

User must change password at next logon

User cannot change password

Is Active

Password never expires

**TIP:** Press F2 to insert variable.

Save

Cancel

**Comment:** Add the newly created User (S001/John Smith) under Group named 'Tier1 Customer'

**Active Directory:** Add User(s) to Group 'Tier1 Customer' at "LDAP://sharepoint/OU=SalesDepartment,DC=TESTING,DC=COM"



*Tip: You can use the 'Disable' option for other commands, if you wish to test a specific command.*