Open2Test

QTP Open Source Test Automation Framework Scripting Standards for Windows

Version 1.0

April 2009

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF FIGURES

1. Introduction

1.1. Purpose

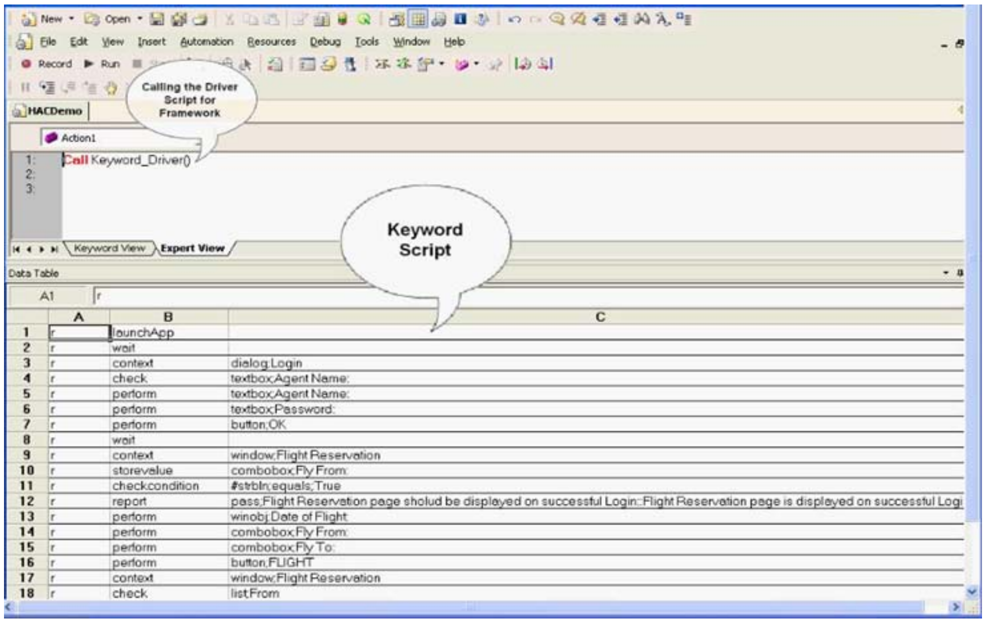
This document provides details about the various columns used, the keywords and their descriptions, along with some methodologies that need to be followed while scripting using keywords.

2. Standards for Keyword Scripting

2.1. Before Starting

Before going into the details about the columns used for keyword scripting, the user should be familiar with what is known as the 'keyword script' and how to call the framework from the test script.

As shown in the figure below, the keyword script is the actual automation test script that corresponds to the manual test case. It is written in the global sheet of the tool. In the 'Expert View' of the tool, the framework is called using the command 'Call Keyword_Driver()'.



```
Call Keyword_Driver()
```

A1 r

	A	B	C
1	r	launchApp	
2	r	wait	
3	r	context	dialog;Login
4	r	check	textbox;Agent Name:
5	r	perform	textbox;Agent Name:
6	r	perform	textbox;Password:
7	r	perform	button;OK
8	r	wait	
9	r	context	window;Flight Reservation
10	r	storevalue	combobox;Fly From:
11	r	checkcondition	#strbln;equals;True
12	r	report	pass;Flight Reservation page sholud be displayed on successful Login::Flight Reservation page is displayed on successful Logi
13	r	perform	winobj;Date of Flight:
14	r	perform	combobox;Fly From:
15	r	perform	combobox;Fly To:
16	r	perform	button;FLIGHT
17	r	context	window;Flight Reservation
18	r	check	list;From

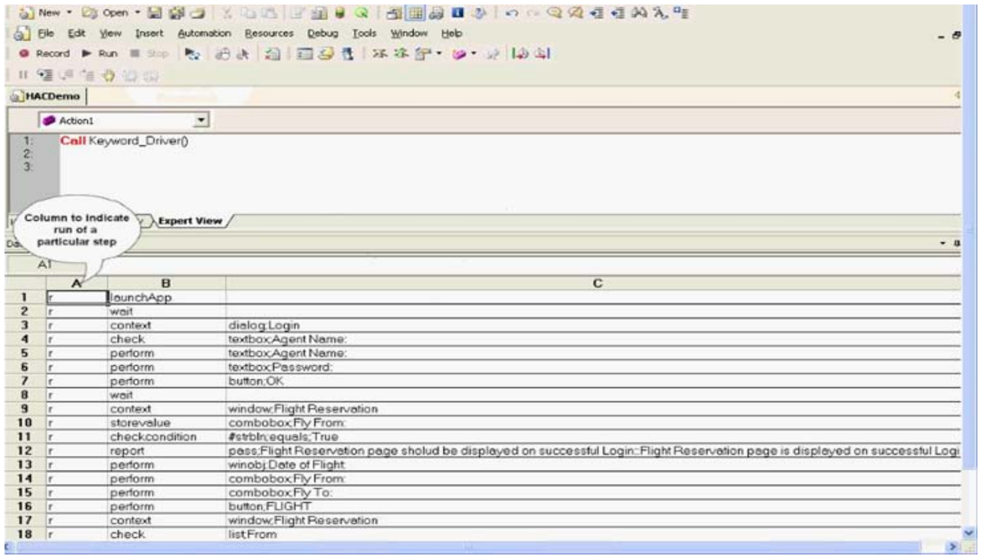
Figure 1: Keyword Script and Calling the Framework

2.2. Column Description

This section gives a description of the columns used for keyword scripting.

2.2.1. Automate (Column 'A')

The data in the 'Automate' column decides whether the current step in the test case is to be run (automated) or not. This column has the letter 'r', which denotes that the current step should be run. If any step in the test case is not being run then the corresponding row in the first column is to be left blank. The steps will run only based on the data in this column.



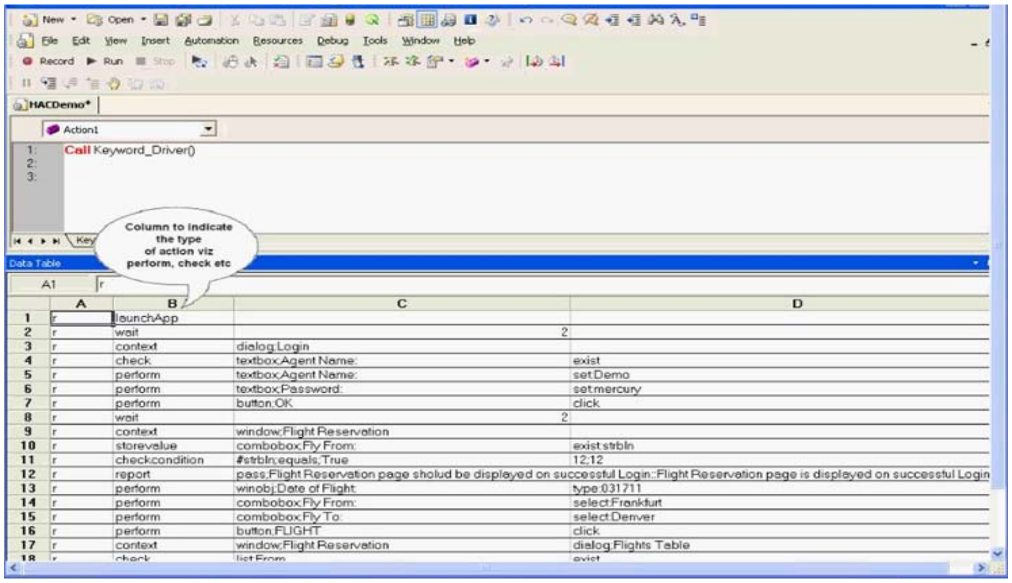
```
Call Keyword_Driver()
```

	A	B	C
1	r	launchApp	
2	r	wait	
3	r	context	dialog;Login
4	r	check	textbox;Agent Name:
5	r	perform	textbox;Agent Name:
6	r	perform	textbox;Password:
7	r	perform	button;OK
8	r	wait	
9	r	context	window;Flight Reservation
10	r	storevalue	combobox;Fly From:
11	r	checkcondition	#strbIn;equals;True
12	r	report	pass;Flight Reservation page sholud be displayed on successful Login::Flight Reservation page is displayed on successful Logi
13	r	perform	winobj;Date of Flight
14	r	perform	combobox;Fly From:
15	r	perform	combobox;Fly To:
16	r	perform	button;FLIGHT
17	r	context	window;Flight Reservation
18	r	check	list;From

Figure 2: Column 'Automate'

2.2.2. Action (Column 'B')

The second column of the global sheet is used to indicate the generic type of action being performed on the application under test (AUT). The action column is dedicated to different types of actions that are to be performed on a particular object.


HACDemo*
Action1
Column to indicate the type of action viz perform, check etc
Key
Data Table
A1

```
Call Keyword_Driver()
```

	A	B	C	D
1	r	launchApp		
2	r	wait	2	
3	r	context	dialog;Login	
4	r	check	textbox;Agent Name:	exist
5	r	perform	textbox;Agent Name:	set;Demo
6	r	perform	textbox;Password:	set;mercury
7	r	perform	button;OK	click
8	r	wait	2	
9	r	context	window;Flight Reservation	
10	r	storevalue	combobox;Fly From:	exist;strbIn
11	r	checkcondition	#strbIn;equals;True	12;12
12	r	report	pass;Flight Reservation page sholud be displayed on successful Login::Flight Reservation page is displayed on successful Login	
13	r	perform	winobj;Date of Flight	type;031711
14	r	perform	combobox;Fly From:	select;Frankfurt
15	r	perform	combobox;Fly To:	select;Denver
16	r	perform	button;FLIGHT	click
17	r	context	window;Flight Reservation	dialog;Flights Table
18	r	check	list;From	exist

Figure 3: Column 'Action'

The keywords that can be used in this column are:-

1. **LaunchApp**

   'LaunchApp' is used to launch the AUT. This keyword triggers the driver script to launch the application either from a specified folder (the location specified in the third column) or if the application is already synchronized with HP QuickTest Professional (QTP) then this automatically launches the application from the location specified in QTP.

2. **Context**

   'Context' is used only on a window object or a dialog object. This keyword brings a particular Window or dialog to the current context, so that any operation or checking can be performed on that particular window or dialog.

3. **Perform**

   'Perform' is used to perform an operation on a particular object such as clicking on a button, closing an open window, typing some text in a textbox, etc. This keyword should be entered in the corresponding row in the second column if any such operations are to be performed.

4. **Check**

   'Check' is used to check if the required property of a particular object is attained at runtime. This is a type of validation step (expected result).

5. **Condition**

   'Condition' provides a feature for comparing two variables, checking properties, checking for the existence of windows, etc.

6. **CallFunction**

   'Call Function' is used to call any declared function that is use, in a particular script. These functions should be declared in a different .vbs file.

7. **Storevalue**

   'Storevalue' is used to store the property values of different objects in different environment variables. These environment variables can later be used as input parameters in various functions and also in scripts.

8. **PressKey**

   'Press Key' is used to pass hot keys such as Enter, F3, F10, Ctrl-S, etc.

9. **Msgbox**

   'Msgbox' is used for debugging to display the contents of a variable.

10. **Report**

    'Report' is used for customized reporter events. It is displayed in the result sheet. The report can be of 4 types: i) Pass, ii) Fail, iii) Done, iv) Warning.

11. **Strsearch**

    'Strsearch' is used to search for a 'sub string' inside a 'main string'.

12. **Strreplace**

    'Strreplace' is used for replacing a 'sub string' inside a 'main string' with a new 'sub string'.

13. **Strconcat**

    'Strconcat' is used to concatenate any number of strings with each other.

14. **Wait**

    'Wait' is used to place static waits in the keyword script.

15. **Arith**

    'Arith' is used to perform the arithmetic operations on the variables.

16. **Assignvalue**

    'Assignvalue' is used to assign dynamically generated values from the application to environment variables.

17. **Callaction**

    'Callaction' is used to call reusable actions that are declared in the script.

18. **Loop**

    'Loop' is used to loop a set of actions given in the datatable.

19. **convert**

    'convert' is used to typecast from one data type to another.

20. **Function**

    'Function' is used to perform FSO (File system Object) operations such as creating a folder in a specified path, creating a file in a specified path, etc.

21. **Importdata**

    'Importdata' is used to import the external test data sheet into the Action1 sheet of the QTP.

A detailed description of the keywords is provided in the Keyword Reference Document.

2.2.3. Object (Column 'C')

The third column of the global sheet is used to indicate the object on which a particular type of action is to be performed. When the object is present in the object repository, the object class and object name are specified in column C (as shown in example 1).

However, if the object is not added to the repository, descriptive programming can be used by specifying any property and its value (as shown in example 2). The object column or column 'C' contain all the required details for an object (viz. Class to which the objects belong to and the object name) on which various operations and validations are to be performed.

Example 1:

Action	Object
Perform	Tab;OK
Perform	Textbox; Lastname

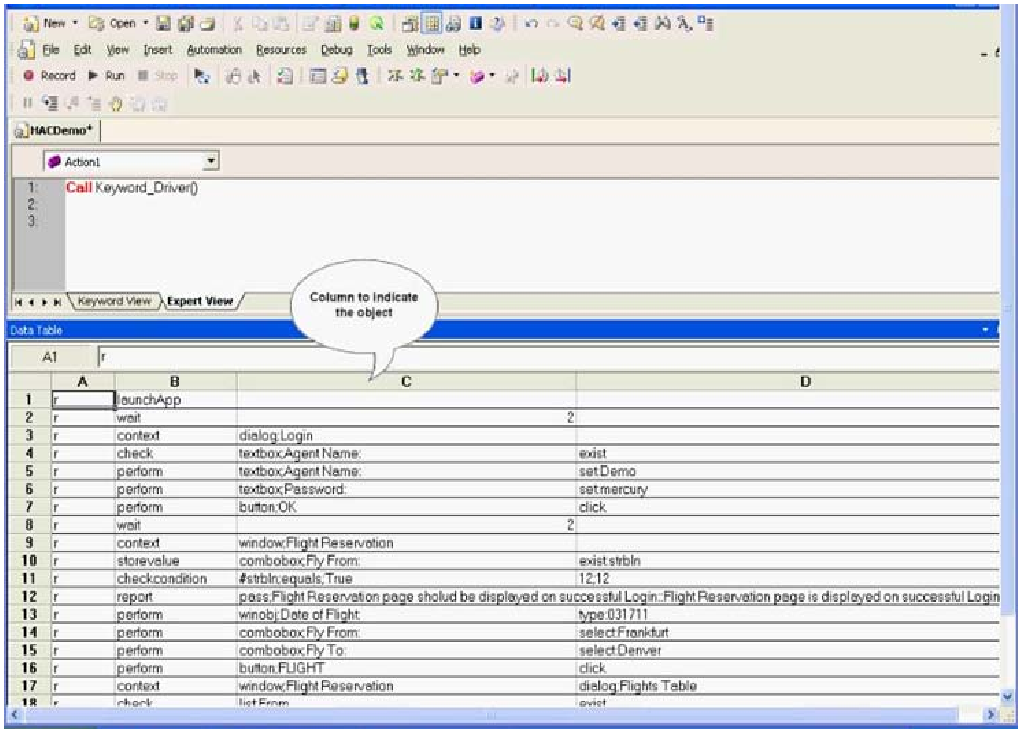
In the above example, the object column indicates that some operation has to be performed on an object of class 'WinTab' having the name 'OK'. Similarly, in the next line some operation has to be performed on an object of class 'Textbox' having the name 'Lastname'.

Example 2:

Action	Object
Perform	Tab;text:=OK
Perform	Textbox;name:=Lastname

In the above example, the following method is used when the object is not added to the object repository. Some operation has to be performed on an object of class WinTab having a property 'text', the value of which is 'OK'. Similarly, some operation has to be performed on an object of class 'Textbox' having a property 'name', the value of which is 'Lastname'.

The object and its name are usually separated by a delimiter ';' as shown in the above example. (Delimiters will be covered in a later topic).



```
Call Keyword_Driver()
```

A1 r

	A	B	C	D
1	r	launchApp		
2	r	wait	2	
3	r	context	dialog;Login	
4	r	check	textbox;Agent Name:	exist
5	r	perform	textbox;Agent Name:	set;Demo
6	r	perform	textbox;Password:	set;mercury
7	r	perform	button;OK	click
8	r	wait	2	
9	r	context	window;Flight Reservation	
10	r	storevalue	combobox;Fly From:	exist;strbIn
11	r	checkcondition	#strbIn;equals;True	12;12
12	r	report	pass;Flight Reservation page sholud be displayed on successful Login::Flight Reservation page is displayed on successful Login	
13	r	perform	winobj;Date of Flight:	type;031711
14	r	perform	combobox;Fly From:	select;Frankfurt
15	r	perform	combobox;Fly To:	select;Denver
16	r	perform	button;FLIGHT	click
17	r	context	window;Flight Reservation	dialog;Flights Table
18	r	check	list;From	exist

Figure 4: Column 'Object'

The objects that are commonly used are:-

Sl.No	Objects used in the Open Source Test Automation Framework	Window Object Class
1.	Window	Window
2.	Dialog	Dialog
3.	Button	WinButton
4.	Checkbox	WinCheckBox
5.	Listbox	WinList
6.	Textbox	WinEdit
7.	Radiobutton	WinRadioButton
8.	Spinner	WinSpin
9.	Toolbar	WinToolBar
10.	Treeview	WinTreeView
11.	Listview	WinlistView
12.	Menu	WinMenu
13.	Object	WinObject
14.	Editor	WinEditor
15.	Static	Static

16.	Statusbar	WinStatusBar
17.	Scrollbar	WinScrollBar
18.	Tab	WinTab
19.	ActiveX	ActiveX
20.	Combobox	WinCombobox

Table 1: Objects used in the Open Source Test Automation Framework

A detailed description of the keywords is given in the Keyword Reference Document.

2.2.4. ActionValue1 (Column 'D')

The fourth column of the global sheet indicates the specific action being performed on the object present in the AUT. It contains the details of all the operations or verifications that have to be performed on the objects listed in the 'Objects' column.

Consider the example of the object 'WinButton' with the name OK.

One of the actions that can be performed on a WinButton would be Click, so in column 4 the above operation is put in the keyword form as "CLICK".

Example 2: The keyword CLICK on an OK button is as follows:

Action	Object	Operation
Perform	Button;OK	Click

ACTION

If the user wants to **check** if the button is enabled before clicking, the syntax would be:

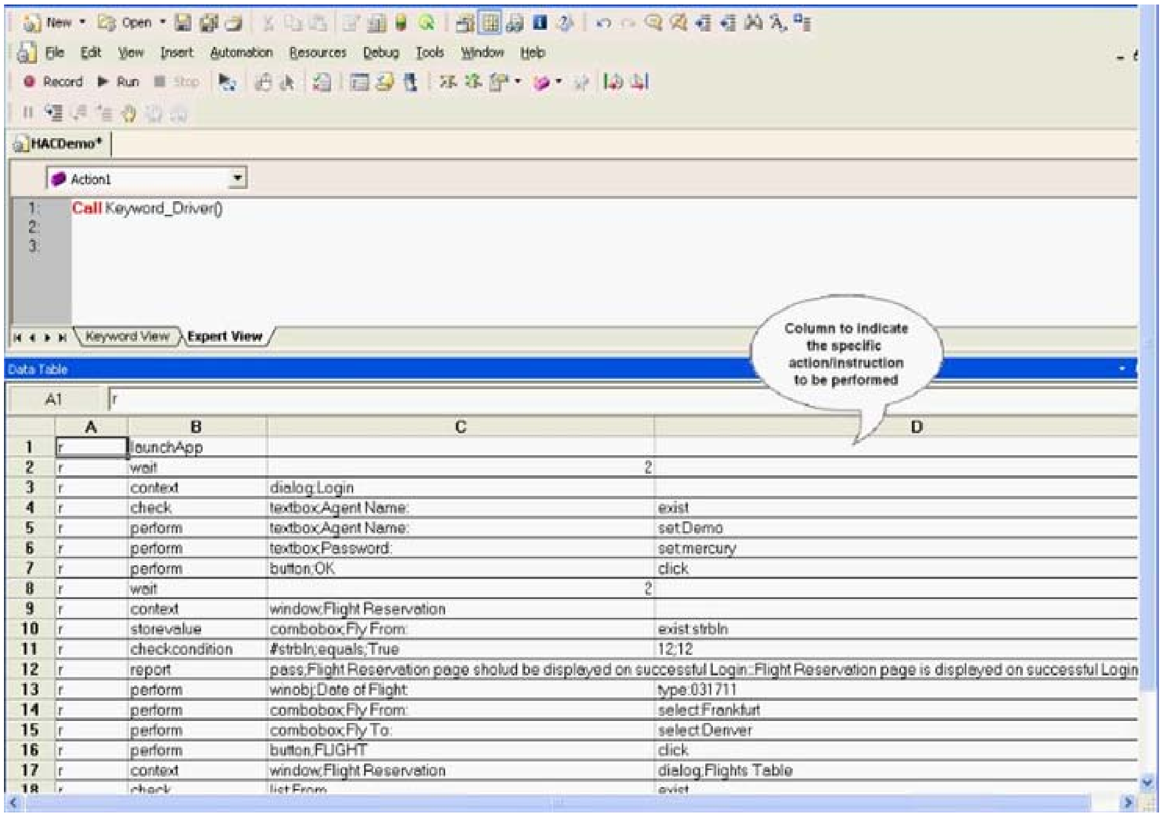
Action	Object	Operation
Check	Button;OK	Enabled:True

CHECKING

It would be the same if the user wants to check whether the object is disabled. The syntax would be:

Action	Object	Operation
Check	Button;OK	Enabled:False

CHECKING



```
Call Keyword_Driver()
```

A1 r

	A	B	C	D
1	r	launchApp		
2	r	wait	2	
3	r	context	dialog;Login	
4	r	check	textbox;Agent Name:	exist
5	r	perform	textbox;Agent Name:	set;Demo
6	r	perform	textbox;Password:	set;mercury
7	r	perform	button;OK	click
8	r	wait	2	
9	r	context	window;Flight Reservation	
10	r	storevalue	combobox;Fly From:	exist;strbln
11	r	checkcondition	#strbln;equals;True	12;12
12	r	report	pass;Flight Reservation page sholud be displayed on successful Login::Flight Reservation page is displayed on successful Login	
13	r	perform	winobj;Date of Flight:	type;031711
14	r	perform	combobox;Fly From:	select;Frankfurt
15	r	perform	combobox;Fly To:	select;Denver
16	r	perform	button;FLIGHT	click
17	r	context	window;Flight Reservation	dialog;Flights Table
18	r	check	list;From	exist

Figure 5: Column 'Actionvalue1'

The most commonly used keywords for specific actions that can be used with the generic keyword '**Perform**' written in Column 'Action' are:-

1. **Click**

   'Click' is used to perform the click operation on objects. It is used with the perform keyword in keyword scripts (ex: clicking a Winbutton).

2. **Close**

   'Close' is used to perform the close operation on any open objects. It is used with the perform keyword in keyword scripts (ex: closing a window, dialog, etc.)

3. **Maximize**

   'Maximize' is used to perform the maximize operation on any open objects. It is used with the perform keyword in keyword scripts.

4. **Minimize**

   'Minimize' is used to perform the minimize operation on any open objects. It is used with the perform keyword in keyword scripts.

5. **Restore**

   'Restore' is used to perform the restore operation on any open objects. It is used with the perform keyword in keyword scripts.

6. **Select:<name/Item>**

   'Select' is used to select an item from Combobox, Listbox, Treeview, Listview, and Tab. It is used with the perform keyword in keyword scripts.

7. **Selectindex:<index>**

   'Selectindex' is used to select an item from a Tab or Combobox. It is used with the perform keyword in keyword scripts.

8. **Set:<Text>**

   'Set' is used to assign a value to an edit field. It is used with the perform keyword.

9. **Type:<Value>**

   'Type' is used to assign a value to an edit field. It is used with the perform keyword.

10. **Type:<Item>**

    'Type' is used to select a item from Combobox. It is used with the perform keyword.

11. **Type:d_currenttime**

    This sets the current system time to the edit field. It is used with the perform keyword in keyword scripts.

12. **Type:d_currentdate**

    This sets the current system date to the edit field. It is used with the perform keyword in keyword scripts.

13. **Type:d_d; <value to be added/subtracted>**

    This adds or subtracts the value specified to the current system date and sets the edit field to a given value. It is used with the perform keyword in keyword scripts.

14. **Type:d_m; <value to be added/subtracted>**

    This adds or subtracts the value specified to the current system month and sets the edit field to a given value. It is used with the perform keyword in keyword scripts.

15. **Type:d_y; <value to be added/subtracted>**

    This adds or subtracts the value specified to the current system year and sets the edit field to a given value. It is used with the perform keyword in keyword scripts.

16. **Setdate:Date/Now/<Date>**

    This sets the current system date (Date)/ current system date and time.

    (Now)/specified date(<date>) to the calendar object. It is used with the perform keyword in keyword scripts.

17. **SetTime:Now/<Time>**

    This sets the current system time (Now)/specified time(<Time>) to the calendar object. It is used with the perform keyword in keyword scripts.

18. **Set**

    This is used to select a radio button. It is used with the perform keyword in keyword scripts.

19. **Doubleclick**

    ‘Doubleclick’ is used to perform the doubleclick operation on objects. It is used with the perform keyword in keyword scripts.

20. **Press:<name>**

    This is a perform operation to click on the specified toolbar item.

21. **Expand:<Item name>**

    This is used to expand the treeitem specified. It is used with the perform keyword in keyword scripts.

22. **ExpandAll:<item name>**

    This is used to expand all the treeitems in a treeview. It is used with the perform keyword in keyword scripts.

23. **<conversiontype>:<variable name>:<format type>**

    This is used to convert a variable from one data type to another.

24. **Collapse:<name>**

    This is used to collapse the treeitem specified. It is used with the perform keyword in keyword scripts.

25. **SelectRange:<item name1>:<item name2>**

    This is used to select the range of items in a listview.

26. **NextLine[:<line number>]**

    This is a perform operation to scroll to next line number.

27. **PrevLine[:<line number>]**

    This is a perform operation to scroll to the previous line number.

28. **NextPage[:<line number>]**

    This is a perform operation to scroll to the next page.

29. **PrevPage[:<line number>]**

    This is a perform operation to scroll to the previous page.

30. **Create;<Folder Path/Name>/<File Path/Name>**

    This is used to create a folder/file in the specified path.

31. **Delete;<Folder Path/Name>/<File Path/Name>**

    This is used to delete a folder/file in the specified path.

32. **Copy;<Source Path/Name>;<DestinationFolder Path/Name>/<Source File Path/Name>;<Destination Folder Path>**

    This is used to copy a folder/file from the source to the destination path specified.

33. **Move;<Source Path/Name>;<DestinationFolder Path/Name>/<Source File Path/Name>;<Destination Folder Path>**

    This is used to move a folder/file from the source to the destination path

34. **Write;<File Path/Name>;<The value to be entered>**

    This is used to write the file with the data mentioned in the specified path.

35. **Read;<File Path/Name>;<Variable to store data from file>**

    This is used to read the contents of a mentioned file and store the values in the specified variable.

36. **Append;<File Path/Name>;<text to be appended to file>**

    This is used to append the data specified with the data contained in the file.

37. **DBObjectName:OutputCheckPointName**

    This is used for capturing multiple values from the database.

    DBObjectName is the name of the DB Object to be present in the repository and the Output Checkpoint is the name of the checkpoint placed inside where many output values are captured.

38. **TextClick:<text>**

    This is used to click on the specified text in the Window.

The most commonly used keywords for specific actions that can be used with the generic keyword '**Check**' written in Column 'Action' are:-

1. **Selection:<item name>**

   This is a check operation that is used to verify whether the desired item is selected or not from the combobox, Listbox, and Tab.

2. **Checked:<On/OFF/Dimmed>**

   This is a check operation that is used to verify whether a radio button is selected or not.

3. **Checked:<On/OFF>**

   This is a check operation that is used to verify whether a checkbox is checked or not.

4. **Enabled:<True/False>**

   This is a check operation that is used to verify whether the given window object is enabled or not.

5. **Exist:<True/False>**

   This is a check operation that is used to verify whether the window object whose name is specified exists or not.

6. **Focused:<True/False>**

   This is a check operation that is used to verify whether the object is focused or not.

7. **ItemCount:<Item>**

   This is a check operation that is used to verify the number of items present or not in a window object.

8. **Text:<text/#Variable_Name>**

   This is a check operation that is used to verify whether the required text is present or not in the object.

9. **Prop_name:<variable_name>**

   This is used to store the property value in the specified variable. It is used with the storevalue keyword.

10. **Windowtext: <Text>:<True/False>**

    This is a check operation that is used to verify whether a windows text is present or not in the window object.

11. **Tabexist:<Tabitemname>**

    This is a check operation that is used to verify whether the tab item specified is present or not.

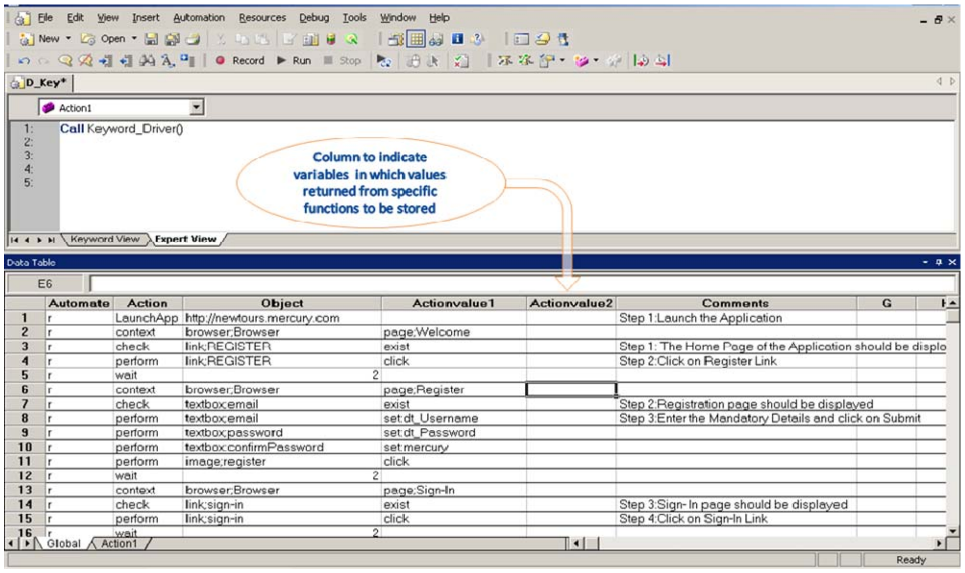
12. **ItemExist:<Item name>**

    This is a check operation that is used to verify whether an Item is present or not in the window object.

A detailed description of the keywords is provided in the Keyword Reference Document.

2.2.5. ActionValue2 (Column 'E')

The fifth column of the global sheet may be used to store the values returned from specific functions (Ex. User-defined functions).


File Edit View Insert Automation Resources Debug Tools Window Help
New Open Record Run Stop
D_Key*
Action1
Column to indicate variables in which values returned from specific functions to be stored
Keyword View
Expert View
Data Table
E6
Global
Action1
Ready

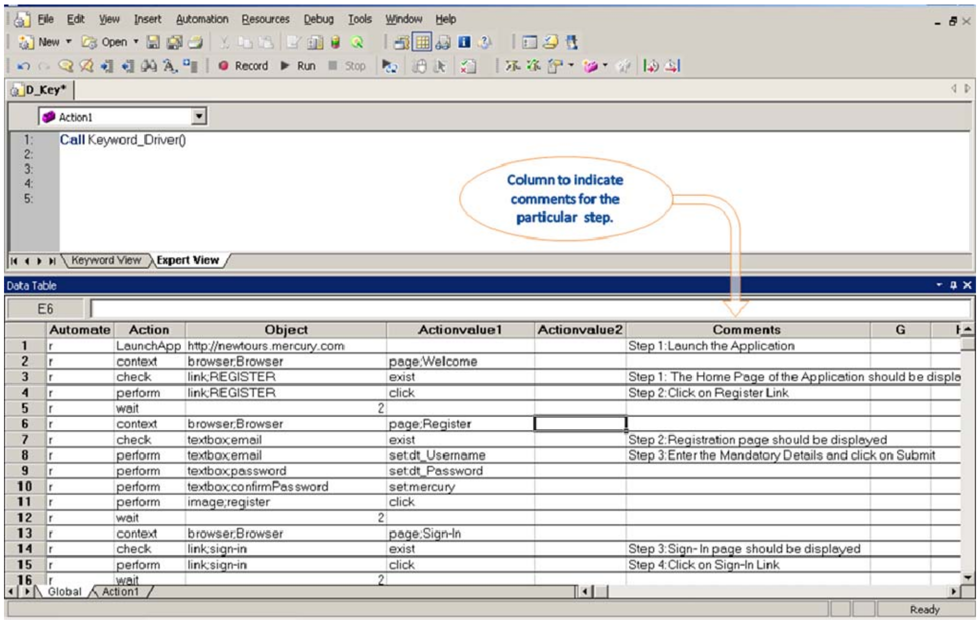
```
Call Keyword_Driver()
```

	Automate	Action	Object	Actionvalue1	Actionvalue2	Comments	G
1	r	LaunchApp	http://newtours.mercury.com			Step 1:Launch the Application	
2	r	context	browser;Browser	page;Welcome			
3	r	check	link;REGISTER	exist		Step 1: The Home Page of the Application should be displo	
4	r	perform	link;REGISTER	click		Step 2:Click on Register Link	
5	r	wait	2				
6	r	context	browser;Browser	page;Register			
7	r	check	textbox;email	exist		Step 2:Registration page should be displayed	
8	r	perform	textbox;email	set:dt_Username		Step 3:Enter the Mandatory Details and click on Submit	
9	r	perform	textbox;password	set:dt_Password			
10	r	perform	textbox:confirmPassword	set:mercury			
11	r	perform	image;register	click			
12	r	wait	2				
13	r	context	browser;Browser	page;Sign-In			
14	r	check	link;sign-in	exist		Step 3:Sign- In page should be displayed	
15	r	perform	link;sign-in	click		Step 4:Click on Sign-In Link	
16	r	wait	2				

Figure 6: Column 'Actionvalue2'

2.2.6. Comments (Column 'F')

The 'Comments' column is used to enter generic information about the current step being run. It provides a better understanding of the steps being performed in the particular test script and also helps to map the test script to the manual test case.


File Edit View Insert Automation Resources Debug Tools Window Help
New Open Record Run Stop
D_Key*
Action1
Column to indicate comments for the particular step.
Keyword View
Expert View
Data Table
E6
Global
Action1
Ready

```
Call Keyword_Driver()
```

	Automate	Action	Object	Actionvalue1	Actionvalue2	Comments	G
1	r	LaunchApp	http://newtours.mercury.com			Step 1:Launch the Application	
2	r	context	browser;Browser	page;Welcome			
3	r	check	link;REGISTER	exist		Step 1: The Home Page of the Application should be disple	
4	r	perform	link;REGISTER	click		Step 2:Click on Register Link	
5	r	wait	2				
6	r	context	browser;Browser	page;Register			
7	r	check	textbox;email	exist		Step 2:Registration page should be displayed	
8	r	perform	textbox;email	set:dt_Username		Step 3:Enter the Mandatory Details and click on Submit	
9	r	perform	textbox;password	set:dt_Password			
10	r	perform	textbox:confirmPassword	set:mercury			
11	r	perform	image;register	click			
12	r	wait	2				
13	r	context	browser;Browser	page;Sign-In			
14	r	check	link;sign-in	exist		Step 3:Sign- In page should be displayed	
15	r	perform	link;sign-in	click		Step 4:Click on Sign-In Link	
16	r	wait	2				

Figure 7: Column 'Comments'

2.2.7. Delimiters

Delimiters are any string characters used to identify the sub-string limits. Delimiters are generally used with the Split function, which is used to split the input in to different substrings.

When a delimiter is omitted, the space character (" ") is assumed to be a delimiter.

Purpose of using delimiters:

The main purpose of using delimiters in this framework is to break down the input values to different strings and take them as keywords to perform any operation concerned with that object.

Delimiters Used in this framework:

The most important point to keep in mind while scripting using the keyword-driven approach is to place separators or delimiters between two keywords. Delimiters that are used in the framework are:

- **:** (colon)
- **;** (semi colon)
- **::** (double colon)

Understanding the usage of delimiters:

There are four columns involved in the keyword-driven approach. The role of delimiters comes in the 'Objects' column (column 3) and 'Operations' column (column 4).

'Objects' column (column 3):

This column is used to define the class and the name of the object. The delimiter used in this column separates the class of the object and the name of the object with a semi-colon ';'.

Example:

```
Textbox; <textboxname>
```

'ActionValue1' column (column 4):

This column usually provides the details of the operations that need to be performed on the object. The delimiter used to separate the property and the property values in this column is a colon ':'.

Example:

```
Selectindex: <index>
```

To specify the child objects present in a window, browser, or dialog box, the delimiter that is used is a double colon '::'

Example:

```
page;<name> :: window;<name> :: Dialog;<name>
```

To specify the optional parameters to be used for certain keywords, the delimiter used is double hyphen '--'

Example:

```
TableSearch:<colname1>;<rowval1>::<colname2>;<rowval2>--<[no of columns]>
```

'ActionValue1' column (column 4):

This column is usually used to specify variables in which the output parameters of certain functions are to be stored. The delimiter used is a colon ':'

Example:

TableSearch:<colname1>;<rowval1>::<colname2>;<rowval2>--<[no of columns]>	intx:inty

2.2.8. Variables

- To store a value in a variable, an environment variable is used.

  Example:

assignvalue	strName;Smith	

  Here in the variable 'strName', the value 'Smith' is stored.

- To store the property value of an object, an environment variable is used.

  Example:

storevalue	Textbox;<textbox name>	Prop_name:<varName>

  Here, the value in the textbox is stored to a variable 'varName'

- To input a value to a field from a variable, the variable should be preceded by '#'.

  Example:

Perform	Textbox;<textbox name>	Set:#varName

  Here, the value stored in varName is typed into the textbox.

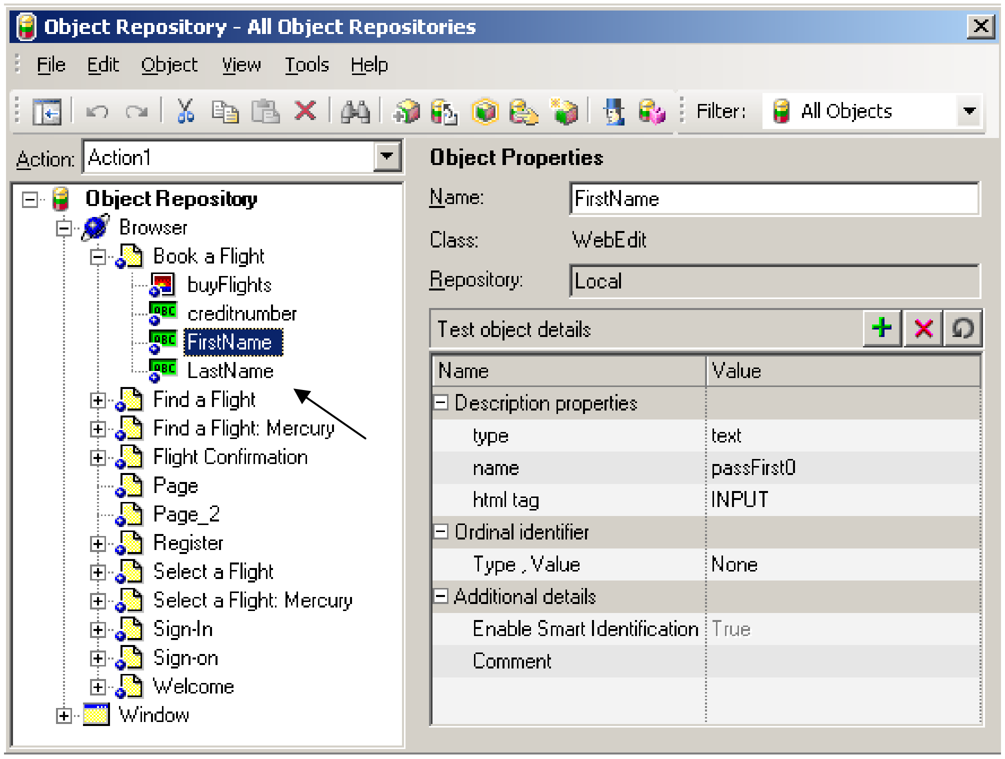
To define a variable certain standards need to be followed. For example, for a variable to store a string value it should be appended with "Str" ex.StrVarName. Similarly, for integer, it should be appended with "int" and for Boolean should be appended with "bln".

3. Sequence of Keywords

While scripting using keywords, some keywords have to be written in combination with other keywords. This section deals with those methodologies.

3.1. Use of keyword 'Context'

The keyword 'context' has to be used whenever the AUT screen changes. Example:


Object Repository - All Object Repositories
File Edit Object View Tools Help
Filter: All Objects
Action: Action1

Object Repository
- Browser
  - Book a Flight
    - buyFlights
    - creditnumber
    - FirstName
    - LastName
  - Find a Flight
  - Find a Flight: Mercury
  - Flight Confirmation
  - Page
  - Page_2
  - Register
  - Select a Flight
  - Select a Flight: Mercury
  - Sign-In
  - Sign-on
  - Welcome
- Window

Object Properties

Name: FirstName
Class: WebEdit
Repository: Local

Test object details

Name	Value
Description properties	
type	text
name	passFirst0
html tag	INPUT
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	True
Comment	

Figure 8: Keyword 'Context'

If the object 'FirstName' has to be used in the script then the preceding row should have the context set to the previous object in the hierarchy.

Therefore, the combination to be used while performing an action on the object 'FirstName' is :

Context	Browser; Browser	Page;Book a Flight
Perform	Textbox;FirstName	Set:Smith

If we have to use another object on the same page then the context need not be set again.

Context	Browser; Browser	Page;Book a Flight
Perform	Textbox;FirstName	Set:Smith

Perform	Textbox;LastName	Set:Smith

3.2. Use of 'Conditional statements'

If the user is implementing an If – Else conditional statement, then the keyword is followed by a semi-colon ';' and the values that indicate the start row and the end row should be separated by a semi-colon ';'.

Example:

Condition	<var1>;comparator;<var2>	startrow;endrow

If the condition mentioned is 'True', execution starts from the startrow and would end at the endrow specified. If the condition specified is 'False' there would be no effect in the script and the execution would continue as normal.

When an operation has to be used, then two conditional statements have to be used together to satisfy the 'and' condition

Condition	<var1>;comparator;<var2>	startrow;endrow
Condition	<var2>;comparator;<var3>	startrow;endrow

Therefore, this effectively implies that an 'and' operation is being performed.