



Open2Test Flex Test Automation Framework Scripting Standards for QTP

Version 1.0

March 2011

DISCLAIMER

Verbatim copying and distribution of this entire article are permitted worldwide, without royalty, in any medium, provided this notice is preserved.

TABLE OF CONTENTS

1. INTRODUCTION	3
1.1. Purpose	3
2. STANDARDS FOR KEYWORD SCRIPTING	4
2.1. Getting Started	4
2.2. Column Description	4
2.2.1. Automate (Column 'A')	4
2.2.2. Action (Column 'B')	5
2.2.3. Object (Column 'C')	7
2.2.4. ActionValue1 (Column 'D')	9
2.2.5. ActionValue2 (Column 'E')	14
2.2.6. Comments (Column 'F')	15
2.2.7. Delimiters.....	16
2.2.8. Variables.....	17
3. SEQUENCE OF KEYWORDS	18
3.1. Use of the Keyword 'Context'	18
3.2. Use of 'Conditional Statements'	19

1. Introduction

1.1. Purpose

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

2. Standards for Keyword Scripting

2.1. Getting Started

Before learning about the columns used for keyword scripting, the user should know 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()'. In the 'Data Table', the framework is called using the command 'Call Keyword_Driver()'.

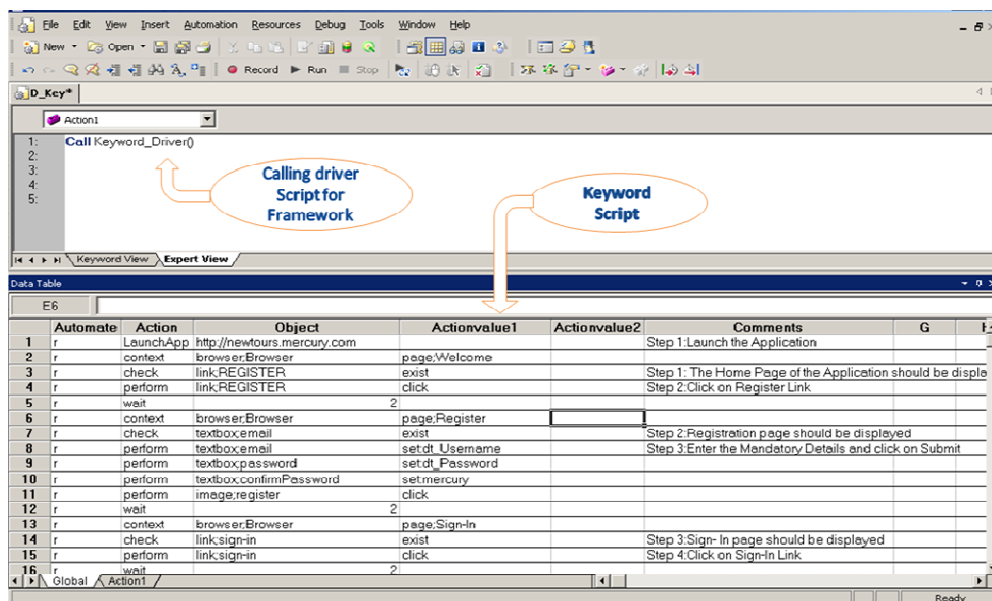


Figure 1: Keyword Script and Calling the Framework

2.2. Column Description

This section describes the columns used for keyword scripting.

2.2.1. Automate (Column 'A')

The data in the 'Automate' column identifies 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 to be run, then the corresponding row in the first column is to be left blank. All the steps will run based on the data in this column.

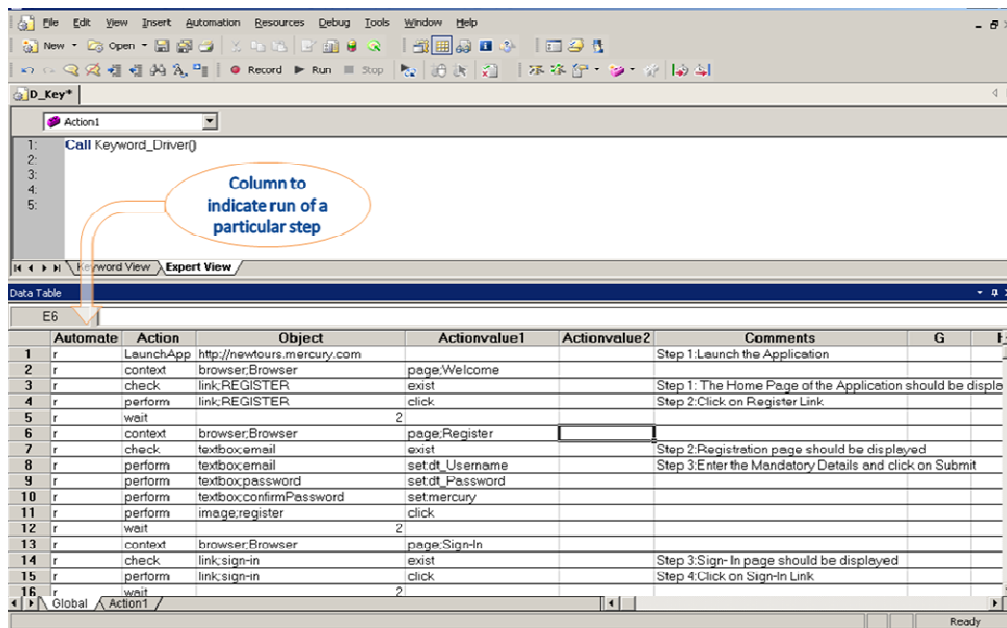


Figure 2: Column 'Automate'

2.2.2. Action (Column 'B')

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

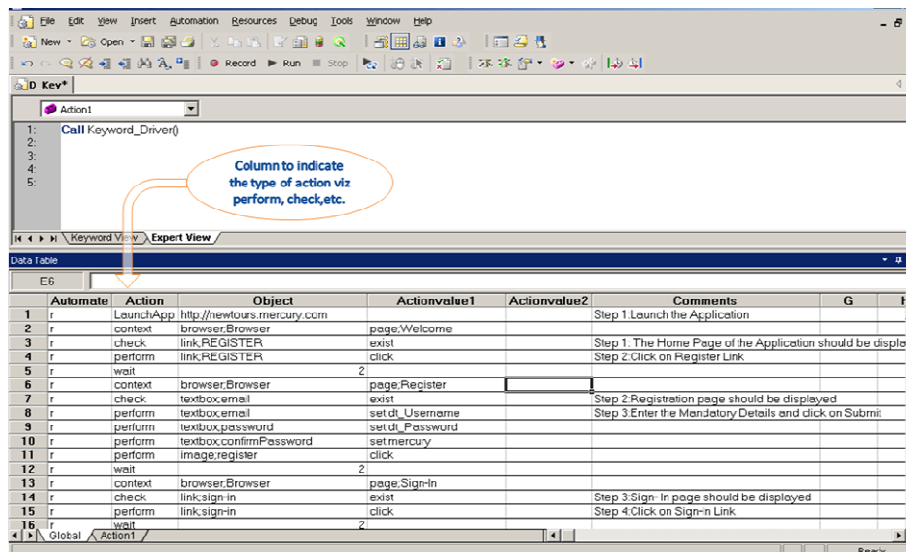


Figure 3: Column 'Action'

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 of which is specified in the third column) or, if the application is already synchronized with QuickTest Professional (QTP), then this automatically launches the application from the location specified in QTP.

2. Context

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

3. Perform

'Perform' is used to perform an operation on a particular object (ex: clicking on a button, closing an open Flex Application, or typing some text in a textbox). 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' is used to compare two variables, check properties for the existence of windows, etc.

6. CallFunction

'Call Function' is used to call any declared function used 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 results can include: 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 another.

14. Wait

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

15. Arith

'Arith' is used to perform 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.



A detailed description of the keywords is provided in the Open2Test Flex Test Automation Framework Keywords for QTP 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 object repository, descriptive programming can be used by specifying any property and its value (as shown in Example 2). The object column, or column 'C', contains 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	Button;OK
Perform	Textbox;Lastname

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

to be performed on an object of class 'FlexTextArea' having the name 'Lastname'.

Example 2:

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

In the above example, the object is not added to the object repository. Some operation has to be performed on an object of class FlexButton having a property 'text', the value of which is 'OK'. Similarly, some operation has to be performed on an object of class FlexTextArea 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 later in this document).

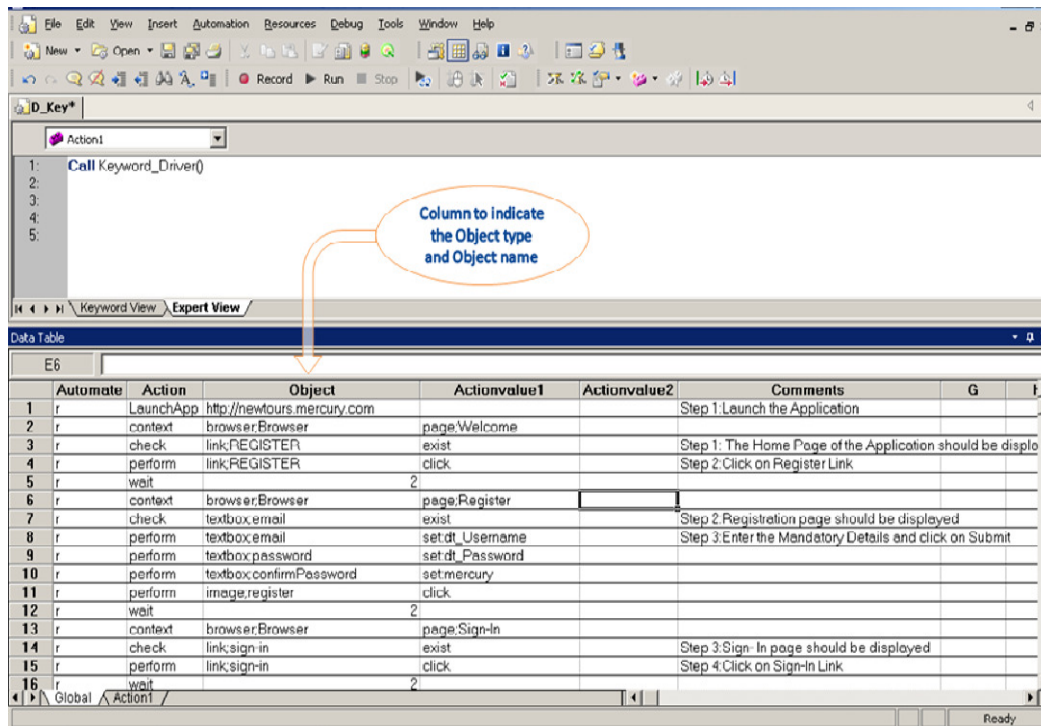


Figure 4: Column 'Object'

Commonly used objects include:-

S1.No	Objects used in the Open2Test Flex Test Automation Framework for QTP	Object Class
1.	Browser	Browser
2.	Dialog	Dialog
3.	Windows	Window
4.	Application/SparkApplication	FlexApplication/ SparkApplication

5.	Canvas/SparkCanvas	FlexCanvas/ SparkCanvas
6.	Panel/SparkPanel	FlexPanel/ SparkPanel
7.	Alert	FlexAlert
8.	Box	FlexBox
9.	Form	FlexForm
10.	DivBox	FlexDividedBox
11.	TextBox/SparkTextBox	FlexTextArea/ SparkTextArea
12.	Checkbox/SparkCheckbox	FlexCheckBox/ SparkCheckbox
13.	Combobox/SparkCombobox	FlexComboBox/ SparkComboBox
14.	Radiobutton/SparkRadiobutton	FlexRadioButton/ SparkRadioButton
15.	Button/SparkButton	FlexButton/ SparkButton
16.	Togglebutton/SparkTogglebutton	FlexToggleButton/ SparkToggleButton
17.	Tree/SparkTree	FlexTree/ SparkTree
18.	Menu/SparkMenu	FlexMenu/ SparkMenu
19.	AdvGrid/SparkAdvGrid	FlexAdvancedDataGrid/ SparkAdvancedDataGrid
20.	Datagrid/SparkDatagrid	FlexDataGrid/ SparkDataGrid
21.	List/SparkList	FlexList/ SparkList
22.	Label/SparkLabel	FlexLabel/ SparkLabel
23.	ListLabel/SparkListLabel	FlexListLabel/ SparkListLabel
24.	PopUpButton/SparkPopUpButton	FlexPopUpButton/ SparkPopUpButton
25.	Spinner/SparkSpinner	FlexSpinner/ SparkSpinner
26.	Slider/SparkSlider	FlexSlider/ SparkSlider
27.	MenuBar/SparkMenuBar	FlexToggleButtonBar/ SparkToggleButtonBar

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

2.2.4. ActionValue1 (Column 'D')

The fourth column of the global sheet is used to indicate the specific action being performed on the objects present in the AUT. It contains the details of all the operations or verifications to be performed on the objects listed in the objects column.

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

One of the actions that can be performed on a FlexButton would be Click.

Therefore, in column 4 the above operation is mentioned 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

Similarly, if the user wants to check whether the object is disabled, the syntax would be

Action	Object	Operation
Check	Button;OK	Enabled:False

CHECKING

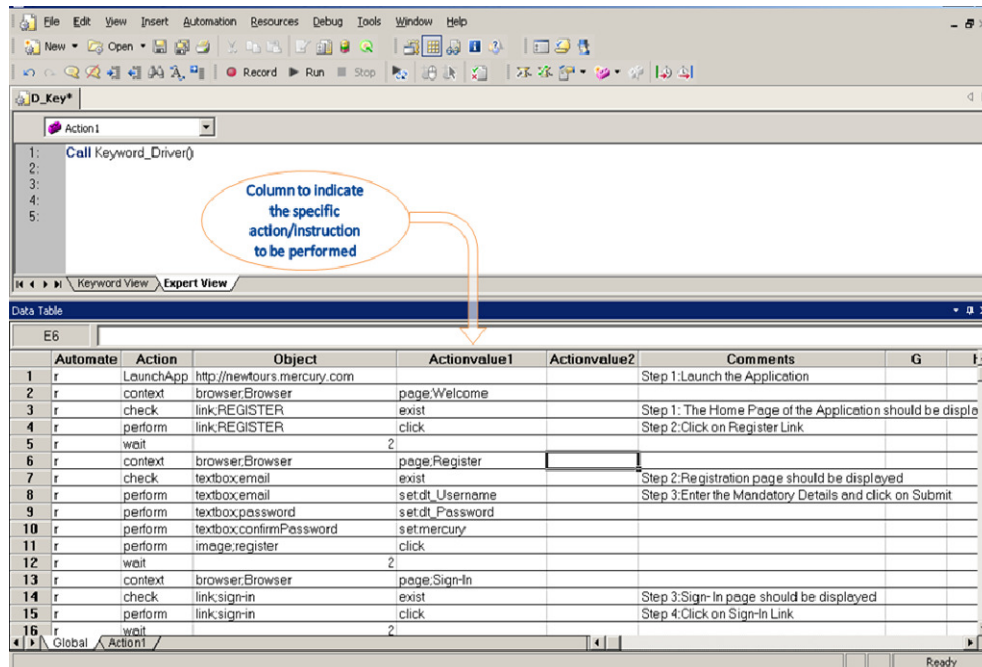


Figure 5: Column 'Actionvalue1'

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 (ex. clicking a button, clicking a link, etc).

2. Close

'Close' is used to perform the close operation on any open objects. (ex. closing a browser, window, or dialog box).

3. Select:<name>

'Select' is used to select an item from Combobox, Listbox, menu bar or tree node.

4. Selectindex:<index>

'Selectindex' is used to select an item from a Listbox or Combobox.

5. Set:<Text>

'Set' is used to assign a value to an edit field.

6. Set:d_currenttime

'Set:d_currenttime' sets the current system time in the edit field.

7. Set:d_currentdate

'Set:d_currentdate' sets the current system date in the edit field.

8. Set:d_d; <value to be added/subtracted>

This function adds or subtracts the value specified to the current system date and sets the edit field to a given value.

9. Set:d_m; <value to be added/subtracted>

This function adds or subtracts the value specified to the current system month and sets the edit field to a given value.

10. Set:d_y; <value to be added/subtracted>

This function adds or subtracts the value specified to the current system year and sets the edit field to a given value.

11. Set:<On/Off>

This function is used to select or deselect a Radio button

12. set:env_<Environment Name>

This function is used to assign the value, which is stored in environment variable, to the edit field.

13. set:#<variable>

This function is used to assign the value, which is stored in the variable, to the edit field.

14. set:p_<parameter>

This function is used to assign the value, which is stored in the Input Parameter, to the edit field.

15. **set:dt_Parameter**
This function is used to assign the value, which is given in the external test data sheet, to the edit field.
16. **<conversiontype>:<variable name>:<format type>**
This function is used to convert a variable from one data type to another.
17. **Set:blank**
This function is used to clear the edit field.
18. **Type:<value>**
This function is used to enter the values at the current cursor position, in the edit field.
19. **Getrownum:**
This function is used to get the row number of a grid based on the row value.
20. **Append:<value>**
This function is used to append the values to the existing text in the edit field.
21. **Selecttext:<startvalue>:<endvalue>**
This function is used to select a part of text based on the start and end values.
22. **HeaderClick:<position>:<headername>**
This function is used to click on the Header of a grid based on the Header column name and position.
23. **Check:**
This function is used to check a particular checkbox.
24. **Uncheck:**
This function is used to uncheck a particular checkbox.
25. **Navigate:<URL>**
This function is used to navigate the browser to the specified URL.
26. **SetFocus:**
This function is used to set the focus onto the specified object.
27. **Maximize/Minimize:**
This function is used to maximize or minimize the browser.

28. **Create;<Folder Path/Name>/<File Path/Name>**
This function is used to create a folder/file in the specified path.
29. **Delete;<Folder Path/Name>/<File Path/Name>**
This function is used to delete a folder/file in the specified path.
30. **Copy;<Source Path/Name>;<DestinationFolder Path/Name>/<Source File Path/Name>;<Destination Folder Path>**
This function is used to copy a folder/file from the source to the destination path specified.
31. **Move;<Source Path/Name>;<DestinationFolder Path/Name>/<Source File Path/Name>;<Destination Folder Path>**
This function is used to move a folder/file from the source to the destination path.
32. **Write;<File Path/Name>;<The value to be entered>**
This function is used to write the file with the data mentioned in the specified path.
33. **Read;<File Path/Name>;<Variable to store data from file>**
This function is used to read the contents of a mentioned file and store the values in the specified variable.
34. **Append;<File Path/Name>;<text to be appended to file>**
This function is used to append the data specified with the data contained in the file.
35. **Descriptive Programming**
Descriptive programming can also be used with the above object types when the object is not added to the object repository.

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

1. **Selecteditem:<item name>**
This is a Check operation that is used to verify whether the desired item is selected in the Combobox or listbox or data grid.
2. **Selected:<True/False>**
This is a Check operation that is used to verify whether a Checkbox or Radiobutton is selected or not.
3. **Enabled:<True/False>**
This is a Check operation used to verify whether the given object is enabled or not.
4. **Exist:<True/False>**
This is a Check operation that is used to verify whether the Web object whose name is specified exists or not.

5. **Focused:<True/False>**
This is a Check operation that is used to verify whether the object is focused or not.
6. **Visible:<True/False>**
This is a Check operation that is used to verify whether the specified object is visible or not.
7. **ItemCount:<Item>**
This is a Check operation that is used to verify the number of items in a Combobox or Listbox or Grids.
8. **Text:<text/#Variable_Name>**
This is a Check operation that is used to verify whether the required text is present in the object.
9. **ErrorText:<text>**
This is a Check operation that is used to verify whether the textbox holds the appropriate error text.
10. **Selectedindex:<index value>**
This is a Check operation that is used to verify whether the desired item is selected in the Combobox or listbox or data grid based on index value.
11. **Rows:<value>**
This is a Check operation used to verify the number of rows in a grid.
12. **Columns:<value>**
This is a Check operation used to verify the number of columns in a grid.
13. **prop_name:<input value>**
This is a Check operation used to verify whether the property of an object is as per the input value.
14. **Text:blank**
This is a Check operation that is used to verify whether a text field is blank or not.

The most commonly used keywords for specific actions used with the generic keyword 'Storevalue' written in Column 'Action' is:-

1. **prop_name:<variable name>**
This function is used to store the property value of an object in the specified variable.

2.2.5 ActionValue2 (Column 'E')

The fifth column of the global sheet may be used to store the values that are returned from specific functions (ex:user-defined functions).

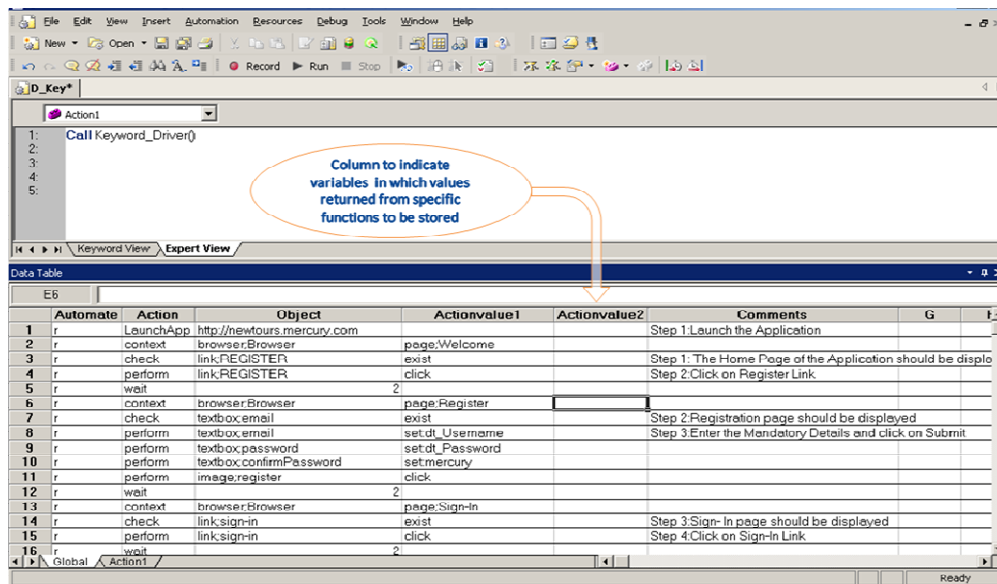


Figure 6: Column 'Actionvalue2'

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.

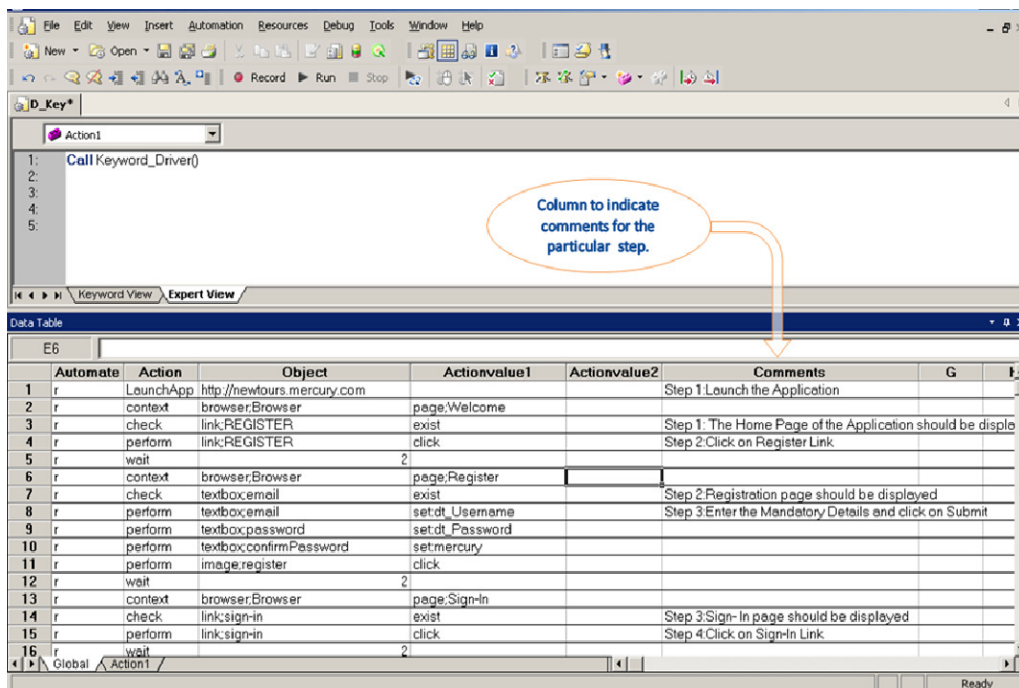


Figure 7: Column 'Comments'

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

Purpose of Delimiters:

Delimiters 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:

When scripting using the keyword-driven approach, use separators or delimiters appropriately between two keywords. Delimiters that are used in the framework are:

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

Using Delimiters:

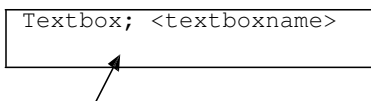
There are four columns involved in the keyword-driven approach. The role of delimiters comes in the 'Objects' column (column 3) and the 'Operations' column (column 4).

'Objects' column (column 3):

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

Example:

```
Textbox; <textboxname>
```

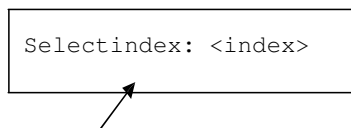


'ActionValue1' column (column 4):

This column usually provides 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:

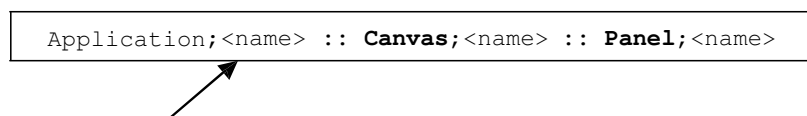
```
Selectindex: <index>
```



To specify the child objects present in a window, browser, or dialog box, use the double colon '::' delimiter.

Example:


```
Application; <name> :: Canvas; <name> :: Panel; <name>
```



To specify the optional parameters to be used for certain keywords, use the symbol '>' delimiter.

Example:

Select: Inbox>Sent Items




'ActionValue1' column (column 4):

This column is most often used to specify variables in which the output parameters of certain functions are to be stored.

Example

GetRowNum:<row value>	intx
-----------------------	------



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;<name>	Prop_value:<name>
------------	----------------	-------------------

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

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

Example:

Perform	Textbox;<name>	Set:#varName
---------	----------------	--------------

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

Defining a variable requires the use of certain standards. For example, for a variable to store a string value it should be appended with "Str" ex.StrVarName. Similarly, an integer should be appended with "int" and 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 methodologies that should be followed while scripting using keywords.

3.1. Use of the Keyword 'Context'

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

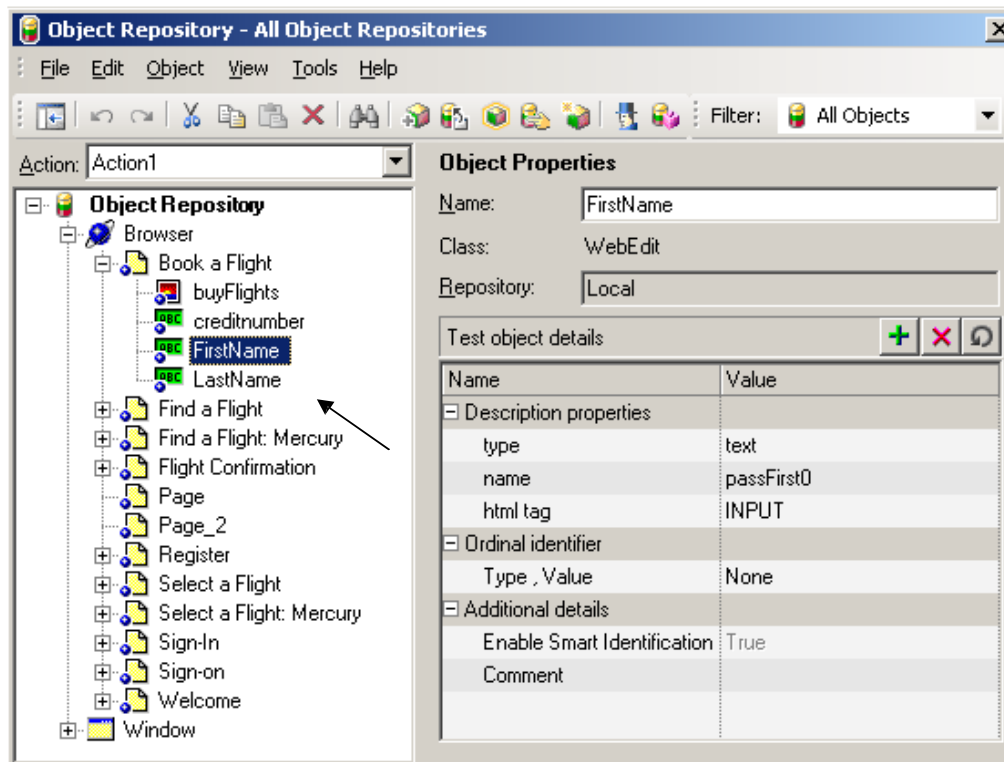


Figure 8: Keyword 'Context'

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 are going 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.

Two conditional statements must be used together to satisfy the 'and' condition.

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

This implies that an 'and' operation is being performed.

COPYRIGHT

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.