



Open2Test Flex Test Automation Framework for QTP – Keyword Naming Conventions for Developers

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.	PURPOSE OF THE DOCUMENT	4
2.	KEYWORD NAMING CONVENTIONS	5
2.1.	Naming Conventions Used for Keywords	5
2.2.	Naming Conventions Used for Objects	7
2.3.	Naming Conventions Used for Actions	9
2.4.	Naming Conventions Used for Variables	14
2.5.	Naming Convention Used for Datatable	15
2.6.	Naming Convention Used for Parameter	15
2.7.	Naming Convention Used for Environment	15

LIST OF TABLES

Table 1: Naming Conventions Used for Objects	7
--	---

1. Purpose of the Document

This document provides the naming conventions followed in Open2Test Test Automation Framework for the Flex technology.

2. Keyword Naming Conventions

2.1. Naming Conventions Used for Keywords

1. LaunchApp

‘LaunchApp’ is used to launch the application under test (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 QTP, then this automatically launches the application from the location specified in QTP.

2. Context

‘Context’ is used only on a Web page or a dialog object. This keyword brings a particular Web page or dialog to the current context, so that any operation or checking can be performed on that particular Web page or dialog.

3. Perform

‘Perform’ is used to perform an operation on a particular object such as clicking on a button, closing an open Web page, or typing 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, checking properties, existence of windows, etc.

6. CallFunction

‘Call Function’ is used to call any declared function, which is 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 as well as in scripts.

8. PressKey

‘Press Key’ is used to pass the 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 four 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 for typecasting from one data type to another.

2.2. Naming Conventions Used for Objects

The table below provides the naming conventions followed for different Web objects in the Open Source Test Automation Framework.

	Flex Object Class	Naming Conventions
1	FlexApplication	Application
2	FlexCanvas	Canvas
3	FlexPanel	Panel
4	FlexAlert	Alert
5	FlexBox	Objectbox
6	FlexDividedBox	Divbox
7	FlexForm	Form
8	FlexAdvancedDataGrid	Advgrid
9	FlexList	List
10	FlexTitleWindow	Titlewindow
11	SparkApplication	Sparkapplication
12	SparkCanvas	Sparkcanvas
13	SparkApplication	Sparkapplication
14	Window	Window
15	Dialog	Dialog
16	Static	Static
17	WinButton	Winbutton
18	FlexDataGrid	Datagrid
19	FlexTextArea	Textbox
20	FlexTextInput	Textinput
21	FlexButton	Button
22	FlexButtonBar	Buttonbar
23	FlexPopUpButton	Popupbutton
24	FlexToggleButton	Togglebutton
25	FlexComboBox	Combobox
27	FlexRadioButton	Radiobutton
28	FlexCheckBox	Checkbox
29	FlexImage	Image
30	FlexLabel	Label

31	FlexListLabel	Listlabel
32	FlexList	List
33	FlexBox	Objectbox
34	FlexDividedBox	Divbox
35	FlexForm	Form
36	FlexSlider	Slider
37	FlexSpinner	Spinner
38	FlexTabNavigator	Tab
39	FlexToggleButtonBar	Menubar
40	FlexMenu	Menu
41	FlexTree	Tree
42	FlexLinkButton	Link
43	SparkAdvancedDataGrid	Sparkadvanceddatagrid
44	SparkDataGrid	Sparkdatagrid
45	SparkTabNavigator	Sparktabnavigator
46	SparkTextArea	Sparktextarea
47	SparkTextInput	Sparktextinput
48	SparkButton	Sparkbutton
49	SparkButtonBar	Sparkbuttonbar
50	SparkPopUpButton	Sparkpopupbutton
51	SparkComboBox	Sparkcombobox
52	SparkRadioButton	Sparkradiobutton
53	SparkCheckBox	Sparkcheckbox
54	SparkImage	Sparkimage
55	SparkLabel	Sparklabel
56	SparkListLabel	Sparklistlabel
57	SparkList	Sparklist
58	SparkSlider	Sparkslider
59	SparkSpinner	Sparkspinner
60	SparkMenu	Sparkmenu
61	SparkTree	Sparktree
62	SparkGroup	Sparkgroup
63	SparkDropDownList	Sparkdropdown

2.3. Naming Conventions Used for Actions

1. Click
‘Click’ is used to perform the click operation on objects.
(ex. Clicking a WebButton, Link WebElement, etc.)
2. Close
‘Close’ is used to perform the close operation on any open objects.
(ex. Closing a Browser, Window, Dialog, etc).
3. Select:<name>
‘Select’ is used to select an item from Combobox or Radiobutton.
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 a edit field.
6. Set:d_currenttime
This sets the current system time to the edit field.
7. Set:d_currentdate
This sets the current system date to the edit field.
8. Set: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.
9. Set: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.
10. Set: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.
11. Set:<On/Off>
This is used to select or deselect a Radiobutton.
12. Submit
This is used to submit the value entered in a Textbox, Combobox, and Radiobutton, etc.

13. `Deselect:<name>`
This is used to deselect a selected item in a list.
14. `Extendselect:<name>`
This is used to select more than one item from a list.
15. `Selection:<item name>`
This is a check operation that is used to verify whether the desired item is selected or not from the combobox.
16. `Checked:<Name>`
This is a check operation that is used to verify whether a Radiobutton in a Webradiogroup whose name is specified is checked or not.
17. `Checked:<True/False>`
This is a check operation that is used to verify whether a checkbox is checked or not.
18. `Enabled:<True/False>`
This is a check operation that is used to verify whether the given Web object is enabled or not.
19. `Exist:<True/False>`
This is a check operation that is used to verify whether the Web object whose name is specified exists or not.
20. `Focused:<True/False>`
This is a check operation that is used to verify whether the object is focused or not.
21. `Visible:<True/False>`
This is a check operation that is used to verify whether the specified object is visible or not.
22. `ItemsCount:<Item>`
This is a check operation that is used to verify the number of items in a Combobox.
23. `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.
24. `TableSearch:<colname1>; <rowvall>:: <colname2>; <rowval2>--<[no of columns]>`
This is used to get column and row number based on the column names (ex. <colname1>) and row values (ex.<rowvall>) specified in the keyword script if it is used with the perform keyword. This can also be used to verify whether the specified column

name and row value is present or not in the table if it is used with the Check keyword.

Note: The search can also be restricted by specifying the number of columns (<[no of columns]>) to search.

25. `prop_name:<variable_name>`

This is used to store the property value in the specified variable. It is used with the storevalue keyword.
26. `set:env_<Environment Name>`

This is used to assign the value, which is stored in environment variable to the edit field.
27. `set:dt_Parameter`

This is used to assign the value, which is given in the external test data sheet to the edit field.
28. `set:#<variable>`

This is used to assign the value, which is stored in the variable to the edit field.
29. `set:p_<parameter>`

This is used to assign the value, which is stored in the input parameter to the edit field.
30. `text:<text to search>`

This is a check operation that is used to verify whether a string is present or not in the table.
31. `tablecheck:<input sheet name>`

This is used to check the value of a table with an input sheet.
32. `search:<text>`

This is a check operation that is used to verify whether the value of the string is in current row of the object or not.
33. `search:#<variable name>`

This is a check operation that is used to verify a value stored in environment variable is in the current row of the combo box.
34. `text:blank`

This is a check operation that is used to verify whether a text field is blank.
35. `tableoperation:type:<Object type>:<value>:<Selection>`

This is an operation to set on or off an object present inside a table depending on the object type and value of that object.

36. `tableoperation:html tag:A:<Link Name>`
This is used to click on a link present inside a table, depending on the link value.
37. `RowNum:<rowval>--<[number of columns]>`
This is used to get the row number of a required string(<rowval>) specified in the keyword script from a table . The search can also be restricted by specifying the number of columns (<[number of columns]>).
38. `<conversiontype>:<variable name>:<format type>`
This is used to convert a variable from one data type to another.
39. `set:<text>:row:col:index`
'Set' is used to set the text field with the required string in the specified row and column in a table.
40. `submit:row:col:index`
This is used to submit the value entered in a `tabletextbox`, `tablecombobox`, `tableradiobutton`, etc. for a specified row and column in a table.
41. `Select:<name>:row:col:index`
This is used to select a value from a Weblist by specifying the name for a specified row and column in a table.
42. `deselect:<name>:row:col:index`
This is used to deselect an item from a Weblist for a specified row and column in a table.
43. `extendselect:<name>:row:col:index`
This is used to select an additional item in a Weblist for a specified row and column in a table.

(Note: Supported only if multi-selection is available for the Weblist.)
44. `prop_name:<variable_name>:row:col:index`
This is used get the value of the property of the specified row and column in a table and store it in the specified variable.
45. `Select:#<index>:row:col:index` or `Select index:<index>:row:col:index`
This is used to select the `tableradiobutton` (`radiobutton` in a table) for a specified row and column in a table.
46. `click:row:col:index`
This is used to click on a table image, table link, web table, etc. for a specified row and column in a table.

47. `enabled:True/False:row:col:index`

This is a check operation that is used to verify whether the object is enabled or disabled for a specified row and column in a table.
 48. `visible:True:row:col:index`

This is a check operation that is used to verify whether the object is visible or not for a specified row and column in a table.
 49. `selection:<name>:row:col:index`

This is a check operation that is used to verify whether the specified table Radiobutton is selected or not for a specified row and column in a table.
 50. `itemscount:<# items>:row:col:index`

This is a check operation that is used to verify the number of elements in the radio group for a specified row and column in a table.
 51. `tablecheck:<input sheet name>:row:col:index`

This is used to check whether the specified value is present or not in a table.
 52. `set:ON/OFF:row:col:index`

This is used to select or deselect a checkbox in a table for a specified row or column.
 53. `focused:True/False:row:col:index`

This is a check operation that is used to verify whether the object is focused or not for a specified row and column in a table.
 54. `Create;<Folder Path/Name>/<File Path/Name>`

This is used to create a folder/file in the specified path.
 55. `Delete;<Folder Path/Name>/<File Path/Name>`

This is used to delete a folder/file in the specified path.
 56. `Copy;<Source Path/Name>;<DestinationFolder Path/Name>/<Source File Path/Name>;<Destination Folder Path>`

This is used to copy a folder/file from source to destination path specified.
 57. `Move;<Source Path/Name>;<DestinationFolder Path/Name>/<Source File Path/Name>;<Destination Folder Path>`

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

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

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

59. 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.

60. 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.

61. 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 Output Checkpoint is the name of the checkpoint placed inside where many output values are captured.

2.4. Naming Conventions Used for Variables

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

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

Here, the value stored in varName is assigned to the textbox.

Note: Variable Naming Standards to be followed as below

StrVarName –

Ex. To store a username, which is a string value in a variable.

Storevalue	Textbox; Username	Text: StrUserName
------------	-------------------	-------------------

intVarName –

Ex. To store an order number, which is a integer value in a variable.

Storevalue	Textbox; OrderNo.	Text: intOrderNo
------------	-------------------	------------------

blnVarName –

Ex. To store an exit property value., which is a boolean value in a variable.

Storevalue	Textbox; OrderNo.	Exist: blnOrderProValue
------------	-------------------	----------------------------

2.5. Naming Convention Used for Datatable

To input a value to a field from a datatable, the naming convention used is dt.

Example:

Perform	Textbox;<textbox name>	Set: dt_<Parameter>
---------	------------------------	---------------------

Here, the ‘dt’ signifies to take the value from the datatable from the column “Parameter”.

2.6. Naming Convention Used for Parameter

To input a value to a field from a parameter, the naming convention used is P.

Example:

Perform	Textbox;<textbox name>	Set: P_<Parameter>
---------	------------------------	--------------------

Here the ‘P’ signifies to take the value from the parameter.

2.7. Naming Convention Used for Environment

To input a value to a field from an environment variable, the naming convention used is env.

Example:

Perform	Textbox;<textbox name>	Set: env_<VarName>
---------	------------------------	--------------------

Here the ‘env’ signifies to take the value from the environment variable ‘VarName’.

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.
