QTP Open Source Test Automation Framework Keyword Naming Conventions for Developers (Java)

Version 1.0

June 2009

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

This document provides the naming conventions followed in the Open Source Test Automation Framework for Java 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 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 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, 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. The scope of the environment variable is only for the current script.

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, and 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 data table.

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 Java objects in the Open Source Test Automation Framework.

	Java Object Class	Naming Conventions

1.	Browser	Browser
2.	Page	Page
3.	Frame	Frame
4.	JavaWindow	Window
5.	JavaDialog	Dialog
6.	Window	WinWindow
7.	Dialog	WinDialog
8.	JavaApplet	Applet
9.	JavaInternalFrame	Frame
10.	ActiveX	ActiveX
11.	JavaButton	Button
12.	JavaCalendar	Calendar
13.	JavaCheckBox	Checkbox
14.	JavaExpandBar	Expandbar
15.	JavaLink	Link
16.	JavaList	Listbox
17.	JavaMenu	Menu
18.	JavaObject	Object
19.	JavaRadiobutton	Radiobutton
20.	JavaSlider	Slider
21.	JavaSpin	Spinner
22.	JavaStaticText	Static
23.	JavaTab	Tab
21.	JavaTable	Table
22.	JavaEdit	Textbox
23.	JavaToolBar	Toolbar
23.	JavaTree	Treeview

Table 1: Naming Conventions Used for Objects

2.3. Naming Conventions Used for Actions

1. Click

‘Click’ is used to perform the click operation on objects.

(ex. Clicking a JavaButton, JavaLink, etc.)

2. Close

‘Close’ is used to perform the close operation on any open objects.

(ex. Closing a browser, window, dialog, etc).

3. Selectindex:<index>

‘Selectindex’ is used to select an item from a Listbox, Combobox, Tab, and ListView.

4. **Set:<Text>**

   ‘Set’ is used to assign a value to an edit field.

5. **Set:d_currenttime**

   This sets the current system time to the edit field.

6. **Set:d_currentdate**

   This sets the current system date to the edit field.

7. **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.

8. **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.

9. **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.

10. **Set**

    This is used to select or deselect a Radiobutton.

11. **Deselect:<name>**

    This is used to deselect a selected item in a list.

12. **Extendselect:<name>**

    This is used to select more than one item from a list.

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

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

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

    This is a check operation that is used to verify whether a Radiobutton whose name is specified is checked or not.

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

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

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

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

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

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

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

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

19. **ItemsCount:<Count>**

    This is a check operation that is used to verify the number of items in a Listbox.

20. **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.

21. **<PropertyName>:<variable_name>**

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

22. **ExtendSelectIndex:<index>**

    This is used to select more than one item from a list.

23. **DeSelectIndex:<index>**

    This is used to deselect an item from a list based on the index provided.

24. **SelectRangeIndex:<index1>:<index2>**

    This is used to select a list of items in the specified range

25. **Click:<X>:<Y>**

    This keyword is used to click on specific X and Y coordinates.

26. **SetDate:Now**

    This is used to set the current date in a calendar object.

27. **SetDate:Date**

    This is used to set a specific date in a calendar object.

28. **SetTime:Now**

    This is used to set the current time in a calendar object.

29. **SetTime:Date**

    This is used to set a specific time in a calendar object.

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

38. **Check**

    This is used to select the Checkbox. It is used with the perform keyword.

39. **Uncheck**

    This is used to deselect the Checkbox. It is used with the perform keyword.

40. **Tabexist:<Tabitemname>**

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

41. **Maximize**

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

42. **Minimize**

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

43. Restore

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

44. Activate

This is used to activate a particular window. It is used with perform keywords

45. Text:p_<parameter name>

This is used to check the value, which is stored in the input parameter with the text displayed in the edit field.

46. Text:env_<Environment Name>

This is used to check the value, which is stored in an environment variable with the text displayed in the edit field.

47. Select:<name/Item>

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

48. ItemExist:<Item>

This is a Check operation that is used to verify whether an item is present in the Listbox object.

49. Type:<Value/Item>

This is used to type the required value in a TextBox, Object, Editor, or ComboBox. This is used with the perform keyword.

50. DoubleClick:<X>:<Y>

This keyword is used to double click on specific X and Y coordinates in a TextBox. This is used along with the perform keyword.

51. DoubleClick:<name>

This keyword is used to double click on a specific item in a ListView. This is used along with the perform keyword.

52. NextLine[:<line number>]

This is a Perform operation to scroll to the next line number.

53. PrevLine[:<line number>]

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

54. NextPage[:<line number>]

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

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

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

56. **Set:<Value>**

This is a Perform operation to scroll/spin to the required value.

57. **SelectRange:<item1>:<item2>**

This is used to select the list of items from a list.

58. **SelectRangeIndex:<item1>:<item2>**

This is used to select the list of items from a list.

Activate:<item>

59. **Activate:<item>**

This is used to activate an item in a listbox or listview.

60. **Next**

This is a Perform operation to spin to the next value.

61. **Previous**

This is a Perform operation to spin to the previous value.

62. **Set:<Value>**

This is a Perform operation to scroll/spin to the required value.

63. **closetab:<tabname>**

This is a perform operation that is used to close the required tab.

64. **RowCount:<value>**

This is a check operation that is used to verify the row count in a table.

65. **Press:<name>**

This is a perform operation that is used to press on the required item in the toolbar.

66. **Expand:<name>**

This is a perform operation that is used to expand the required tree item.

67. **ExpandAll:<name>**

This is a perform operation that is used to expand all the required tree items.

68. **Click:<name>**

This is a perform operation that is used to click on the required tree item.

69. **Check:<node path>**

This is a perform operation that is used to check the required checkbox in the tree item.

70. **UnCheck:<node path>**

This is a perform operation that is used to uncheck the required checkbox in the tree item.

71. **SetCaretPos:<row>:<column>**

This used to position the caret depending upon the specified row and column number. This is used along with the object Textbox.

72. **SetCaretPos:<line number>:<column number>**

This is used to position the caret depending upon the specified line number and column number. This is used along with the object Editor.

73. **Check:<name>**

This is a perform operation that is used to check the required checkbox in the list view.

74. **UnCheck:<name>**

This is a perform operation that is used to uncheck the required checkbox in the list view.

75. **ItemClick:<item>**

This is a perform operation that is used to click on the required item in the list view.

76. **Click:<name>**

This is a perform operation that is used to click on the required item in the list view.

77. **Select**

This is a perform operation that is used to select the required menu item.

78. **setselection:<LineFrom>:<ColumnFrom>:<LineTo><ColumnTo>**

This is used to select the required text depending upon the specified line number and column number. This is used along with the object Editor.

79. **TextClick:<text>**

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

80. **WindowText:<Text>:<True/False>**

This is used to check for the existence or non-existence of text in a window.

81. text:<text to search>

This is a check operation that is used to verify whether a string is present or not in the table.

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 of property 'Prop_name' for the textbox is stored in the 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 an 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 data table 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'.