



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

Version 1.0

May 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	3
2. KEYWORD NAMING CONVENTIONS.....	4
2.1. Naming Conventions Used for Keywords	4
2.2. Naming Conventions Used for Objects	5
2.3. Naming Conventions Used for Actions	6
2.4. Naming Conventions Used for Variables	12
2.5. Naming Convention Used for a Datatable	12
2.6. Naming Convention Used for Parameter	12
2.7. Naming Convention Used for Environment	13

1. Purpose of the Document

This document provides the naming conventions followed in the Open Source Test Automation Framework for .NET 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 .NET window. This keyword brings a particular Window to the current context, so that any operation or checking can be performed on that particular Window.

3. **Perform**

'Perform' is used to perform an operation on a particular object such as clicking on a button, selecting an option from combobox, 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, check properties, verify the existence of windows, etc.

6. **CallFunction**

'Call Function' is used to call any declared function that 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, or 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 .NET objects in the Open Source Test Automation Framework.

	.Net Object Class	Naming Conventions
1.	Dialog	Dialog
2.	Page	Page
3.	Window	popupwindow
4.	SwfWindow	Window

5.	vbWindow	vbWindow
6.	SwfListBox	SwfList
7.	SwfButton	Button
8.	SwfCheckBox	Checkbox
9.	SwfComboBox	Combobox
10.	SwfEdit	Textbox
11.	SwfRadioButton	Radiobutton
12.	SwfLink	Link
13.	SwfImage	Image
14.	SwfTable	Grid
15.	SwfElement	Element
16.	SwfTreeView	TreeView
17.	SwfListView	ListView
18.	SwfObject	Object
19.	SwfStatusBar	StatusBar
20.	SwfLabel	Label
21.	SwfTab	Tab
22.	SwfEditor	Editor
23.	SwfSpin	Spinner
24.	SwfToolBar	ToolBar
25.	SwfScrollBar	ScrollBar

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 SwfButton, SwfLink, or SwfElement)

2. Close

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

(ex: Closing a SwfWindow or Dialog box).

3. Select:<name>

'Select' is used to select an item from SwfCombobox or SwfRadiobutton.

4. Selectindex:<index>

'Selectindex' is used to select an item from a SwfListbox or SwfCombobox.

5. Set:<Text>

'Set' is used to assign a value to an 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 SwfRadiobutton.

12. Activate

'Activate' is used to activate the window.

13. Deselect:<name>

This is used to deselect a selected item from 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. Clickitem:<item name>

This is used to click an item in the listview.

17. Checked:<ON/OFF>

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

18. Enabled:<True/False>

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

19. Exist:<True/False>

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

20. Focused:<True/False>

This is a check operation that is used to verify whether the specified 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>; <rowval1>:: <colname2>; <rowval2>--<[no of columns]>:

This is used to get the column and row number based on the column names (ex: <colname1>) and row values (ex: <rowval1>) 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. Deselectindex:<item_index>

This is used to de-select an item from the SwfListBox or SwfComboBox.

33. text:blank

This is a check operation that is used to verify whether a text field is blank.

34. 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.

35. tableoperation:html tag:A:<Link Name>

This is used to click on a link present inside a table, depending on the link value.

36. 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>).

37. <conversiontype>:<variable name>:<format type>

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

38. Doubleclick

'DoubleClick' is used to double click on an object.

(Ex: Double clicking on a textbox).

39. Expand:<node_name> or Expandall<node_name>

This is used to expand the specified TreeNode or expand the specified TreeNode and its child nodes to display all of its child nodes.

40. Extendselectindex:<item_index>

This is used to select an additional item from the multi-selection list for the specified index value.

(Note: This is supported only if multi-selection is available for the SwfList.)

41. Getitems

This is used to get the menu names and store them in the Action sheet.

42. Grid:<empty/notempty>

This is a check operation used to verify whether the SwfTable is empty or not.

43. GridText:<value>

This is a check operation used to verify if the text(value) provided is present in the SwfTable.

44. ItemCount:<value>

This is a check operation used to verify the number of items in the object's list.

(Ex: Check for the number of items in a combobox, listbox, treeview, or listview).

45. ItemExist:<itemname>

This is a check operation used to verify whether the specified item is present in the object's list.

(Ex: Check for the specified item in a combobox based on the name).

46. Maximize or Minimize

This is used to maximize or minimize a window.

47. Next

This is a perform operation used to set the spin object to its next value.

48. NextLine:<line_no>

This is used to move the scroll bar downward or to the right a specified number of lines.

49. Previous:<line_no>

This is used to move the scroll bar upward or to the left a specified number of lines.

50. Press:<toolbar_button_name>

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

51. SelectRangeIndex:<item_index1>:<item_index2>

This is a perform operation used to select all the items from the list between (and including) two specified items based on the index.

(Ex: Select all the items in the listview or listbox between the range specified).

52. SetCaretPos:<line>:<col>

This is a perform operation used to place the cursor at the specified point in a multi-line edit box.

53. Setselection:<from_col>:<to_col>

This is a perform operation used to select the text in a single line edit box.

54. Showdropdown:<toolbar_button_name>

This is used to open the dropdown menu associated with the toolbar button.

55. Tabnotexist:<value>

This is a check operation used to verify that the specified tab name does not exist.

56. Check

This is a perform operation used to check the checkbox.

57. Uncheck

This is a perform operation used to uncheck or clear the checkbox.

58. Uncheck:<node_name>

This is a perform operation used to uncheck or clear an item in the treeview.

59. Create;<Folder Path/Name>/<File Path/Name>

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

60. Delete;<Folder Path/Name>/<File Path/Name>

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

61. 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.

62. 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 specified.

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

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

64. 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.

65. 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.

66. 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 it, 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 are shown 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 a 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.