



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

Version 1.0

April 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	4
2. KEYWORD NAMING CONVENTIONS	5
2.1. Naming Convention used for Keywords	5
2.2. Naming Conventions used for Objects	6
2.3. Naming Conventions Used for Actions	7
2.4. Naming Conventions used for Variables	11
2.5. Naming Convention used for Datatable	12
2.6. Naming Convention used for Parameter	13
2.7. Naming Convention used for Environment	13

LIST OF TABLES

Table 1: Naming Conventions used for Objects 7

1. Purpose of the Document

This document provides the naming conventions followed in the Open Source Test Automation Framework for Window Technology.

2. Keyword Naming Conventions

2.1. Naming Convention 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 can be performed on that particular window or dialog.

3. **Perform**

'Perform' is used to perform an operation on a particular object (ex: clicking on a button, closing an open window, 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, 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 different .vbs files.

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 QTP.

2.2. Naming Conventions used for Objects

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

Sl.No	Window Object Class	Naming Conventions used in the Open Source Test Automation Framework
1.	Window	Window
2.	Dialog	Dialog
3.	WinButton	Button
4.	WinCheckBox	Checkbox
5.	WinList	Listbox
6.	WinEdit	Textbox
7.	WinRadioButton	Radiobutton
8.	WinSpin	Spinner
9.	WinToolBar	Toolbar
10.	WinTreeView	Treeview
11.	WinListView	Listview
12.	WinMenu	Menu
13.	WinObject	Object
14.	WinEditor	Editor
15.	Static	Static
16.	WinStatusBar	Statusbar
17.	WinScrollBar	Scrollbar
18.	WinTab	Tab
19.	ActiveX	ActiveX
20.	WinCombobox	Combobox

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. 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, or 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 a edit field. It is used with the perform keyword.

9. Type:<Value>

'Type' is used to assign a value to a 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. 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.
- 20. Checked:<On/OFF/Dimmed>**

This is a Check operation that is used to verify whether a Radio Button is selected or not.
- 21. Checked:<On/OFF>**

This is a Check operation that is used to verify whether a Checkbox is checked or not.
- 22. Enabled:<True/False>**

This is a Check operation that is used to verify whether the given window object is enabled or not.
- 23. Exist:<True/False>**

This is a Check operation that is used to verify whether the window object whose name is specified exists or not.
- 24. Focused:<True/False>**

This is a Check operation that is used to verify whether the object is focused or not.
- 25. ItemCount:<Item>**

This is a Check operation that is used to verify the number of items present or not in a window object.
- 26. 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.
- 27. Prop_name:<variable_name>**

This is used to store the property value in the specified variable. It is used with storevalue keyword.
- 28. 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.

29. **Check**
This is used to select the Checkbox. It is used with the perform keyword.
30. **Uncheck**
This is used to deselect the Checkbox. It is used with the perform keyword.
31. **Tabexist:<Tabitemname>**
This is a Check operation that is used to verify whether the tab item specified is present or not.
32. **ItemExist:<Item name>**
This is a Check operation that is used to verify whether an item is present or not in the window object.
33. **DoubleClick**
'DoubleClick' is used to perform the doubleclick operation on objects. It is used with the perform keyword in keyword scripts.
34. **Press:<name>**
This is a Perform operation to Click on the specified toolbar item.
35. **Expand:<Item name>**
This is used to expand the treeitem specified. It is used with the perform keyword in keyword scripts.
36. **ExpandAll:<item name>**
This is used to expand all the treeitems in a treeview. It is used with the perform keyword in keyword scripts.
37. **<conversiontype>:<variable name>:<format type>**
This is used to convert a variable from one data type to another.
38. **Collapse:<name>**
This is used to collapse the treeitem specified. It is used with the perform keyword in keyword scripts.
39. **SelectRange:<item name1>:<item name2>**
This is used to select the range of items in a listview.
40. **NextLine[:<line number>]**
This is a Perform operation to scroll to next line number.
41. **PrevLine[:<line number>]**
This is a Perform operation to scroll to the previous line number.
42. **NextPage[:<line number>]**
This is a Perform operation to scroll to the next page.

-
43. **PrevPage[:<line number>]**
This is a Perform operation to scroll to the previous page.
44. **Create;<Folder Path/Name>/<File Path/Name>**
This is used to create a folder/file in the specified path.
45. **Delete;<Folder Path/Name>/<File Path/Name>**
This is used to delete a folder/file in the specified path.
46. **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.
47. **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.
48. **Write;<File Path/Name>;<The value to be entered>**
This is used to write the file with the data mentioned in the specified path.
49. **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.
50. **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.
51. **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.
52. **TextClick:<text>**
This is used to click on the specified text in the window.

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 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 a Order No., which is a integer value in a Variable.

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

blnVarName -

Ex. To Store a 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.