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

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** **3**

2. **Keyword Naming Conventions** **4**

2.1. Naming Conventions Used for Keywords 4

2.2. Naming Conventions Used for Objects 6

2.3. Naming Conventions Used for Variables 12

2.4. Naming Convention Used for Datatable 13

2.5. Naming Convention Used for Parameter 13

2.6. Naming Convention Used for Environment 13

1. Purpose of the Document

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

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 window, a session, or a dialog. 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, check 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 as well as in scripts.

8. **PressKey**

   'Press Key' is used to press 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 or values stored in a particular variable 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.

20. **Split**

‘Split’ is used to split a string (statement) based on the delimiter.

21. **Strreverse**

‘Strreverse’ is used to reverse a string.

22. **ImportData**

Importdata’ is used to import the external test data sheet into the Action1 sheet of QTP.

23. **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.

2.2. Naming Conventions Used for Objects

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

	SAP Object Class	Naming Conventions
1.	SAPGuiWindow	Window
2.	Dialog	Dialog
3.	SAPButton	Button*
4.	SAPGuiButton	Button*
5.	SAPGuiCheckBox	Checkbox*
6.	SAPCheckBox	Checkbox*
7.	SAPList	Listbox
8.	SAPEdit	Textbox*
9.	SAPGuiEdit	Textbox*
10.	SAPGuiRadioButton	Radiobutton*
11.	SAPRadioButton	Radiobutton*
12.	SAPGuiToolBar	Toolbar
13.	SAPGuiTree	Treeview*
14.	SAPTreeView	Treeview*
15.	SAPMenu	Menu*
16.	SAPGuiMenubar	Menu*
17.	SAPGuiTextArea	Editor
18.	Static	Static
19.	SAPStatusBar	Statusbar*
20.	SAPGuiStatusBar	Statusbar*
21.	SAPTabStrip	Tabstrip*
22.	SAPGuiTabStrip	Tabstrip*
23.	ActiveX	ActiveX
24.	WinCombobox	Combobox
25.	SAPGuiSession	Session
26.	SAPGuiAPOGrid	Apogrid
27.	SAPCalendar	Calendar*
28.	SAPGuiCalendar	Calendar*
29.	SAPOKCode	Codebox*
30.	SAPGuiOKCode	Codebox*
31.	SAPFrame	Frame

32.	SAPNavigationBar	Navigationbar
33.	SAPDropDownMenu	dropdownmenu
34.	SAPGuiElement	Element
35.	SAPGuiGrid	Grid
36.	SAPTable	Table*
37.	SAPGuiTable	Table*
38.	SAPPortal	Portal
39.	SAPiView	iView

Table 1: Naming Conventions Used for Objects

*Based on the context setting (either SAPGui or SAP),the corresponding SAP web-based or SAP windows-based objects will be called.

Naming Conventions used for the commonly used actions are shown below:

1. **Click**

   ‘Click’ is used to perform the click operation on objects. (ex. Clicking a SAPGuiButton or SAPGuiElement, etc.)

2. **Close**

   ‘Close’ is used to perform the close operation on any open objects. (ex. Closing a SAPGuiWindow or dialog box, etc).

3. **Select:<name>**

   ‘Select’ is used to select an item from combobox, radiobutton, tabStrip, treeView, menu, or navigationbar.

4. **Selectindex:<index>**

   ‘Selectindex’ is used to select an item from a list, Combobox, or tabstrip object.

5. **Activate:<item>**

   This is used to activate an item in a tree object.

6. **Exist:<variable>**

   ‘Exist’ is used to store the status of the object, whether the object exists or not, in a variable.

7. **Collapse:<name>**

   ‘Collapse’ is used to collapse all items under the specified node of the navigationbar or the treeview object.

8. **Selectcolumn:<column_name>**

   This is used to click a column in a treeview object.

9. **Type:<value>**

   This is used to type the specified value in the element.

10. **Setsecure:<value>**

    'Setsecure' sets the encrypted value to an edit field.

11. **Set:<Text>**

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

12. **Set:d_currenttime**

    This sets the current system time to the edit field.

13. **Set:d_currentdate**

    This sets the current system date to the edit field.

14. **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.

15. **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.

16. **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.

17. **Set: dt_<value>**

    This is used to set a database parameter value to an edit field.

18. **Set:<On/Off>**

    This is used to select or deselect a radiobutton.

19. **Set**

    This is used to select a radiobutton.

20. **SetDate:<Date>**

    This is used to set a specified date to an edit field.

21. **SetDate:Date**

    This is used to set the current date to an edit field.

22. **Set:Now**

    This is used to set the current date and time to an edit field.

23. **Setvalue:r:c:variable**

    This is used to set the value present in the variable onto the specified row and column of the table.

24. **Setindexvalue:row:column index:variable**

This is used to set the value present in the variable onto the specified row and column of the table.

25. **Selectrow:columnname:cellvalue**

This is used to select the specified row from the table.

26. **Activate**

'Activate' is used to activate the window.

27. **Setfocus**

This is used to set the focus on the specified object.

28. **Rowsearch:<column>:<value>:intx**

This is used to search for the row in the grid based on the column name and value and store the row number into the variable(intx).

29. **Getrow:columname:cellvalue:<variable>**

This is used to search for the row in a table based on the column name and cell value and store the row number into the variable.

30. **Deselect:<value>**

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

31. **Extendselect:<value>**

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

32. **Selection:<item>**

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

33. **Checked:<ON/OFF>**

This is a check operation used to verify whether the radiobutton or checkbox is selected or not.

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

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

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

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

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

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

37. **ItemCount:<Item>**

This is a check operation used to verify the number of items in a combobox or statusbar.

38. **Text:<text/#Variable_Name>**

This is a check operation used to verify whether the required text is present or not in the object.

39. **TableSearch:<colname1>; <rowval1>:: <colname2>; <rowval2>**

If it is used with the perform keyword, 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 check keyword, it is used to verify whether the specified column name and row value is present or not in the table. If it is used with the storevalue keyword, it is used to store the status of the search performed in an environment variable.

40. **prop_name:<variable_name>**

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

41. **set:env_<Environment Name>**

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

42. **set:dt_Parameter**

This is used to assign the value, which is given in the external test data sheet, to the edit field.

43. **set:#<variable>**

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

44. **set:p_<parameter>**

This is used to assign the value, which is stored in the input parameter, to the edit field.

45. **text:p_<parameter name>**

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

46. **text:env_<environment name>**

This is a check operation used to verify whether a string specified in an environment variable is present or not in the table.

47. **text:dt_<parameter name>**

This is a check operation used to verify whether a string specified in the action sheet variable is present or not in the table.

48. **<conversiontype>:<variable name>:<format type>**

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

49. **Doubleclick**

‘DoubleClick’ is used to double click on an object.

(Eg: Double clicking on a textbox).

50. **Expand:<node_name>**

    This is used to expand the specified tree node to display all of its child nodes.

51. **ItemExist:<item>**

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

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

52. **Maximize or Minimize**

    This is used to maximize or minimize a window.

53. **Restore**

    This is used to restore a window or a dialog box.

54. **Hide/ Show**

    This is used to hide or show the iPanel of the navigationbar.

55. **Press:<name>**

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

56. **SetCaretPos:<location>**

    This is a perform operation used to place the cursor at the specified point in a label.

57. **Setselection:<from_col>:<to_col>**

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

58. **Check**

    This is a perform operation used to check the checkbox.

59. **Uncheck**

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

60. **Create;<Folder Path/Name>/<File Path/Name>**

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

61. **Delete;<Folder Path/Name>/<File Path/Name>**

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

62. **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.

63. **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.

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

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

65. **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.

66. **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.

67. **OutputCheckPointName**

This is used for capturing multiple values from the database.

OutputCheckpointName is the name of the output checkpoint placed, inside where many output values are captured and stored in variables specified in the output checkpoint.

68. **Commit: <yes/no>**

This is used to commit the changes to the database if 'yes' is specified. It is used with the SQL execute keyword.

69. **CheckPointName**

This is used to check if the required data is present in the database. It is used with the SQLCheckpoint keyword.

2.3. 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 in a variable, which is a string value:

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

2.4. 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”.

2.5. 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.

2.6. 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’.