Open2Test Test Automation Framework Keyword Naming Conventions for Developers (QTP - PowerBuilder)

Version 1.0

April 2011

DISCLAIMER

TABLE OF CONTENTS

LIST OF TABLES

1. Purpose of the Document

This document provides the naming conventions followed in Open2Test Test Automation Framework for the PowerBuilder 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 PowerBuilder window or a dialog object. This keyword brings a particular PowerBuilder 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 PowerBuilder 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.

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 PowerBuilder objects in the Open Source Test Automation Framework.

	PowerBuilder Object Class	Naming Conventions
1.	PbWindow	Window
2.	PbDataWindow	DataWindow

3.	WinButton	WinButton
4.	PbButton	Button
5.	WinEdit	WinEdit
6.	WinMenu	WinMenu
7.	WinScrollBar	WinScrollBar
8.	PbScrollbar	Scrollbar
9.	PbCheckBox	Checkbox
10.	PbCombobox	Combobox
11.	PbEdit	Textbox
12.	PbList	List
13.	PbListview	Listview
14.	PbRadioButton	Radiobutton
15.	PbTabstrip	Tabstrip
16.	PbToolbar	Toolbar
17.	PbTreeview	Treeview
18.	PbObject	Object
19.	Dialog	Dialog

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 Button etc.)

2. **Close**

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

   (ex. Closing a 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 List or Combobox.

5. **Set:<Text>**

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

6. **Setcelldataindex:<rownumber>:<columnnumber>:<Text>**

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

7. **Set:d_currenttime**

This sets the current system time to the edit field.

8. **Set:d_currentdate**

   This sets the current system date to the edit field.

9. **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.

10. **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.

11. **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.

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

    This is used to select or deselect a Radiobutton.

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. **Selectcellname:<rownumber>:<columnnumber>**

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

17. **Selectrow:<rownumber>**

    This is used to select a row of a datawindow.

18. **Checked:<Name>**

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

19. **Checked:<True/False>**

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

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

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

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

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

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

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

23. **Visible:<True/False>**

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

24. **ItemsCount:<Item>**

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

25. **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.

26. **prop_name:<variable_name>**

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

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

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

28. **set:dt_Parameter**

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

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

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

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

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

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

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

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

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

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

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

38. **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.

39. **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.

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

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

41. **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.

42. **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.

43. **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’.