



**Open2Test Test Automation
Framework for SilkTest – Scripting Standards for
Java**

Version 1.0

January 2010

DISCLAIMER

Verbatim copying and distribution of this entire article is permitted worldwide, without royalty, in any medium, provided this notice is preserved.

TABLE OF CONTENTS

1.	Introduction.....	3
1.1	Purpose	3
2.	Standards for Keyword Scripting.....	4
2.1	Getting Started.....	4
2.2	Column Description.....	5
2.2.1.	Automate (Column 'A').....	5
2.2.2.	Action (Column 'B').....	6
2.2.3.	Object (Column 'C').....	10
2.2.4.	ActionValue1 (Column 'D')	12
2.2.5.	ActionValue2 (Column 'E')	18
2.2.6	Comments (Column 'F')	18
2.2.7	Delimiters.....	19
2.2.8.	Variables.....	20
3.	Sequence of Keywords.....	21
3.1.	Use of Keyword 'Context'.....	21
3.2.	Use of 'Conditional Statements'	22

TABLE OF FIGURES

Figure 1	Calling the Framework.....	5
Figure 2	Keyword driven script.....	5
Figure 3	Column 'Automate'.....	6
Figure 4	Column 'Action'.....	7
Figure 5	Column 'Object'.....	11
Figure 6	Column 'Actionvalue1'.....	13
Figure 7	Column 'Actionvalue2'.....	18
Figure 8	Column 'Comments'.....	19
Figure 9	Keyword 'Context'.....	22

1. Introduction**1.1 Purpose**

This document provides details about the various columns used, the

keywords and their descriptions, as well as some methodologies that need to be followed while scripting using keywords.

2. Standards for Keyword Scripting

2.1 Getting Started

Before going into the details about the columns used for keyword scripting, the user should be familiar with what is known as the 'keyword script' and how to call the framework from the test script.

As shown in the figure below, the keyword script is the actual automation test script that corresponds to the manual test case. It is written in the external data sheet (.xls). In the DataDriven script file (g.t), the framework is called using 'Keyword_Driver(<Parameters>)'.

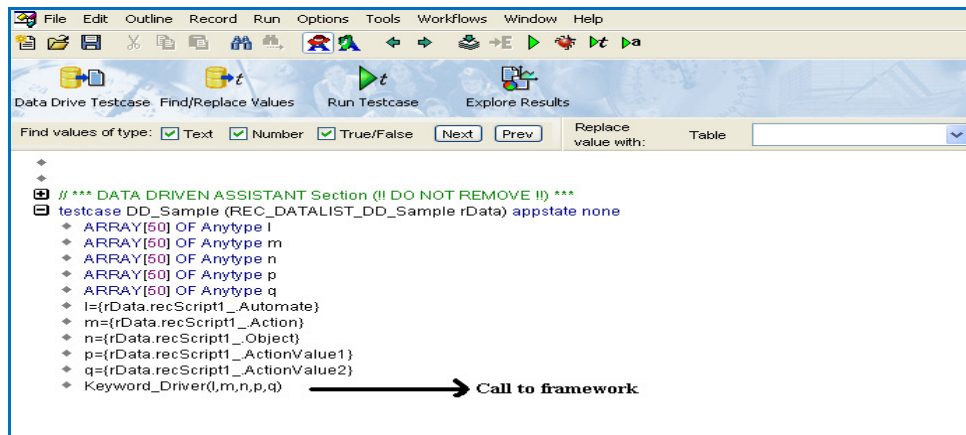
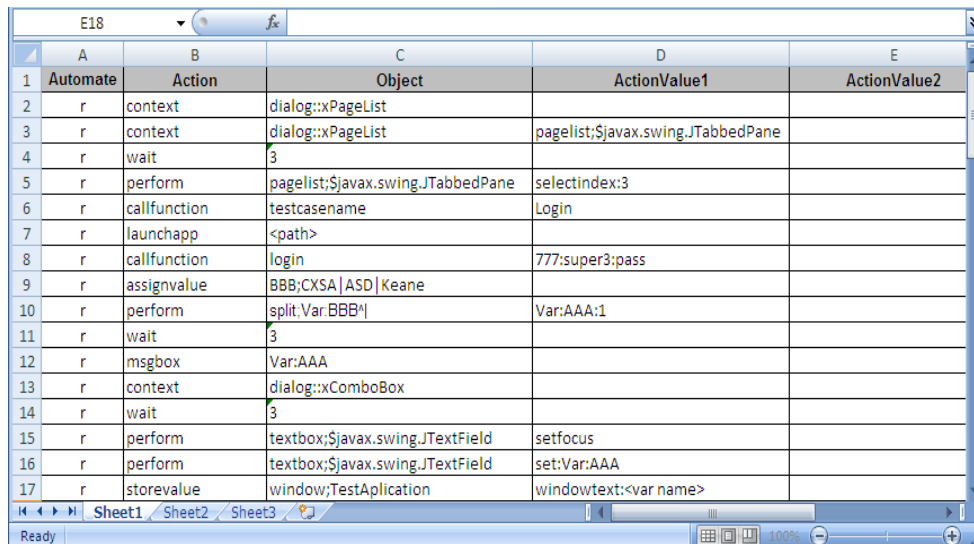


Figure 1 Calling the Framework



	A	B	C	D	E
1	Automate	Action	Object	ActionValue1	ActionValue2
2	r	context	dialog::xPageList		
3	r	context	dialog::xPageList	pagelist;\$javax.swing.JTabbedPane	
4	r	wait	3		
5	r	perform	pagelist;\$javax.swing.JTabbedPane	selectindex:3	
6	r	callfunction	testcasename	Login	
7	r	launchapp	<path>		
8	r	callfunction	login	777:super3:pass	
9	r	assignvalue	BBB;CXSA ASD Keane		
10	r	perform	split,Var.BBB^	Var:AAA:1	
11	r	wait	3		
12	r	msgbox	Var:AAA		
13	r	context	dialog::xComboBox		
14	r	wait	3		
15	r	perform	textbox;\$javax.swing.JTextField	setfocus	
16	r	perform	textbox;\$javax.swing.JTextField	set:Var:AAA	
17	r	storevalue	window;TestApplication	windowtext:<var name>	

Figure 2 Keyword-Driven Script

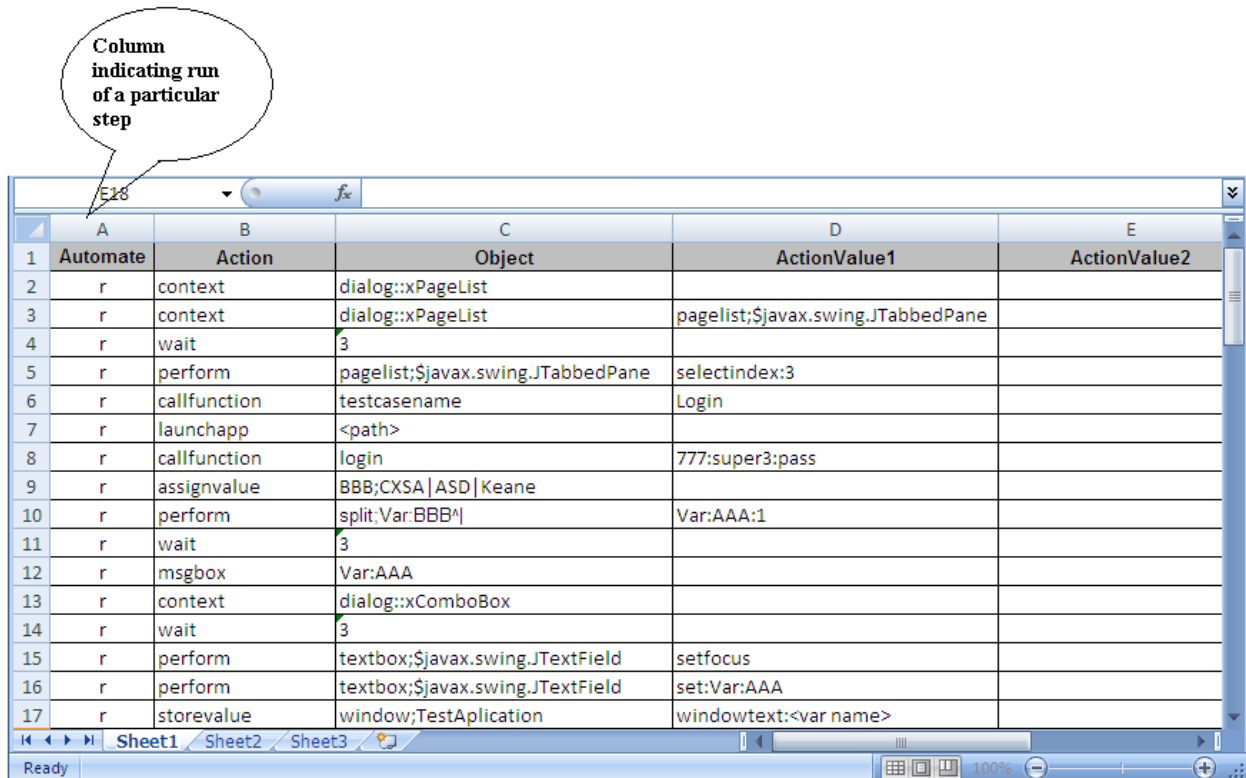
2.2 Column Description

This section gives a description of the columns used for keyword scripting.

2.2.1. Automate (Column 'A')

The data in the 'Automate' column decides whether the current step in the test case is to be run (automated) or not. This column has the letter 'r', which denotes that the current step should be run. If any step in

the test script is not required to be executed, then the corresponding row in the first column is to be left blank. The steps will run only based on the data in this column.



	A	B	C	D	E
1	Automate	Action	Object	ActionValue1	ActionValue2
2	r	context	dialog::xPageList		
3	r	context	dialog::xPageList	pagelist;\$javax.swing.JTabbedPane	
4	r	wait	3		
5	r	perform	pagelist;\$javax.swing.JTabbedPane	selectindex:3	
6	r	callfunction	testcasename	Login	
7	r	launchapp	<path>		
8	r	callfunction	login	777:super3:pass	
9	r	assignvalue	BBB;CXSA ASD Keane		
10	r	perform	split;Var:BBB^	Var:AAA:1	
11	r	wait	3		
12	r	msgbox	Var:AAA		
13	r	context	dialog::xComboBox		
14	r	wait	3		
15	r	perform	textbox;\$javax.swing.JTextField	setfocus	
16	r	perform	textbox;\$javax.swing.JTextField	set:Var:AAA	
17	r	storevalue	window;TestApplication	windowtext:<var name>	

Figure 3 Column 'Automate'

2.2.2. Action (Column 'B')

The second column of the keyword script is used to indicate the generic type of action being performed on the application under test (AUT). The action column is provided for different types of actions that are to be performed on a particular object.

Column to indicate the type of action.

	A	B	C	D	E
1	Automate	Action	Object	ActionValue1	ActionValue2
2	r	context	dialog::xPageList		
3	r	context	dialog::xPageList	pagelist;\$javax.swing.JTabbedPane	
4	r	wait	3		
5	r	perform	pagelist;\$javax.swing.JTabbedPane	selectindex:3	
6	r	callfunction	testcasename	Login	
7	r	launchapp	<path>		
8	r	callfunction	login	777:super3:pass	
9	r	assignvalue	BBB;CXSA ASD Keane		
10	r	perform	split;Var.BBB^	Var:AAA:1	
11	r	wait	3		
12	r	msgbox	Var:AAA		
13	r	context	dialog::xComboBox		
14	r	wait	3		
15	r	perform	textbox;\$javax.swing.JTextField	setfocus	
16	r	perform	textbox;\$javax.swing.JTextField	set:Var:AAA	
17	r	storevalue	window;TestApplication	windowtext:<var name>	

Figure 4 Column 'Action'

The keywords that can be used in this column are:

1. LaunchApp

'LaunchApp' is used to launch the application under test (AUT). This keyword triggers the driver script to launch the application from a specified folder (the location specified in the third column) or to launch the application that is already synchronized with Microfocus SilkTest.

2. Context

'Context' is used on a window/dialog object. This keyword brings a particular window/dialog to the current context, so that any operation or checking can be performed on that particular window/dialog.

3. Perform

'Perform' is used to perform an operation on a particular object, such as clicking on a button, closing an open window, typing a text in a textbox, etc. This keyword should be entered in the corresponding row in the second column.

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' provides a feature for comparing two variables, checking properties, checking for the existence of windows, etc.

6. **CallFunction**

'CallFunction' is used to call any user-defined function that is used in a particular script. These functions should be declared in a different .inc file.

7. **Storevalue**

'Storevalue' is used to store the property values of objects in different variables. These variables can be used in later parts of the script.

8. **PressKey**

'PressKey' is used to pass hot keys such as Enter, F3, F10, 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 two types: i) Pass, ii) Fail.

11. **Strsearch**

'Strsearch' is used to search for a 'sub string' inside a 'mainstring'.

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. **Assignvalue**

'Assignvalue' is used to assign dynamically generated values from the application to the variables.

16. **Loop**

'Loop' is used to loop a set of actions given in the data table.

17. **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.

18. **ExportResult**

'ExportResult' is used to export the result to the specified Excel sheet. The user must specify the path of the Excel sheet in the third column of the Excel sheet.



A detailed description of the keywords is provided in the Keyword Reference Document.

2.2.3. Object (Column 'C')

The third column of the keyword script sheet is used to indicate the object on which a particular type of action is to be performed. The object class and object tag are specified in column C (as shown in example 1).

The tag name of the object is mentioned in the keyword script sheet; it is preceded by \$.

The object column or column 'C' contain all the required details for an object (viz. class to which the objects belong to and the object tag name) on which various operations and validations are to be performed.

Example 1.

Action	Object
perform	button;\$javax.swing.JButton[2]
perform	textbox;\$javax.swing.JTextField

In the above example, the object column indicates that some operation has to be performed on an object of class 'button' having the tagname '\$javax.swing.JButton[2]'. Similarly, in the next line some operation has to be performed on an object of class 'textbox' having the tagname '\$javax.swing.JTextField'.

Example 2.

Action	Object
context	dialog::xPageList
perform	textbox;\$javax.swing.JTextField

In the above example, the object column indicates that some operation has to be performed on an object of class 'dialog' having the dialog name 'xPageList'.

In the above case, we use the dialog name and not the tag name. This is only for operations on window and dialog.



The object and its tagname are usually separated by a delimiter ';' as shown in the above example. (Delimiters will be covered in a later topic.)

Column to indicate the object.

	A	B	C	D	E
1	Automate	Action	Object	ActionValue1	ActionValue2
2	r	context	dialog::xPageList		
3	r	context	dialog::xPageList	pagelist;\$javax.swing.JTabbedPane	
4	r	wait	3		
5	r	perform	pagelist;\$javax.swing.JTabbedPane	selectindex:3	
6	r	callfunction	testcasename	Login	
7	r	launchapp	<path>		
8	r	callfunction	login	777:super3:pass	
9	r	assignvalue	BBB;CXSA ASD Keane		
10	r	perform	split;Var.BBB^	Var:AAA:1	
11	r	wait	3		
12	r	msgbox	Var:AAA		
13	r	context	dialog::xComboBox		
14	r	wait	3		
15	r	perform	textbox;\$javax.swing.JTextField	setfocus	
16	r	perform	textbox;\$javax.swing.JTextField	set;Var:AAA	
17	r	storevalue	window;TestApplication	windowtext:<var name>	

Figure 5 Column 'Object'

The objects that are commonly used are:

Sl.No	Objects used in the Open2Test Test Automation Framework	Java Object Class
1	Window	JavaJFCWindow
2	Dialog	JavaJFCDialog
3	Button	JavaJFCButton
4	Checkbox	JavaJFCCheckBox
5	Listbox	JavaJFCList
6	Textbox	JavaJFCTextField
7	Radiobutton	JavaJFCRadioButton
9	Toolbar	JavaJFCToolBar
10	Treeview	JavaJFCTreeView
11	Listview	JavaJFCListView
12	Menu	JavaJFCMenu
15	Tab	JavaJFCPagelist
16	Slider	JavaJFCSlider
17	Scrollbar	JavaJFCScrollBar
20	Combobox	JavaJFCCombobox
21	WinListBox	JavaJFCListBox
22	WebListBox	WebListBox
23	Static	JavaJFCStatic
25	Table	JavaJFCTable
26	Togglebutton	JavaJFCTogglebutton
27	Popupmenu	JavaJFCPopupmenu
28	Submenu	JavaJFCSubmenu
29	Tablecombobox	JavaJFCTablecombobox



A detailed description of the keywords is given in the Keyword Reference Document.

2.2.4. ActionValue1 (Column 'D')

The fourth column of the keyword script indicates the specific action being performed on the object present in the AUT. It contains the

details of all the operations or verifications that have to be performed on the objects listed in the 'Objects' column.

Consider the example of the object 'JavaJFCButton' with the name OK.

One of the actions that can be performed on a JButton would be Click. So in column 4, the above operation is put in the keyword form as "Click".

Example 1: The keyword syntax to perform the "Click" operation on the button is as follows:

Action	Object	ActionValue1
perform	button;<button tag name>	click

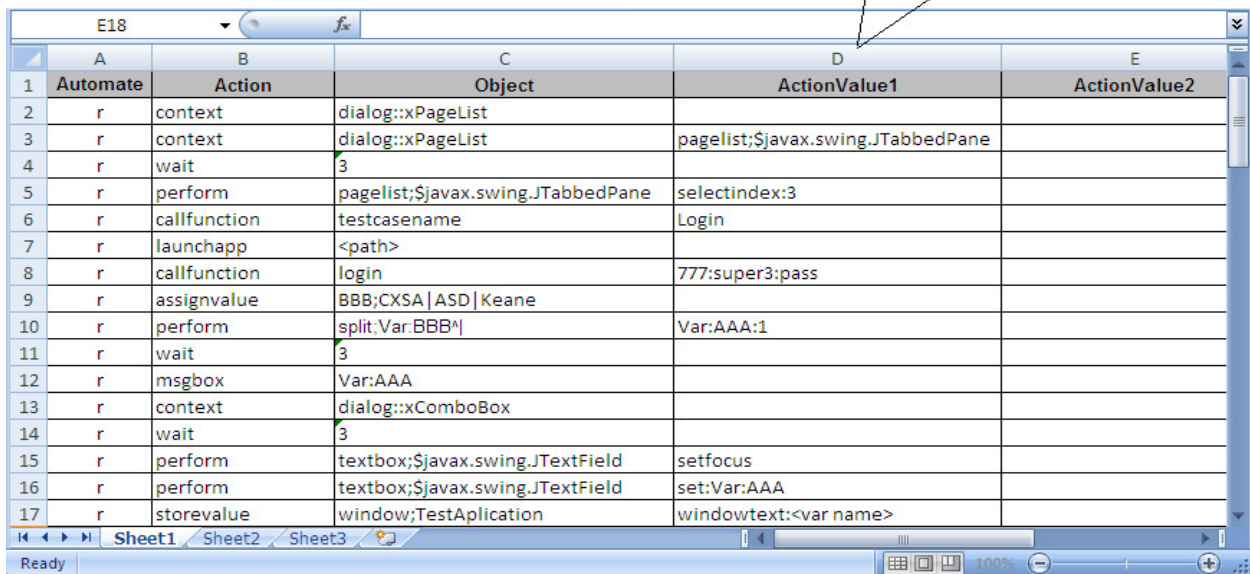
ACTION

Example 2: The keyword syntax to check the "Enabled" operation on the button is as follows:

Action	Object	ActionValue1
check	button;<button tag name>	enabled:true

CHECKING

Column to indicate specific action to be performed.



	A	B	C	D	E
	Automate	Action	Object	ActionValue1	ActionValue2
1	r	context	dialog::xPageList		
2	r	context	dialog::xPageList	pagelist;\$javax.swing.JTabbedPane	
3	r	wait	3		
4	r	perform	pagelist;\$javax.swing.JTabbedPane	selectindex:3	
5	r	callfunction	testcasename	Login	
6	r	launchapp	<path>		
7	r	callfunction	login	777:super3:pass	
8	r	assignvalue	BBB;CXSA ASD Kean		
9	r	perform	split;Var.BBB^	Var:AAA:1	
10	r	wait	3		
11	r	msgbox	Var:AAA		
12	r	context	dialog::xComboBox		
13	r	wait	3		
14	r	perform	textbox;\$javax.swing.JTextField	setfocus	
15	r	perform	textbox;\$javax.swing.JTextField	set:Var:AAA	
16	r	storevalue	window;TestAplication	windowtext:<var name>	

Figure 6 Column 'Actionvalue1'

The most commonly performed actions with the 'Perform' keyword are:

1. Click

'Click' is used to perform the click operation on objects. It is used with the perform keyword in keyword scripts (.g., clicking a Javabutton).

2. Close

'Close' is used to perform the close operation on any open objects. It is used with the perform keyword in keyword script (e.g., closing a window, dialog, etc.).

3. Maximize

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

4. Minimize

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

5. Restore

'Restore' is used to perform the restore operation on any open window. 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, and 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 by providing the index of respective items. It is used with the perform keyword in keyword scripts.

8. Set:<Text>

'Set' is used to assign a value to a text field. It is used with the perform keyword in keyword scripts.

9. Set:Var:<Var_Name>

'Set' is used to assign a value in variable to the text field. It is used with the perform keyword in keyword scripts.

10.Set:d_currenttime

'Set' is used to assign the current system time to the text field. It is used with the perform keyword in keyword scripts.

11.Set:d_currentdate

'Set' is used to assign the current system date to the text field. It is used with the perform keyword in keyword scripts.

12.Set:d_d:<value to add/subtract>

This adds or subtracts the value specified to the current system date and sets it to the text field. It is used with the perform keyword in keyword scripts.

13.Set:d_m:<value to add/subtract>

This adds or subtracts the value specified to the current system month and sets it to the text field. It is used with the perform keyword in keyword scripts.

14.Set:d_y:<value to add/subtract>

This adds or subtracts the value specified to the current system year and sets it to the text field. It is used with the perform keyword in keyword scripts.

15.Type:<Value>

'Type' is used to assign a value to a text field. It is used with the perform keyword.

16.Type:Var:<Var_Name>

'Type' is used to assign a value in variable to the text field. It is used with the perform keyword in keyword scripts.

17.Type_dt:<row number>:<col name>

'Type_dt' is used to assign a cell value present in the table to the text field. It is used with the perform keyword in keyword scripts.

18.Select_dt:<row number>:<col name>

'Select_dt' is used to select an item in combobox specified in a cell of the table. It is used with the perform keyword in keyword scripts.

19.Doubleclick

This is used to perform the double-click operation on objects. It is used with the perform keyword in keyword scripts.

20.Press:<button tag name>

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

21.Expand:<root node name>/ (Path of node)

This is used to expand the tree item specified. It is used with the perform keyword in keyword scripts.

22.Collapse:<root node name>/ (Path of node)

This is used to collapse the tree item specified. It is used with the perform keyword in keyword scripts.

23.Collapseindex:<index>

This is used to collapse the tree item specified in index. It is used with the perform keyword in keyword scripts.

24.Releaseitemindex:<index>

This is used to move the mouse pointer to the item specified. It is used with the perform keyword in keyword scripts.

25.Setfocus

This is used to set the focus on the object specified. It is used with the perform keyword in keyword scripts.

26.Pick:<item>

This is used to select an item from the menu, submenu, and popupmenu. It is used with the perform keyword in keyword scripts.

27.Extendselect:<item>

This is used to extend select an item in the list box. It is used with the perform keyword in keyword scripts.

28.Multiselect:<item>

This is used to multi select an item in the list box. It is used with the perform keyword in keyword scripts.

29.Doubleselect:<item>

This is used to double select an item in the list box. It is used with the perform keyword in keyword scripts.

30.Toggle

This is used to toggle the state of a toggle button. It is used with the perform keyword in keyword scripts.

31.Setstate:<true/false>

This is used to change the state of a toggle button. It is used with the perform keyword in keyword scripts.

32.Scrollmax

This is used to scroll the scrollbar to maximum position. It is used with the perform keyword in keyword scripts.

33.Scrollmin

This is used to scroll the scrollbar to minimum position. It is used with the perform keyword in keyword scripts.

34.Setposition:<position>

This is used to set the scrollbar to a specified position. It is used with the perform keyword in keyword scripts.

35.Check

This is used to check the specified checkbox. It is used with the perform keyword in keyword scripts.

36.Uncheck

This is used to uncheck the specified checkbox. It is used with the perform keyword in keyword scripts.

37.Select:<Row no>:<Col no>

This is used to select a particular cell in a table. It is used with the perform keyword in keyword scripts.

38.Select:Var:<Var Name>:<Col no>

This is used to select a particular cell in a table, having row number in a variable. It is used with the perform keyword in keyword scripts.

39.Column1:Column2...

This is used to specify the column that should be used for the execution of multiple capture of the Sql query. It is used with the perform keyword in keyword scripts.

40.Clear

This is used to clear the text in the textbox. It is used with the perform keyword in keyword scripts.

The most commonly used keywords for specific actions that can be used with the generic keyword **Check** written in Column 'Action' are:

1. Enabled:<True/False>

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

2. Exist:<True/False>

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

3. Focused:<True/False>

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

4. Text:<text / Var:Variable_Name>

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

5. ItemExist:<Item name>

This is a check operation that is used to verify whether an item is present or not in the given object.

6. Verifytext:<text / Var:Variable_Name>:<Row no>:<Col no>

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

7. ItemsCount:<Item / Var:Variable_Name>

This is a check operation that is used to verify the number of items present or not in a given object.

8. Select:<item name>

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

9. Selectindex:<index>

This is a check operation that is used to verify whether the desired item is selected or not from the Combobox, Radiobutton, and Listbox.

10. Blank:<True/False>

This is a check operation that is used to verify whether the textbox is blank or not.

11. Multiselect:<True/False>

This is a check operation that is used to verify whether the Table or Listbox has the option to multiselect.

12. Rowcount:<no of rows>

This is a check operation that is used to verify whether the Table has the specified number of rows.

13. Colcount:<no of cols>

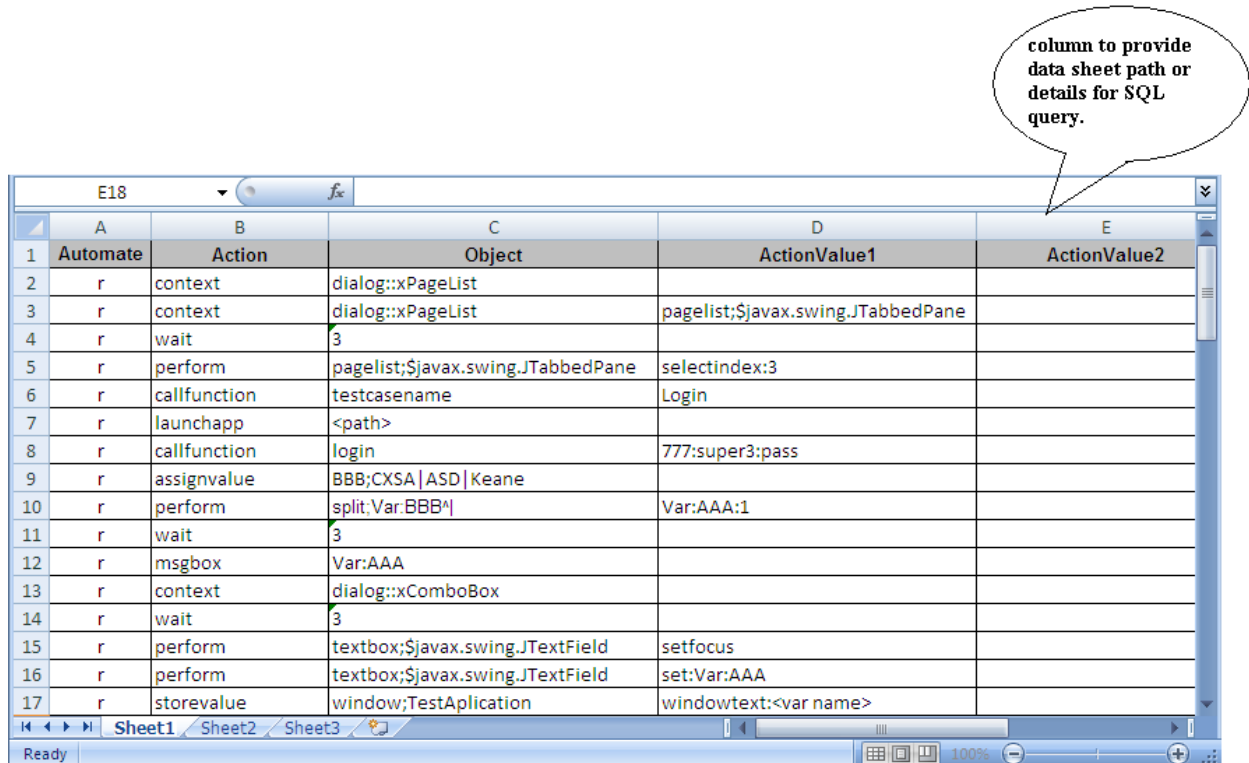
This is a check operation that is used to verify whether the Table has the specified number of columns.

14. Expand:<True/False>

This is a check operation that is used to verify whether the Tree item is expandable or not.

2.2.5. ActionValue2 (Column 'E')

The fifth column of the keyword script sheet may be used to specify the data sheet path (for keywords like type_dt, select_dt) or to provide details for execution of the SQL query.

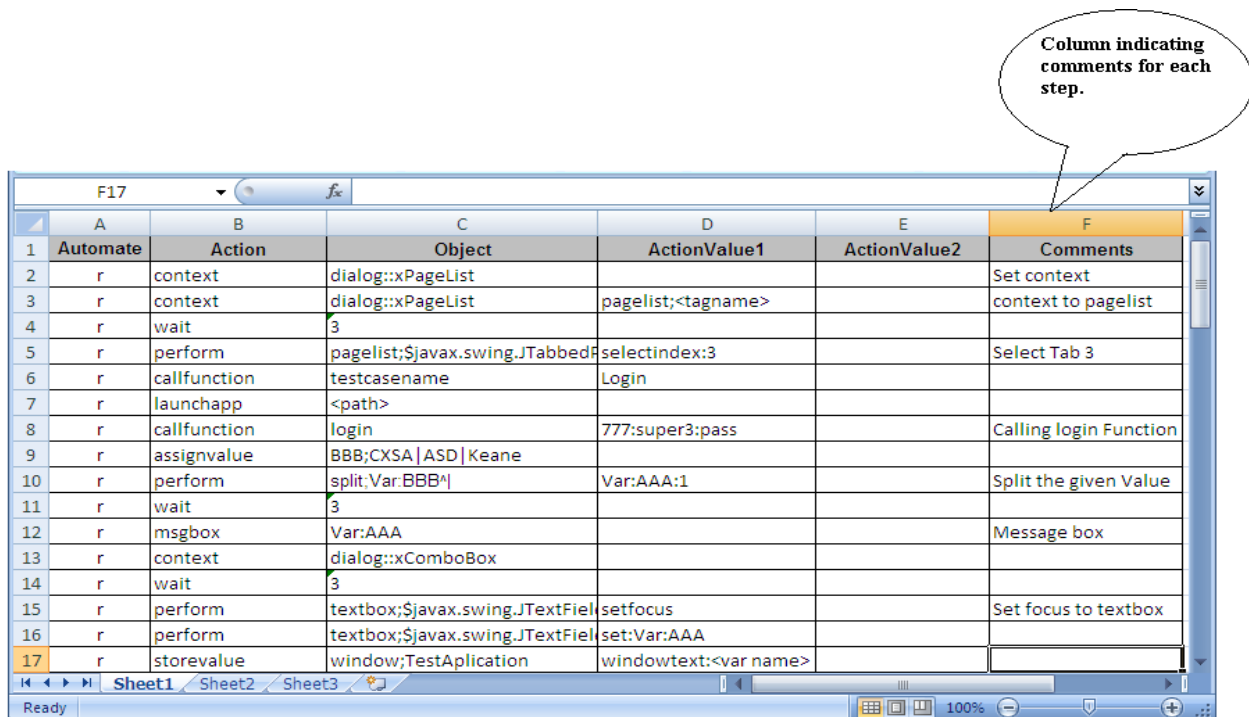


	A	B	C	D	E
1	Automate	Action	Object	ActionValue1	ActionValue2
2	r	context	dialog::xPageList		
3	r	context	dialog::xPageList	pagelist;\$javax.swing.JTabbedPane	
4	r	wait	3		
5	r	perform	pagelist;\$javax.swing.JTabbedPane	selectindex:3	
6	r	callfunction	testcasename	Login	
7	r	launchapp	<path>		
8	r	callfunction	login	777:super3:pass	
9	r	assignvalue	BBB;CXSA ASD Keane		
10	r	perform	split;Var.BBB^	Var:AAA:1	
11	r	wait	3		
12	r	msgbox	Var:AAA		
13	r	context	dialog::xComboBox		
14	r	wait	3		
15	r	perform	textbox;\$javax.swing.JTextField	setfocus	
16	r	perform	textbox;\$javax.swing.JTextField	set:Var:AAA	
17	r	storevalue	window;TestApplication	windowtext:<var name>	

Figure 7 Column 'Actionvalue2'

2.2.6 Comments (Column 'F')

The 'Comments' column is used to enter generic information about the current step being run. It provides a better understanding of the steps being performed in the particular test script and also helps to map the test script to the manual test case.



	A	B	C	D	E	F
1	Automate	Action	Object	ActionValue1	ActionValue2	Comments
2	r	context	dialog::xPageList			Set context
3	r	context	dialog::xPageList	pagelist;<tagname>		context to pagelist
4	r	wait	3			
5	r	perform	pagelist;\$javax.swing.JTabbedP	selectindex:3		Select Tab 3
6	r	callfunction	testcasename	Login		
7	r	launchapp	<path>			
8	r	callfunction	login	777:super3:pass		Calling login Function
9	r	assignvalue	BBB;CXSA ASD Keane			
10	r	perform	split:Var:BBB^	Var:AAA:1		Split the given Value
11	r	wait	3			
12	r	msgbox	Var:AAA			Message box
13	r	context	dialog::xComboBox			
14	r	wait	3			
15	r	perform	textbox;\$javax.swing.JTextField	setfocus		Set focus to textbox
16	r	perform	textbox;\$javax.swing.JTextField	set:Var:AAA		
17	r	storevalue	window;TestApplication	windowtext:<var name>		

Figure 8 Column 'Comments'

2.2.7 Delimiters

Delimiters are any string characters used to identify the substring limits. Delimiters are generally used with the GetField function, which is used to split the input into different substrings.

Purpose of using delimiters:

The main purpose of using delimiters in this framework is to break down the input values to different strings and take them as keywords to perform any operation on the object.

Delimiters used in this framework:

The most important point to keep in mind while scripting using the keyword-driven approach is to place separators or delimiters between two keywords.

Delimiters that are used in the framework are:

- : (Colon)
- ; (Semi colon)
- ::(Double colon)
- ^ (Caret)

Understanding the usage of delimiters:

There are four columns involved in the keyword-driven approach. The role of delimiters comes in the 'Objects' column (column 3) and 'Action Value1' column (column 4).

- **'Objects' column (column c):**

This column is used to define the class and the name of the object. The delimiter used in this column separates the class of the object and the name of the object with a semi-colon ';'.

Example:

```
Checkbox;<checkbox tagname>
```

- **'ActionValue1' column (column D):**

This column usually provides the details of the operations that need to be performed on the object. The delimiter used to separate the property and the property values in this column is a colon ':'.

Example:

```
Selectindex:< index >
```

Example: To specify the child objects present in a window, browser, or dialog box, the delimiter that is used is a double colon '::'.

```
pagelist;<pagelist tag name>::Customwin;<customwin tagname>
```

- **'ActionValue2' column (column E):**

This column is usually used to specify details like drivename, servername, databasename, databaseuserid, or databasepassword, which is required to execute SQL queries. The delimiter used is a colon ':'.

Example:

```
<Drivename>:<servername>:<databasename>:<databaseuserid>  
:<databasePassword>
```



If the delimiters are present in the Object Name/Text, then they cannot be used directly. They should be passed through variables.

2.2.8. Variables

To store a value in a variable, use assignvalue keyword.

Example:

r	assignvalue	Strname;John
---	-------------	--------------

Here in the variable 'Strname', the value 'John' is stored.

- To store the property value of an object, use Storevalue keyword.
Example:

Storevalue	combobox;<Combobox tagname>	itemcount:<Variable Name>
------------	-----------------------------	---------------------------

Here, the itemcount of the combobox is stored to a variable.

- To input a value to a field from a variable, the variable should be preceded by 'Var:'

Example:

Perform	textbox;<textfield tag name>	set:Var:<Var_Name>
---------	------------------------------	--------------------

Here, the value stored in Variable is set into the textbox.

3. Sequence of Keywords

While scripting using keywords, some keywords have to be written in combination with other keywords. This section deals with the methodologies.

3.1. Use of Keyword 'Context'

The keyword 'context' has to be used whenever the AUT screen changes.

Example:

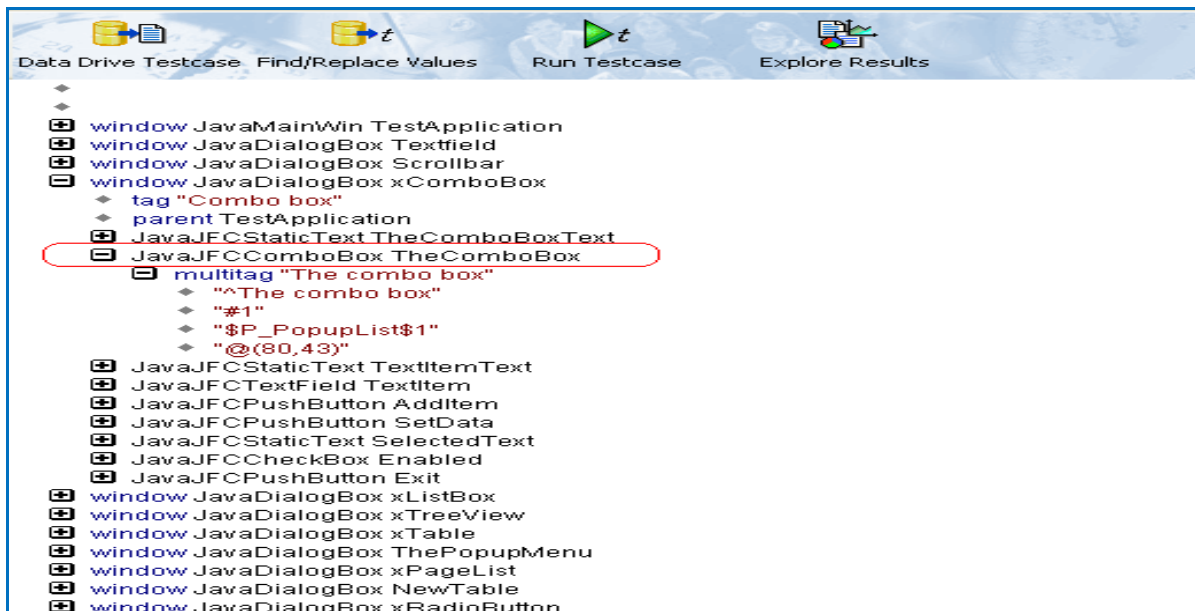


Figure 9 Keyword 'Context'

If the object "TheComboBox" has to be used in the script, then the preceding row should have the context set to the respective dialog in the hierarchy.

Therefore, the combination to be used while performing an action on the object "TheComboBox" is:

context	dialog::xComboBox	
perform	Combobox;\$P_PopupList\$1	Select:Apple

If we have to use another object on the same page, then the context need not be set again.

context	dialog::xComboBox	
perform	Combobox;\$P_PopupList\$1	Select:Apple
perform	textbox;\$javax.swing.JTextField	set:Apple

3.2. Use of 'Conditional Statements'

If the user is implementing an If - Else conditional statement, then the keyword is followed by a semi-colon ';' and the values that indicate the start row and the end row should be separated by a semicolon ';'.

Example:

condition	<expression1>;Conditional operation;<expression2>	<true start row>;<>false start row>
-----------	---	-------------------------------------

If the condition mentioned is 'True', execution starts from the true start row and would end at the false start row specified. If the

condition specified is 'False', execution starts from false start row and the execution would continue as normal.

Two conditional statements have to be used together to satisfy the 'AND' logical operation.

Example:

condition	<expression1>;Conditional operation;<expression2>	<true start row>;<false start row>
condition	<expression2>;Conditional operation;<expression3>	<true start row>;<false start row>

Therefore, this effectively implies that an 'AND' operation is being performed.

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.