



Open2Test Test Automation Framework Keywords - WEB framework for e-Tester

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.

1 .Keyword for Launching the Application

1 .Launch Web Application

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
LaunchApp	<URL Name>		

Description: Launch the specific URL in e-Tester Browser Pane

Example: Launch the sample web application
LaunchApp|www.google.com

2 .Keyword for Performing an Action

1 .Set

Object Type: HTML TextBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	TextBox<>HTML::Value;Tag; Property	set;<Value>	

Description: Set the value of the required field identified by the HTML parameters of the field to the required value.

Example: Set the Username field with the value 'User1'
Perform|TextBox<>HTML::UserName;input;ID|set;User1

2 .Set

Object Type: HTML TextBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	TextBox<>DOM::ElementPath	set;<Value>	

Description: Set the value of the required field identified by the DOM path of the field to the required value.

Example: Set the Username field with the value 'User1'
Perform |TextBox<>DOM::<DOMPath> |set;User1

3 .Set value from Variable

Object Type: HTML TextBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	TextBox<>HTML::Value;Tag;Property	frmvar;<VarName>	

Description: Sets the value of the required Text Field (identified by H from the variable captured in the Test Run.

Example: Sets the text box with the specified variable value
Perform |TextBox<>DOM::<DOMPath> |frmvar;Var1

4 .Set value from Variable

Object Type: HTML TextBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	TextBox<>DOM::ElementPath	frmvar;<VarName>	

Description: Sets the value of the required Text Field (identified by H from the variable captured in the Test Run.

Example: Sets the text box with the specified variable value
Perform |TextBox<>DOM::<DOMPath> |frmvar;Var1

5 .Click

Object Type: HTML Button

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	Button<>HTML::Value;Tag;Property	click	

Description: Click on the specified button (identified by HTML Parameters)
 Example: Click on the Login Button
 Perform | Button<>HTML::LgnBtn;input;name | click

6 .Click

Object Type: HTML Button
 Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	Button<>DOM::ElementPath	click	

Description: Click on the specified button (identified by DOM Path)
 Example: Click on the Login Button
 Perform | Button<>DOM::<DOMPath> | click

7 .Click

Object Type: HTML Tab
 Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	Tab<>HTML::Value;Tag;Property	click	

Description: Click on the specified tab (identified by HTML Parameters)
 Example: Click on the 'Subscriber' tab
 Perform | Tab<>HTML::SubTab;input;name | click

8 .Click

Object Type: HTML Tab
 Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	Tab<>DOM::ElementPath	click	

Description: Click on the specified tab (identified by DOM Path)
 Example: Click on the 'Subscriber' tab
 Perform |Tab<>DOM::<DOMPath>|click

9 .Click

Object Type: HTML Link
 Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	Link<>HTML::Value;Tag;Property	click	

Description: Click on the specified link (identified by HTML Parameters)
 Example: Click on the 'Logout' link
 Perform |Link<>HTML::LgOut;input;name|click

10 .Click

Object Type: HTML Link
 Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	Link<>DOM::ElementPath	click	

Description: Click on the specified Link (identified by DOM Path)
 Example: Click on the 'Logout' link
 Perform |Link<>DOM::<DOMPath>|click

11 .Check

Object Type: HTML CheckBox
 Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	CheckBox<>HTML::Value;Tag;Property	check	

Description: Checks the required checkbox identified by the HTML Parameter

Example: Checks the 'Remember' checkbox
Perform | CheckBox<>HTML::rememcbox;input;id | check

12 .Uncheck

Object Type: HTML CheckBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	CheckBox<>HTML::Value;Tag;Property	uncheck	

Description: Unchecks the required checkbox identified by the HTML Parameter

Example: Unchecks the 'Remember' checkbox
Perform | CheckBox<>HTML::rememcbox;input;id | uncheck

13 .Check

Object Type: HTML CheckBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	CheckBox<>DOM::ElementPath	check	

Description: Checks the required checkbox identified by the DOM Path

Example: Checks the 'Remember' checkbox
Perform | CheckBox<>DOM::<DOMPath> | check

14 .Uncheck

Object Type: HTML CheckBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	CheckBox<>DOM::ElementPath	uncheck	

Description: Unchecks the required checkbox identified by DOM Path
 Example: Unchecks the 'Remember' checkbox
 Perform | CheckBox<>DOM::<<DOMPath> | uncheck

15 .Click

Object Type: HTML RadioButton

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	RadioButton<>HTML::Value;Tag;Property	click	

Description: Selects the required Radio Button, identified by the HTML Parameters
 Example: Click on the 'State' radio button
 Perform | RadioButton<>HTML::state;input;name | click

16 .Click

Object Type: HTML RadioButton

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	RadioButton<>DOM::ElementPath	click	

Description: Selects the required Radio Button, identified by the DOM Path
 Example: Click on the 'State' radio button
 Perform | RadioButton<>DOM::<<DOMPath> | click

17 .Set

Object Type: HTML Dropdown

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	DropDown<>HTML::Value;Tag;Property	set;<Value>	

Description: Chooses the specified value from the required dropdown (identified by HTML Parameters)
 Example: Sets the dropdown with the specified value
 Perform | DropDown<>HTML::lang;select;name | set;ar

18 .Set

Object Type: HTML Dropdown
 Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	DropDown<>DOM::ElementPath	set;<Value>	

Description: Chooses the specified value from the required dropdown (identified by DOM Path)
 Example: Sets the dropdown with the specified value
 Perform | DropDown<>DOM::<DOMPath> | set;ar

19 .Set

Object Type: HTML ListBox
 Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Perform	ListBox<>HTML::Value;Tag;Property	setlist;<n1>,<n2>,<n3>	

Selects the specified values from the required Listbox (identified by HTML Parameters).

Description: Where n1,n2,n3 corresponds to the index no of the Listbox values.
 Example: Chooses the specified values in the Listbox
 Perform | ListBox<>HTML::lang;multiselect;name | setlist;en,ta,hi

20 .Set

Object Type: HTML ListBox
 Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
--------	-------------------	---------------	---------------

Perform	ListBox<>DOM::ElementPath	setlist;<n1>,<n2>,<n3>	
---------	---------------------------	------------------------	--

Selects the specified values from the required Listbox (identified by DOM Path).

Description: Where n1,n2,n3 corresponds to the index no of the Listbox values.

Example: Chooses the specified values in the Listbox
Perform | ListBox<>DOM::<DOMPath> | setlist;en,ta,hi

3 .Keyword for Checking

1 .Enabled

Object Type: HTML TextBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	TextBox<>HTML::Value;Tag;Property	enabled;<VarName>	

Verifies whether the required field (identified by HTML Parameter) is enabled or not. And stores the value 'True' if it is enabled else stores the value 'False' in the specified variable.

Description: Check whether the Username text box is enabled
Example: Check | TextBox<>HTML::UserName;input;ID | enabled;Var1

2 .Disabled

Object Type: HTML TextBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	TextBox<>HTML::Value;Tag;Property	disabled;<VarName>	

Verifies whether the required field (identified by HTML Parameter) is disabled or not. And stores the value 'True' if it is disabled else stores the value 'False' in the specified variable.

Description: Check whether the Username text box is disabled
Example: Check | TextBox<>HTML::UserName;input;ID | disabled;Var1

3 .Exists

Object Type: HTML TextBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	TextBox<>HTML::Value;Tag;Property	exists;<VarName>	

Description: Verifies whether the required field (identified by HTML Parameters) exists or not. And stores the value 'True' if it exists else stores the value 'False' in the specified variable.

Example: Check whether the Username text box is exists
Check|TextBox<>HTML::UserName;input;ID|exists;Var1

4 .Enabled

Object Type: HTML TextBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	TextBox<>DOM::ElementPath	enabled;<VarName>	

Description: Verifies whether the required field (identified by DOM Path) is enabled or not. And stores the value 'True' if it is enabled else stores the value 'False' in the specified variable.

Example: Check whether the Username text box is enabled
Check|TextBox<>DOM::<DOMPath>;ID|enabled;Var1

5 .Disabled

Object Type: HTML TextBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	TextBox<>DOM::ElementPath	disabled;<VarName>	

Description: Verifies whether the required field (identified by DOM Path) is disabled or not. And stores the value

'True' if it is disabled else stores the value 'False' in the specified variable.

Example:

Check whether the Username text box is disabled
Check|TextBox<>DOM::<DOMPath>|disabled;Var1

6 .Exists

Object Type: HTML TextBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	TextBox<>DOM::ElementPath	exists;<VarName>	

Description:

Verifies whether the required field (identified by DOM Path) exists or not. And stores the value 'True' if it exists else stores the value 'False' in the specified variable.

Example:

Check whether the Username text box is exists
Check|TextBox<>DOM::<DOMPath>|exists;Var1

7 .Enabled

Object Type: HTML Button

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Button<>HTML::Value;Tag;Property	enabled;<VarName>	

Description:

Verifies whether the required 'Button' (identified by HTML Parameters) is enabled or not. And stores the value 'True' in the specified variable if it is enabled else stores 'False'.

Example:

Checks whether the specified button is enabled.
Check|Button<>HTML::LgnBtn;input;name|enabled;Var1

8 .Disabled

Object Type: HTML Button

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
--------	-------------------	---------------	---------------

Check	Button<>HTML::Value;Tag;Property	disabled;<VarName>	
-------	----------------------------------	--------------------	--

Description: Verifies whether the required 'Button' (identified by HTML Parameters) is disabled or not. And stores the value 'True' in the specified variable if it is disabled else stores 'False'.

Example: Checks whether the specified button is disabled.
Check | Button<>HTML::LgnBtn;input;name | disabled;Var1

9 .Exists

Object Type: HTML Button

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Button<>HTML::Value;Tag;Property	exists;<VarName>	

Description: Verifies whether the mentioned button (identified by HTML parameters) exists. And stores the value 'True' in the specified variable if the button exists else stores the value 'False'

Example: Checks whether the specified button exists.
Check | Button<>HTML::LgnBtn;input;name | exists;Var1

10 .Enabled

Object Type: HTML Button

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Button<>DOM::ElementPath	enabled;<VarName>	

Description: Verifies whether the required 'Button' (identified by DOM Path) is enabled or not. And stores the value 'True' in the specified variable if it is enabled else stores 'False'.

Example: Checks whether the specified button is enabled.
Check | Button<>DOM::<DOMPath> | enabled;Var1

11 .Disabled

Object Type: HTML Button

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Button<>DOM::ElementPath	disabled;<VarName>	

Description: Verifies whether the required 'Button' (identified by DOM Path) is disabled or not. And stores the value 'True' in the specified variable if it is disabled else stores 'False'.

Example: Checks whether the specified button is disabled.
Check | Button<>DOM::<DOMPath> | disabled;Var1

12 .Exists

Object Type: HTML Button

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Button<>DOM::ElementPath	exists;<VarName>	

Description: Verifies whether the mentioned button (identified by DOM Path) exists. And stores the value 'True' in the specified variable if the button exists else stores the value 'False'

Example: Checks whether the specified button exists.
Check | Button<>DOM::<DOMPath> | exists;Var1

13 .Enabled

Object Type: HTML Tab

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Tab<>HTML::Value;Tag;Property	enabled;<VarName>	

Description: Verifies whether the required Tab (identified by HTML Parameters) is enabled or not. And stores the value 'True' in the specified variable if it is enabled else stores 'False'.

Example: Checks whether the specified tab is enabled.

Check|Tab<>HTML::SubTab;input;name|enabled;Var1

14 .Disabled

Object Type: HTML Tab

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Tab<>HTML::Value;Tag;Property	disabled;<VarName>	

Description: Verifies whether the required Tab (identified by HTML Parameters) is disabled or not. And stores the value 'True' in the specified variable if it is disabled else stores 'False'.

Example: Checks whether the specified tab is disabled.
Check|Tab<>HTML::SubTab;input;name|disabled;Var1

15 .Exists

Object Type: HTML Tab

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Tab<>HTML::Value;Tag;Property	exists;<VarName>	

Description: Verifies whether the mentioned tab (identified by HTML parameters) exists. And stores the value 'True' in the specified variable if the button exists else stores the value 'False'

Example: Checks whether the specified tab exists.
Check|Tab<>HTML::SubTab;input;name|exists;Var1

16 .Enabled

Object Type: HTML Tab

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Tab<>DOM::ElementPath	enabled;<VarName>	

Description: Verifies whether the required tab (identified by DOM Path) is enabled or not. And stores the value 'True' in the specified variable if it is enabled else stores 'False'.

Example: Checks whether the specified tab is enabled.
Check|Tab<>DOM::<<DOMPath>|enabled;Var1

17 .Disabled

Object Type: HTML Tab

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Tab<>DOM:: <elementpath< td=""> <td>disabled;<VarName></td> <td></td> </elementpath<>	disabled;<VarName>	

Description: Verifies whether the required Tab (identified by DOM Path) is disabled or not. And stores the value 'True' in the specified variable if it is disabled else stores 'False'.

Example: Checks whether the specified tab is disabled.
Check|Tab<>DOM::<<DOMPath>|disabled;Var1

18 .Exists

Object Type: HTML Tab

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Tab<>DOM:: <elementpath< td=""> <td>exists;<VarName></td> <td></td> </elementpath<>	exists;<VarName>	

Description: Verifies whether the mentioned Tab (identified by DOM Path) exists. And stores the value 'True' in the specified variable if the button exists else stores the value 'False'.

Example: Checks whether the specified tab exists.
Check|Tab<>DOM::<<DOMPath>|exists;Var1

19 .Enabled

Object Type: HTML Link

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Link<>HTML::Value;Tag;Property	enabled;<VarName>	

Description: Verifies whether the required Link (identified by HTML Parameters) is enabled or not. And stores the value 'True' in the specified variable if it is enabled else stores 'False'.

Example: Checks whether the specified link is enabled.
Check|Link<>HTML::LgOut;input;name|enabled;Var1

20 .Disabled

Object Type: HTML Link

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Link<>HTML::Value;Tag;Property	disabled;<VarName>	

Description: Verifies whether the required Link (identified by HTML Parameters) is disabled or not. And stores the value 'True' in the specified variable if it is disabled else stores 'False'.

Example: Checks whether the specified link is disabled.
Check|Link<>HTML::LgOut;input;name|disabled;Var1

21 .Exists

Object Type: HTML Link

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Link<>HTML::Value;Tag;Property	exists;<VarName>	

Description: Verifies whether the mentioned Link (identified by HTML parameters) exists. And stores the value 'True' in the specified variable if the button exists else stores the value 'False'

Example: Checks whether the specified link exists.
Check|Link<>HTML::LgOut;input;name|exists;Var1

22 .Enabled

Object Type: HTML Link

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Link<>DOM::ElementPath	enabled;<VarName>	

Description: Verifies whether the required Link (identified by DOM Path) is enabled or not. And stores the value 'True' in the specified variable if it is enabled else stores 'False'.

Example: Checks whether the specified link is enabled.
Check|Link<>DOM::<DOMPath>|enabled;Var1

23 .Disabled

Object Type: HTML Link

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Link<>DOM::ElementPath	disabled;<VarName>	

Description: Verifies whether the required Link (identified by DOM Path) is disabled or not. And stores the value 'True' in the specified variable if it is disabled else stores 'False'.

Example: Checks whether the specified link is disabled.
Check|Link<>DOM::<DOMPath>|disabled;Var1

24 .Exists

Object Type: HTML Link

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Link<>DOM::ElementPath	exists;<VarName>	

Description: Verifies whether the mentioned Link (identified by DOM Path) exists. And stores the value 'True' in the specified variable if the button exists else stores the value 'False'

Example: Checks whether the specified link exists.
Check|Link<>DOM::<DOMPath>|exists;Var1

25 .Enabled

Object Type: HTML CheckBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	CheckBox<>HTML::Value;Tag;Property	enabled;<VarName>	

Description: Verifies whether the required checkbox (Identified by HTML Parameter) has been enabled. And stores the value 'True' in the specified variable if it is enabled. Else stores the value 'False'

Example: Checks whether the specified checkbox is enabled.
Check|CheckBox<>HTML::rememcbox;input;id|enabled;Var1

26 .Disabled

Object Type: HTML CheckBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	CheckBox<>HTML::Value;Tag;Property	disabled;<VarName>	

Verifies whether the required checkbox (Identified by HTML Parameter) has been disabled. And stores the value 'True' in the specified variable if it is disabled.

Description: Else stores the value 'False'

Example: Checks whether the specified checkbox is disabled.
Check|CheckBox<>HTML::rememcbox;input;id|disabled;Var1

27 .Exists

Object Type: HTML CheckBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	CheckBox<>HTML::Value;Tag;Property	exists;<VarName>	

Description: Verifies whether the required Checkbox (Identified by HTML Parameter) exists or not. And stores the value 'True' in the specified variable if it exists, else stores the value 'False'

Example: Checks whether the specified checkbox exists.
Check|CheckBox<>HTML::rememcbox;input;id|exists;Var1

28 .Checked

Object Type: HTML CheckBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	CheckBox<>HTML::Value;Tag;Property	checked;<VarName>	

Description: Verifies whether the required checkbox (identified by HTML Parameter) is checked. And stores the value 'True' in the specified variable if it has been checked else stores the value 'False'.

Example: Checks whether the specified checkbox is checked.
Check|CheckBox<>HTML::rememcbox;input;id|checked;Var1

29 .Unchecked

Object Type: HTML CheckBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	CheckBox<>HTML::Value;Tag;Property	unchecked;<VarName>	

Description: Verifies whether the required checkbox (identified by HTML Parameter) is unchecked. And stores the value 'True' in the specified variable if it has been unchecked else stores the value 'False'.

Example: Checks whether the specified checkbox is unchecked.
Check|CheckBox<>HTML::rememcbox;input;id|unchecked;Var1

30 .Enabled

Object Type: HTML CheckBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	CheckBox<>DOM::ElementPath	enabled;<VarName>	

Verifies whether the required checkbox (Identified by DOM Path) has been enabled. And stores the value 'True' in the specified variable if it is enabled.

Description: Else stores the value 'False'

Example: Checks whether the specified checkbox is enabled.
Check|CheckBox<>DOM::<DOMPath>|enabled;Var1

31 .Disabled

Object Type: HTML CheckBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	CheckBox<>DOM::ElementPath	disabled;<VarName>	

Verifies whether the required checkbox (Identified by DOM Path) has been disabled. And stores the value 'True' in the specified variable if it is disabled.

Description: Else stores the value 'False'

Example: Checks whether the specified checkbox is disabled.
Check|CheckBox<>DOM::<DOMPath>|disabled;Var1

32 .Exists

Object Type: HTML CheckBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
--------	-------------------	---------------	---------------

Check	CheckBox<>DOM::ElementPath	exists;<VarName>	
-------	----------------------------	------------------	--

Description: Verifies whether the required Checkbox (Identified by DOM Path) exists or not. And stores the value 'True' in the specified variable if it exists, else stores the value 'False'

Example: Checks whether the specified checkbox exists.
Check|CheckBox<>DOM::<DOMPath>|exists;Var1

33 .Checked

Object Type: HTML CheckBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	CheckBox<>DOM::ElementPath	checked;<VarName>	

Description: Verifies whether the required checkbox (identified by DOM Path) is checked. And stores the value 'True' in the specified variable if it has been checked else stores the value 'False'.

Example: Checks whether the specified checkbox is checked.
Check|CheckBox<>DOM::<DOMPath>|checked;Var1

34 .Unchecked

Object Type: HTML CheckBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	CheckBox<>DOM::ElementPath	unchecked;<VarName>	

Description: Verifies whether the required checkbox (identified by DOM Path) is unchecked. And stores the value 'True' in the specified variable if it has been unchecked else stores the value 'False'.

Example: Checks whether the specified checkbox is unchecked.
Check|CheckBox<>DOM::<DOMPath>|unchecked;Var1

35 .Enabled

Object Type: HTML RadioButton

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	RadioButton<>HTML::Value;Tag;Property	enabled;<VarName>	

Verifies whether the required radio button (Identified by HTML Parameter) has been enabled. And stores the value 'True' in the specified variable if it is enabled.

Description: Else stores the value 'False'

Example: Checks whether the specified radio button is enabled.

Check|RadioButton<>HTML::state;input;name|enabled;Var1

36 .Disabled

Object Type: HTML RadioButton

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	RadioButton<>HTML::Value;Tag;Property	disabled;<VarName>	

Verifies whether the required radio button (Identified by HTML Parameter) has been disabled. And stores the value 'True' in the specified variable if it is disabled.

Description: Else stores the value 'False'

Example: Checks whether the specified checkbox is disabled.

Check|RadioButton<>HTML::state;input;name|disabled;Var1

37 .Exists

Object Type: HTML RadioButton

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
--------	-------------------	---------------	---------------

Check	RadioButton<>HTML::Value;Tag;Property	exists;<VarName>	
-------	---------------------------------------	------------------	--

Description: Verifies whether the required radio button (Identified by HTML Parameter) exists or not. And stores the value 'True' in the specified variable if it exists, else stores the value 'False'

Example: Checks whether the specified checkbox exists.
Check | RadioButton<>HTML::state;input;name | exists;Var1

38 .Checked

Object Type: HTML RadioButton

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	RadioButton<>HTML::Value;Tag;Property	checked;<VarName>	

Description: Verifies whether the required radio button (identified by HTML Parameter) has been selected. And stores the value 'True' in the specified variable if it has been selected else stores the value 'False'.

Example: Checks whether the specified radio button is checked.
Check | RadioButton<>HTML::state;input;name | checked;Var1

39 .Unchecked

Object Type: HTML RadioButton

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	RadioButton<>HTML::Value;Tag;Property	unchecked;<VarName>	

Description: Verifies whether the required radio button (identified by HTML Parameter) has been deselected. And stores the value 'True' in the specified variable if it has been deselected else stores the value 'False'.

Example: Checks whether the specified radio button is checked.
Check | RadioButton<>HTML::state;input;name | unchecked;Var1

40 .Enabled

Object Type: HTML RadioButton

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	RadioButton<>DOM::ElementPath	enabled;<VarName>	

Verifies whether the required radio button (Identified by DOM Path) has been enabled. And stores the value 'True' in the specified variable if it is enabled.

Description: Else stores the value 'False'

Example: Checks whether the specified radio button is enabled.
Check|RadioButton<>DOM::<DOMPath>|enabled;Var1

41 .Disabled

Object Type: HTML RadioButton

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	RadioButton<>DOM::ElementPath	disabled;<VarName>	

Verifies whether the required radio button (Identified by DOM Path) has been disabled. And stores the value 'True' in the specified variable if it is disabled. Else stores the value 'False'

Description:

Example: Checks whether the specified checkbox is disabled.
Check|RadioButton<>DOM::<DOMPath>|disabled;Var1

42 .Exists

Object Type: HTML RadioButton

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	RadioButton<>DOM::ElementPath	exists;<VarName>	

Description: Verifies whether the required radio button (Identified by DOM Path) exists or not. And stores the value

'True' in the specified variable if it exists, else stores the value 'False'

Example:

Checks whether the specified checkbox exists.

Check|RadioButton<>DOM::<DOMPath>|exists;Var1

43 .Checked

Object Type:

HTML RadioButton

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	RadioButton<>DOM::ElementPath	checked;<VarName>	

Description:

Verifies whether the required radio button (identified by DOM Path) has been selected. And stores the value 'True' in the specified variable if it has been selected else stores the value 'False'.

Example:

Checks whether the specified radio button is checked.

Check|RadioButton<>DOM::<DOMPath>|checked;Var1

44 .Unchecked

Object Type:

HTML RadioButton

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	RadioButton<>DOM::ElementPath	unchecked;<VarName>	

Description:

Verifies whether the required radio button (identified by DOM Path) has been deselected. And stores the value 'True' in the specified variable if it has been deselected else stores the value 'False'.

Example:

Checks whether the specified radio button is checked.

Check|RadioButton<>DOM::<DOMPath>|unchecked;Var1

45 .Exists

Object Type:

HTML Dropdown

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
--------	-------------------	---------------	---------------

Check	DropDown<>HTML::Value;Tag;Property	exists;<VarName>	
-------	------------------------------------	------------------	--

Description: Verifies whether the required dropdown (identified by HTML Parameters) exists. Stores the value 'True' in the specified variable if it exists else stores the value 'False'

Example: Checks whether the specified dropdown exists
Check | DropDown<>HTML::lang;select;name | exists;Var1

46 .Enabled

Object Type: HTML Dropdown

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	DropDown<>HTML::Value;Tag;Property	enabled;<VarName>	

Description: Verifies whether the required dropdown (identified by HTML Parameter) is enabled. Stores the value 'True' in the specified variable if it is enabled else stores the value 'False'.

Example: Checks whether the specified dropdown is enabled
Check | DropDown<>HTML::lang;select;name | enabled;Var1

47 .Disabled

Object Type: HTML Dropdown

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	DropDown<>HTML::Value;Tag;Property	disabled;<VarName>	

Description: Verifies whether the required dropdown (identified by HTML Parameter) is disabled. Stores the value 'True' in the specified variable if it is disabled else stores the value 'False'.

Example: Checks whether the specified dropdown is disabled
Check | DropDown<>HTML::lang;select;name | disabled;Var1

48 .Exists

Object Type: HTML Dropdown

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	DropDown<>DOM::ElementPath	exists;<VarName>	

Description: Verifies whether the required dropdown (identified by DOM Path) exists. Stores the value 'True' in the specified variable if it exists else stores the value 'False'

Example: Checks whether the specified dropdown exists
Check | DropDown<>DOM::<DOMPath> | exists;Var1

49 .Enabled

Object Type: HTML Dropdown

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	DropDown<>DOM::ElementPath	enabled;<VarName>	

Description: Verifies whether the required dropdown (identified by DOM Path) is enabled. Stores the value 'True' in the specified variable if it is enabled else stores the value 'False'.

Example: Checks whether the specified dropdown is enabled
Check | DropDown<>DOM::<DOMPath> | enabled;Var1

50 .Disabled

Object Type: HTML Dropdown

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	DropDown<>DOM::ElementPath	disabled;<VarName>	

Description: Verifies whether the required dropdown (identified by DOM Path) is disabled. Stores the value 'True' in the specified variable if it is disabled else stores the value 'False'.

Example: Checks whether the specified dropdown is disabled

Check|DropDown<>DOM::<DOMPath>|disabled;Var1

51 .Enabled

Object Type: HTML ListBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	ListBox<>HTML::Value;Tag;Property	enabled;<VarName>	

Description: Verifies whether the required Listbox(identified by HTML Parameters) is enabled or not. And stores the value 'True' if it is enabled else stores the value 'False' in the specified variable.

Example: Checks whether the specified Listbox is enabled
Check|Listbox<>HTML::lang;multiselect;name|enabled;Var1

52 .Disabled

Object Type: HTML ListBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	ListBox<>HTML::Value;Tag;Property	disabled;<VarName>	

Description: Verifies whether the required Listbox(identified by HTML Parameters) is disabled or not. And stores the value 'True' if it is disabled else stores the value 'False' in the specified variable.

Example: Checks whether the specified Listbox is disabled
Check|Listbox<>HTML::lang;multiselect;name|disabled;Var1

53 .Exists

Object Type: HTML ListBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	ListBox<>HTML::Value;Tag;Property	exists;<VarName>	

Description: Verifies whether the required Listbox(identified by HTML Parameters) exists or not. And stores the value 'True' if it exists else stores the value 'False' in the specified variable.
 Example: Checks whether the specified Listbox exists.
 Check|Listbox<>HTML::lang;multiselect;name|exists;Var1

54 .Enabled

Object Type: HTML ListBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Listbox<>DOM::ElementPath	enabled;<VarName>	

Description: Verifies whether the required Listbox(identified by DOM Path) is enabled or not. And stores the value 'True' if it is enabled else stores the value 'False' in the specified variable.
 Example: Checks whether the specified Listbox is enabled
 Check|Listbox<>DOM::<DOMPath>|enabled;Var1

55 .Disabled

Object Type: HTML ListBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Listbox<>DOM::ElementPath	disabled;<VarName>	

Description: Verifies whether the required Listbox(identified by DOM Path) is disabled or not. And stores the value 'True' if it is disabled else stores the value 'False' in the specified variable.
 Example: Checks whether the specified Listbox is disabled
 Check|Listbox<>DOM::<DOMPath>|disabled;Var1

56 .Exists

Object Type: HTML ListBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Listbox<>DOM::ElementPath	exists;<VarName>	

Description: Verifies whether the required Listbox(identified by DOM Path) exists or not. And stores the value 'True' if it exists else stores the value 'False' in the specified variable.

Example: Checks whether the specified Listbox exists.
Check | Listbox<>DOM::<DOMPath> | exists;Var1

57 .Exists

Object Type: HTML Image

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Image<>HTML::Value;Tag;Property	exists;<VarName>	

Description: Verifies whether the mentioned image (identified by HTML Parameter) exists or not. And stores the value 'True' when it exists else stores the value 'False'

Example: Checks whether the specified image exists.
Check | Image<>HTML::img1;input;name | exists;Var1

58 .Exists

Object Type: HTML Image

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Check	Image<>DOM::ElementPath	exists;<VarName>	

Description: Verifies whether the mentioned image (Identified by DOM Path) exists or not. And stores the value 'True' when it exists else stores the value 'False'

Example: Checks whether the specified image exists.
Check | Image<>DOM::<DOMPath> | exists;Var1

4 .Keyword for Storing Value

1 .StoreValue

Object Type: HTML TextBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
StoreValue	TextBox<>HTML::Value;Tag;Property	read;<VarName>	

Description: Captures the value of the required field (identified by HTML Parameter) in the specified variable name

Example: Reads the value of the specified Textbox into a variable
StoreValue | TextBox<>HTML::UserName;input;ID | read;Var1

2 .StoreValue

Object Type: HTML TextBox

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
StoreValue	TextBox<>DOM::ElementPath	read;<VarName>	

Description: Captures the value of the required field (identified by DOM Path) in the specified variable name

Example: Reads the value of the specified Textbox into a variable
StoreValue | TextBox<>HTML::UserName;input;ID | read;Var1

3 .StoreValue

Object Type: Text

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
GetHtmlElement	ElementPath	<VarName>	

Description: Captures the value of the required HTML text on the screen in the specified variable. DOM Path of the required element should be passed as an input.

Example: Stores the text of the specified DOM Path into a variable
StoreValue | GetHtmlElement | <DOMPath>

4 .StoreValue

Object Type: HTML Link

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
StoreValue	Link<>HTML::Value;Tag;Property	text;<VarName>	

Description: Stores the text of the link (identified by HTML Parameter) in the specified variable.

Example: Stores the text of the specified link in to a variable.
StoreValue | Link<>HTML::LgOut;input;name | text;Var1

5 .StoreValue

Object Type: HTML Link

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
StoreValue	Link<>DOM::ElementPath	text;<VarName>	

Description: Stores the text of the link (identified by DOM Path) in the specified variable.

Example: Stores the text of the specified link in to a variable.
StoreValue | Link<>DOM::<DOMPath> | text;Var1

6 .StoreValue

Object Type: HTML Dropdown

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
StoreValue	Dropdown<>HTML::Value;Tag;Property	ddown;<VarName>	

Description: Stores the value of selected dropdown (identified by HTML Parameter) element in the specified variable.

Example: Stores the value selected in the specified dropdown in a variable.
StoreValue | Dropdown<>HTML::lang;select;name | ddown;Var1

7 .StoreValue

Object Type: HTML Dropdown

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
StoreValue	Dropdown<>DOM::ElementPath	ddown;<VarName>	

Description: Stores the value of selected dropdown (identified by DOM Path) element in the specified variable.

Example: Stores the value selected in the specified dropdown in a variable.
StoreValue | Dropdown<>DOM::DOMPath | ddown;Var1

5 .Keyword for Wait Operation

1 .Wait

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Wait	Value;Tag;Property	0	

Test run would wait until the element specified by HTML parameter is being displayed in the application.
Default Timeout value is 10 sec.

Description: System would display an error message if the mentioned element is not found in 10 secs.

Example: Waits until the specified object loads or Default Timeout occurs
Wait | lang;multiselect;name

2 .Wait

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Wait	Value;Tag;Property	<Tsec>	

Description: Test run would wait until the element specified by HTML parameter is being displayed in the application.

System would display an error message if the mentioned element is not found in <Tsec> secs.

Tsec should be given in the format '0:00:00'

Example:

Wait until the specified object loads or the time out occurs

Wait|lang;multiselect;name|15

3 .Wait

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Wait	0	<Tsec>	

Description: Test run would wait for the specified time interval. Tsec should be given in the format '0:00:00'

Example:

Waits for the specified period

Wait||15

6 .Keyword for Verify/Compare

1 .Verify Text

Object Type: HTML Text

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Verify	ElementPath;<ExpectedText>	present	

Description: Verifies the presence of the expected text in the web page. Expected Text can be a wildchar character.

Example:

Verifies the presence of specified text.

Verify|<DOMPath>;Welcome|present

2 .Verify Text

Object Type: HTML Text

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
--------	-------------------	---------------	---------------

Verify	ElementPath;<ExpectedText>	present;<VarName>	
--------	----------------------------	-------------------	--

Verifies the presence of the expected text in the web page. And stores the value 'True' if the text present else stores the value 'False' in the specified variable.

Description: Expected Text can be a wildchar character.
 Example: Verifies the presence of specified text and stores 'True' in a variable if the text presents, else stores 'False'.
 Verify | <DOMPath>;Welcome | present;Var1

3 .Verify Text

Object Type: HTML Text

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Verify	ElementPath;<ExpectedText>	absent	

Description: Verifies the presence of the expected text in the web page. Expected Text can be a wildchar character.
 Example: Verifies the absence of specified text.
 Verify | <DOMPath>;Welcome | absent

4 .Verify Text

Object Type: HTML Text

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Verify	ElementPath;<ExpectedText>	absent;<VarName>	

Verifies the absence of the expected text in the web page. And stores the value 'True' if the text is not present else stores the value 'False' in the specified variable.

Description: Expected Text can be a wildchar character.
 Example: Verifies the absence of specified text and stores 'True' in a variable if the text is absent, else stores

'False'.

Verify|<DOMPath>;Welcome|absent;Var1

5 .Verify Variable

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
VerifyVar	Var;<ExpectedValue>	like	

Description: Verifies whether the value of the variable is like the <Expected Text>. Expected Text can be a wild char character

Example: Verifies the value of the variable is like the expected value
VerifyVar|Var1;India|like

6 .Verify Variable

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
VerifyVar	<VarName>;<ExpectedValue>	equals	

Description: Verifies whether the value of the variable is equal to the expected value.

Example: Verifies whether the value of the variable is equal to the expected value
VerifyVar|Var1;100|equals

7 .Verify Variable

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
VerifyVar	<VarName>;<ExpectedValue>	lessthan	

Description: Verifies whether the value of the variable is lesser than the expected value.

Example: Verifies whether the value of the variable is lesser than the expected value
VerifyVar|Var1;100|lessthan

8 .Verify Variable

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
VerifyVar	<VarName>;<ExpectedValue>	greaterthan	

Description: Verifies whether the value of the variable is greater than the expected value.

Example: Verifies whether the value of the variable is greater than the expected value
VerifyVar|Var1;100|greaterthan

9 .Verify Variable

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
VerifyVar	<VarName>;<ExpectedValue>	notequals	

Description: Verifies whether the value of the variable is not equal to the expected value.

Example: Verifies whether the value of the variable is not equal to the expected value
VerifyVar|Var1;100|notequals

10 .Compare Variable

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
CmprVar	<Var1>;<Var2>	like	

Description: Verifies whether the values of the variables <var1> and <var2> are identical

Example: Verifies the value of the variable Var1 is like Var2

CmprVar|Var1;Var2|like

11 .Compare Variable

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
CmprVar	<Var1>;<Var2>	equals	

Description: Verifies whether the value of <Var1> is equal to the value of <Var2>.

Example: Verifies whether the value of the variable is equal to the value of the other variable
CmprVar|Var1;Var2|equals

12 .Compare Variable

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
CmprVar	<Var1>;<Var2>	lessthan	

Description: Verifies whether the value of <Var1> is lesser than the value of <Var2>.

Example: Verifies whether the value of the variable is lesser than the value of the other variable
CmprVar|Var1;Var|lessthan

13 .Compare Variable

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
CmprVar	<Var1>;<Var2>	greaterthan	

Description: Verifies whether the value of <Var1> is greater than the value of <Var2>

Example: Verifies whether the value of the variable is greater than the value of the other variable.
CmprVar|Var1;Var2|greaterthan

14 .Compare Variable

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
CmprVar	<Var1>;<Var2>	notequals	

Description: Verifies that the values of <Var1> and <Var2> are not equal

Example: Verifies whether the value of the variable is not equal to the value of the other variable.

CmprVar|Var1;Var2|notequals

7 .Keyword to denote end of TC

1 .End of Test Case

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
end	0	0	

Description: Indicates the End of Test Case. Every Test Case should end with 'end' keyword.

Example: Indicates the end of Test Case

end||

8 .Keyword for Databank

1 .Databank Variable Count

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
DVarCnt	<Value>	0	

Description: By default user can have 20 different Data Bank Variable. 'DVarCnt' function increases the Data Bank Variable Count to the required value.

Example: Increases the no of Databank Variable count to 38.

DVarCnt|38

2 .Set with Databank Value

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Dbank	TextBox<>HTML::Value;Tag;Property	set;<Dbank Variable>	

Description: Binds the mentioned Text Field (Identified by HTML Parameters) with the Data Bank Variable.

Example: Sets the text box with the specified databank variable value.

Dbank|TextBox<>HTML::LgnBtn;input;name|set;Dvar1

3 .Set with Databank Value

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Dbank	TextBox<>DOM::ElementPath	set;<Dbank Variable>	

Description: Binds the mentioned Text Field (Identified by DOM Path) with the Data Bank Variable.

Example: Sets the text box with the specified databank variable value.

Dbank|TextBox<><DOMPath>|set;Dvar1

9 .Keyword for Setting Variable

1 .Set Variable

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
SetVariable	<VarName>	<VarValue>	

Description: Creates a new variable with specified name and initializes it with the specified value.

Example: Creates a variable with the specified value.

SetVariable|Var1|500

10 .Keyword for Condition

1 .If Construct

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
IfBegin	<VarName>;equals	<Value>	

Description: Executes the preceding set of code until 'IfEnd' keyword row, if Variable value is equal to specified value

Example: If construct begins if the variable value is equal to the provided value.

IfBegin | Var1;equals | 500

2 .If Construct

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
IfBegin	<VarName>;like	<Text>	

Description: Executes the preceding set of code until 'IfEnd' keyword row,if Variable value is like expected text.

Example: If construct begins if the variable value is like the provided value.

IfBegin | Var1;like | India

3 .If Construct

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
IfBegin	<VarName>;greaterthan	<Value>	

Description: Executes the preceding set of code until 'IfEnd' keyword row, if Variable value is greater than specified value

Example: If construct begins if the variable value is greater than the provided value.

IfBegin | Var1;greaterthan | 55

4 .If Construct

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
IfBegin	<VarName>;lesserthan	<Value>	

Description: Executes the preceding set of code until 'IfEnd' keyword row, if variable value is lesser than specified value

Example: If construct begins if the variable value is lesser than the provided value.
IfBegin | Var1;lesserthan | 95

5 .If Construct

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
IfBegin	<VarName>;notequals	<Value>	

Description: Executes the preceding set of code until 'IfEnd' keyword row, if variable value is not equal to the specified value.

Example: If construct begins if the variable value is not equal to the provided value.
IfBegin | Var1;notequals | 99

6 .If Construct

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
IfVar	<VarName>;equals	<Value>	

Description: Executes the preceding set of code until 'IfEnd' keyword row, if Variable1 value is equal to Variable2 value.

Example: If construct begins if the variable value is equal to the value of the other variable.
IfVar | Var1;equals | Var2

7 .If Construct

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
IfVar	<VarName>;like	<Text>	

Description: Executes the preceding set of code until 'IfEnd' keyword row,if Variable1 value is like Variable2.

Example: If construct begins if the variable value is like the value of the other variable.

IfVar | Var1;like | Var2

8 .If Construct

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
IfVar	<VarName>;greaterthan	<Value>	

Description: Executes the preceding set of code until 'IfEnd' keyword row, if Variable1 value is greater than variable2 value

Example: If construct begins if the variable value is greater than the value of the other variable.

IfVar | Var1;greaterthan | Var2

9 .If Construct

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
IfVar	<VarName>;lesserthan	<Value>	

Description: Executes the preceding set of code until 'IfEnd' keyword row, if variable1 value is lesser than variable2 value

Example: If construct begins if the variable value is lesser than the value of the other variable.

IfVar | Var1;lesserthan | Var2

10 .If Construct

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
IfVar	<VarName>;notequals	<Value>	

Description: Executes the preceding set of code until 'IfEnd' keyword row, if variable1 value is not equal to the variable2 value.

Example: If construct begins if the variable value is not equal to the provided value.
IfBegin | Var1;notequals | 99

11 .If Construct

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
IfEnd	0	0	

Description: Specifies the end of the preceding If loop. Nested IF Loop is not supportable in this framework.

Example: Indicates the end of If construct
IfEnd

11 .Keyword for MsgBox

1 .MsgBox

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
MsgBox	<VarName>	0	

Description: Displays the value of the Variable in MsgBox

Example: Displays the value of the variable in the MsgBox
Msgbox | Var1

12 .Keyword for Iterate

1 .Iterate

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Iterate	<no of iterations>	0	

Description: Iterates the TC for the specified no of times. Note that Iterate needs to be mentioned after the 'end' Keyword row of required TC.

Example: Iterates the Test Case for the specified no of times
Iterate|5

13 .Keyword for String Operations

1 .String Operations

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Strlen	<VariableName> (whose length needs to be identified)	<VarName> (to store the result)	

Description: Identifies the length of the variable value and stores it in the specified variable

Example: Stores the length of the variable value in another variable.
Strlen|Var1|Var2

2 .String Operations

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
StrReplace	<Variable1>;<str1>;<str2>	<Variable2>	

Description: Replaces string <str1> with <str2> in a variable <variable1> and stores the result in the variable <variable2>

Example: Replaces the substring from a main string with the specified string and stores the replaced value in a

variable.

StrReplace | Var1;Empirix;Oracle|Var2

3 .String Operations

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
strsearch	<Variable1>;<searchstr>	present;<variable2>	

Searches for the presence of <searchstr> in <variable1> and returns the result 'True' if it is present in the specified variable.

Description: Else returns the result 'False' in the specified variable.

Example: Checks the presense of the substring in a variable.

strsearch | Var1;Empirix | present;Var2

4 .String Operations

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
strsearch	<Variable1>;<searchstr>	absent;<variable2>	

Searches for the absence of <searchstr> in <variable1> and returns the result 'True' if it is absent in the specified variable.

Description: Else returns the result 'False' in the specified variable.

Example: Checks the absense of the substring in a variable.

strsearch | Var1;Empirix | absent;Var2

14 .Keyword for SQL Operation

1 .SQL

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
sqlexecute	<ConnectionString>;<sqlquery>	<DbankVariable>	

Description: Executes the provided query in the db and stores the result in the databank variable.

Example: Executes the provided query in the db and stores the result in the databank variable.
sqlexecute |<ConnectionString>;<sqlquery> |DVar1

2 .SQL

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
sqlopen	<connection string>	0	

Description: Opens the DB connection with reference to the connection string mentioned. Note that Test run can have only one active db connection per run. To open another db connection, close the existing connection and then open the new connection.

Example: Opens the connection with the specified db.
sqlopen |<connection string>

3 .SQL

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
sqlclose	0	0	

Description: Closes the existing db connection.

Example: Closes the existing open sql connection.
sqlclose

4 .SQL

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
sqlexecute	<querystr>	<Variable>	

Description: Executes the mentioned query in the open db connection and stores the result of the query in the specified variable.

Example: Executes the query in the open db and stores the result in the specified variable.
sqlexecute | <sqlquery> | Var1

15 .Keyword for Calling Function

1 .Call Function

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
CallFunction	<FunctionName>	<Param1>;<Param2>;<Param3>;...	

Description: This function calls the specified User Defined Function and supplies the required parameter to function as provided. Function parameters needs to be delimited with ';'.
Example: Calls the specified User Defined Function
CallFunction | Example1 | One;Two

16 .Keyword for Generating Random Number

1 .Random

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Random	<num>	<VariableName>	

Description: This function generates the random number and stores it in the specified variable.

Example: Generates the random number
Random | 10 | Ran1

17 .Keyword for Looping

1 .Loop Construct

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Loop	<num>	0	

Description: This function loops over the specified test steps over the required number of times

Example: Loops over the specified set of test steps for required number of times
Loop|4

2 .Loop Construct

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
endloop	0	0	

Description: This keyword denotes the end of the Loop Construct

Example: Denotes the end of the loop construct
endloop

18 .Keyword for Split

1 .Split

Object Type: General

Keyword Syntax:

Action	Object Parameters	Action Value1	Action Value2
Split	<MainString>^<Delimiter>	<v1>:<e1>;<v2>:<e2>	

Description: This keyword splits the main string as per the delimiter and stores it in a variable

Example: Splits the main string and stores it in a variable
Split|Keane-Inc^-|V1:1;V2;2

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.