



# Open2Test Test Automation Framework for OpenScript (WEB)

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**

<b>1. PURPOSE OF THE DOCUMENT</b> .....	<b>3</b>
1.1. Scope .....	3
1.2. Overview .....	3
<b>2. FRAMEWORK ARCHITECTURE</b> .....	<b>4</b>
<b>3. DRIVER FUNCTIONS</b> .....	<b>5</b>
3.1. Driver Class .....	5
3.2. Web_Framework Class .....	6
3.2.1. Assumptions.....	6
3.2.2. Constructor Parameters.....	6
3.2.3. Variables.....	6
3.2.4. Enum.....	7
3.2.5. Methods.....	7
<b>4. COMMON FUNCTIONS</b> .....	<b>17</b>
4.1. CommonFn Class .....	17
4.1.1. Methods.....	18
<b>5. USER-DEFINED FUNCTIONS</b> .....	<b>19</b>
5.1. UDF Class .....	19
5.1.1. Methods:.....	19

## 1. Purpose of the Document

The purpose of this document is to describe Open2Test Test Automation Framework for the tool OpenScript for the Web Technology in detail.

### 1.1. Scope

The scope of this document is to provide details about Open2Test Test Automation Framework code.

### 1.2. Overview

This document provides details about:

- Framework Architecture
- Driver Script
- Action Functions
- Common Functions
- User-Defined Functions

## 2. Framework Architecture

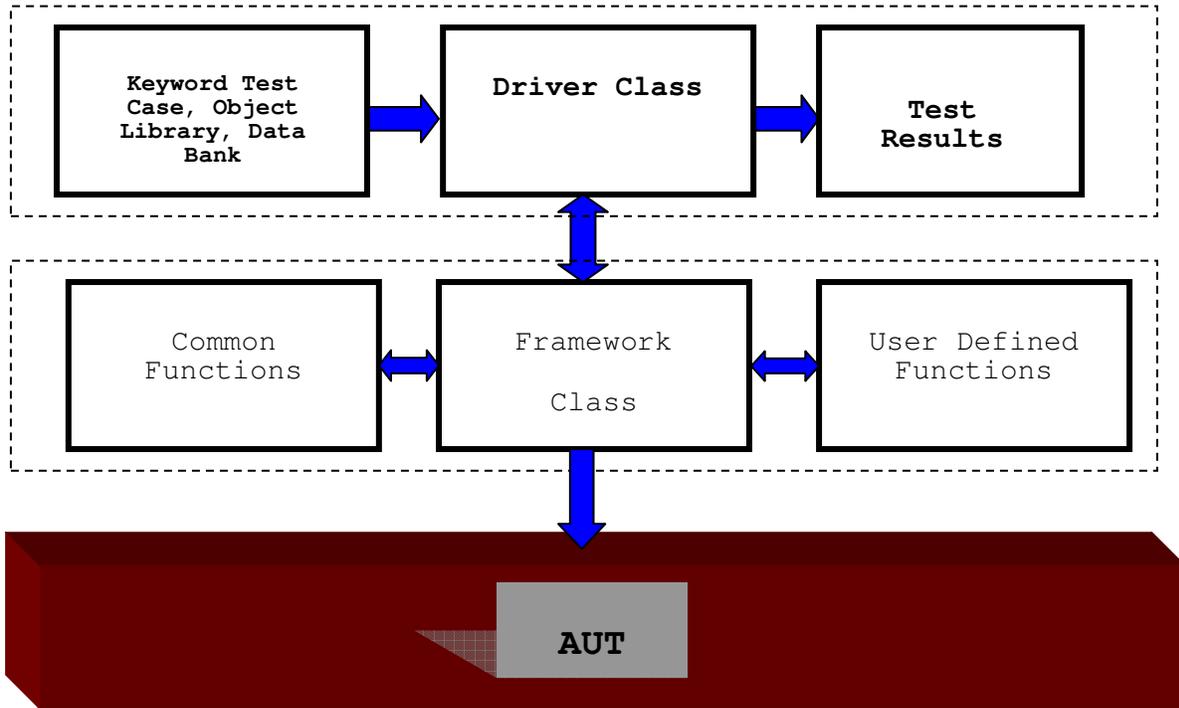


Figure 1 - Framework Architecture

### 3. Driver Functions

#### 3.1. Driver Class

**Name of the Class:** Script

**Description:** Driver Script is used to call the 'run' function of Web\_Framework class by creating an instance of it.

**Extends:** IteratingVUserScript Abstract Class

**Visibility:** Public

**Constructor Parameters:**

- a) Test Case Path: This parameter contains the drive path of the Keyword Test Case.
- b) ObjPath: This parameter contains the drive path of the Object Library of Application under Test.
- c) DBAlias (Optional): This parameter holds the alias name of the Databank file. Required Databank file needs to be configured with the driver script, and the alias name provided while configuring should be stored in this parameter.

In case of test scripts without Databank, a 'null' value should be stored in this parameter.

**Assumptions:**

- a) Web\_Framework class should be set in the environment as described in 'Oracle OpenScript Quick Start Guide'.

**Variables:**

- b) F: Variable is used to store the instance of the Framework Class.
- c) iVuser: This variable is used to store the IScriptVUser object of the <TestName> class instance.
- d) IwDOMServ: This variable is used to store the InternalWebDOMService object of the WebDOMService Object, which is used to retrieve 'PlayBackHandler' object.
- e) Pbhandler: This variable is used to store the PlayBackHandler Object of InternalWebDOMService, which is used to retrieve the object connection.
- f) Iconn: This variable is used to store the connection object of PlayBackHanlder, which is used to establish connection with Web\_Framework Object.

**Functionality:**

- Driver Function creates an instance of the Web\_Framework Object by passing the required parameter values.
- Establishes connection with Web\_Framework Object using 'Set' method of the Web\_Framework with the parameters iVuser and Iconn.
- Executes the 'run' method of Web\_Framework Class.

## 3.2. Web\_Framework Class

**Name of the Class:** Web\_Framework

**Description:** Web\_Framework Class contains all the essential methods that are required to convert the keywords mentioned in Test Case document to action. Driver scripts create the instance of Web\_Framework Class to implement the Keyword Driven Framework.

**Extends:** IteratingVUserScript Class

**Visibility:** Public

### 3.2.1. Assumptions

User has provided valid values of Keyword Test Case Path, Object Library Path, and Alias name of Databank.

### 3.2.2. Constructor Parameters

1. Keyword Test Case Path
2. Object Library Path
3. Databank Alias Name

### 3.2.3. Variables

StpCnt: Holds the integer value of the row count of the Keyword Test Case.

Currow: Holds the row object of the Keyword Test Case.

KTCASE: Holds the table object of the Keyword Test Case.

map: Map Object used to store the Key-Value pair.

COL1, COL2, COL3, COL4, COL5: Constants that hold the values 0,1,2,3,4 respectively, using which corresponding column values of the Currow object can be retrieved.

loopcnt: loopcnt is the list object which holds the number of times a code needs to be looped.

loopind: loopind is the list object which holds the index of the loops. It is used to find the index of the loops in case of nested loops.

lpstrtrow: lpstrtrow is the list object which stores the value of the start row of the loop.

Itrvu: Variable used to hold the IteratingVUser object of the Web\_Framework instance.

ft: Variable used to hold the FunctionalTestService object of the Framework instance.

Utilities: Variable used to hold the UtilitiesService object of the Framework instance.

Browser: Variable used to hold the BrowserService object of the Framework instance.

Web: Variable used to hold the WebDOMService object of the Framework instance.

Iwd: Variable used to hold the InternalWebDOMService object of the Framework instance.

Ph: Variable used to hold the PlaybackHandler object of the Framework instance.

iConnection: Variable used to hold the connection object of the Framework instance.

strKword: Stores the value of the keyword available in the currow object.

[] arrObj: Stores the values available under Object column delimited with ';'.

strObjtype: Stores the value of the Object Type mentioned in the Object column of the Keyword Test Script.

strObjpath: Stores the alias name of the Object mentioned in the Object column of the Keyword Test Case.

strObjpath1: Stores the 3<sup>rd</sup> delimited value mentioned under Object column of the Keyword Test Case.

strActtype: Stores the value of the Action type mentioned in the Action Value column of the Keyword Test Case.

strActvall: Stores the 2<sup>nd</sup> delimited value mentioned under the Action Value column of the Keyword Test Case.

strActval2: Stores the 3<sup>rd</sup> delimited value mentioned under the Action Value column of the Keyword Test Case.

Manstep: Stores the value of the manual test description mentioned under the Manual Test Description column of the Keyword Test Case.

Rownum: Stores the row number of the currow object in the Keyword Test Case.

#### 3.2.4. Enum

Keyword: Contains the various keyword values defined in the framework.

Objects: Contains the various Object names that the framework supports.

Actions: Contains the various Action types that the framework supports.

Attributes: Contains the various Attribute names that the framework can identify.

#### 3.2.5. Methods

##### 3.2.5.1. Set

**Description:** Set method of Framework class is used to establish relation between various internal objects of Driver class with the internal objects of Web\_Framework Class.

**Visibility:** Public

**Parameters:** IScriptVUser, Connection

**Function:**

Initializes the following objects of Web\_Framework class with the IteratingVUser object of driver class:

1. FunctionalTestService
2. Utilities Service
3. Browser Service
4. WebDOMService
5. InternalWebDOMService

**Exception Handling:**

General Exception Handling has been covered.

**3.2.5.2. Run**

**Description:** Run is an abstract method of IteratingVUserScript class. Run method acts as a basic working block of Web\_Framework class. It skims through every row of Keyword Test Case and calls the appropriate function as defined.

**Visibility:** Public

**Parameters:** none

**Functions:**

1. Loads Object Library into Web\_Framework class from the mentioned path.
2. Skims through every row of the Keyword Test Case and calls the appropriate function.

**Exception Handling:**

Following exceptions are handled under Run Method:

1. IndexOutOfBounds Exception
2. CSV Exception
3. AbstractScript Exception
4. IllegalArgument Exception
5. Exception

**Function:**

1. Run method implements the following actions in Test Automation Framework:
  - a. User can perform the required operation on the AUT object (Refer Func\_Perform section).
  - b. User can check the value of the required object's attribute (Refer Func\_StoreCheck section).
  - c. User can store the value of the object attribute in required variable (Refer Func\_StoreCheck section).
  - d. User can halt the system performance for the required time interval (Refer Func\_Wait section).
  - e. User can launch the browser.
  - f. Value of the dictionary variable/databank variable can be displayed on the results pane as comments using MsgBox keyword.

- g. User can assign the required value to the specified variable.
- h. User can perform the required arithmetic operations on the variables (Refer Func\_Arith section).
- i. User can evaluate a condition and then can proceed to set of test steps based on the result (Refer Func\_Condition section.)
- j. User can specify the end of the condition construct using 'endCondition' keyword. Supports nested condition loops.
- k. User can loop over set of test steps for the required number of times.
- l. User can specify the end of the loop construct using 'endloop' keyword. Supports nested loops.
- m. Calls the reusable actions using 'CallAction' keyword.
- n. Measures the length of the provided string and stores it in the variable.
- o. Replaces the substring with a string in Main string and stores the result in the variable.
- p. Verifies whether the main string contains the substring or not.
- q. Concatenates two strings and stores the resulting string in the variable.
- r. Calls the required user-defined function and stores the return value of the function in the specified variable using 'CallFunction' keyword (Refer UDF Class section).
- s. Calls the required common function and performs the file system object operation (Refer CommonFn class).
- t. Define a label in Keyword Test Case.
- u. Jump to the required label defined in the Keyword Test Case.
- v. Iterate/loop over the values of the data bank file using 'getnextdb' keyword.

#### 3.2.5.3. Func\_Wait:

**Description:** System halts the test execution for the specified time period.

**Visibility:** private

**Parameters:** none

**Function:** System halts the test execution for the specified time period as mentioned in the Object column of Keyword Test Case.

#### 3.2.5.4. Func\_arith:

**Description:** System performs the arithmetic operation on the operands as specified in the Keyword Test Case.

**Visibility:** private

**Parameters:** none

**Functions:**

System performs addition, subtraction, multiplication, division and power of functions on the operands as specified in the Keyword Test Case.

#### 3.2.5.5. Func\_Condition

**Description:** This function is used to evaluate the expression according to the inputs given in the keyword script and to proceed with the next set of test steps.

**Parameters:** none

**Visibility:** private

**Variables:**

- a) Cmpres: Boolean variable that gives the result of comparison.
- b) Condflag: Acts as a condition flag, which is used in determining the end row of the condition in case of nested condition constructs.
- c) Temprow: Row Object variable, used to store the value of the current object.

**Functionality:**

- Executes the preceding set of test steps until the corresponding 'endcondition' keyword row, if the expression mentioned in the 'Condition' keyword row is true.
- System skips to the row next to the corresponding 'endcondition' keyword if the expression mentioned is false.
- Supports nested if constructs.
- Comparison Operators supported are equals, notequals, lessthan, greaterthan, lessthanorequalto, greaterthanorequalto, like.
- Executes the comparison by calling 'Func\_cmpr' function.

#### 3.2.5.6. Func\_cmpr

**Description:** This function compares two values based on an operator and returns a Boolean result as 'True' if the comparison passes; otherwise, it returns 'False'.

**Parameters:**

- a) Val1, val2 - Variable values to be compared against each other.
- b) Oper - Operator

**Visibility:** private

**Variables:**

Blnres: Boolean value that indicates the results of the comparison.

**Functionality:**

1. Compares value1 with value2 based on the operator and returns the result.
2. Comparison Operators supported are equals, notequals, lessthan, greaterthan, lessthanorequalto, greaterthanorequalto, like.

#### 3.2.5.7. FetchData

**Description:** This function is used to fetch the data from the Keyword Test Case sheet, split them using a delimiter as applicable, and assign them to class variables - which will be used later in processing.

**Parameter:**

a) Currow: Holds the row object of the current row, which is used in fetching the data.

**Visibility:** private

**Variables:** none

### 3.2.5.8. ClearData

**Description:** This function is used to clear the values of Keyword Test Case variable before every fetch data operation.

**Parameter:** none

**Visibility:** private

**Variables:** none

### 3.2.5.9. getValue()

**Description:** This function is used to retrieve the values of the dictionary variables and databank variables.

**Parameter:**

1. colpar: Name of the parameter

**Visibility:** private

**Variables:**

1. colValue: Holds the name of the dictionary variable and Databank variable after removing the prefixes.

**Function:**

1. Returns the value of the variable as such, if 'colpar' doesn't begin with any prefixes.
2. Returns the value of the Dictionary element 'ColValue', if 'colpar' begins with '#'.
3. Returns the value of the Databank variable 'ColValue', if 'colpar' begins with 'dt\_'.

### 3.2.5.10. Func\_Eloop

**Description:** This function is used to find the value of the Loop Start Row while implementing the Loop Keyword.

**Parameter:** none

**Visibility:** private

**Variable:**

1. Lpnum: Used to hold the count of number of loops in case of nested loops.

**Function:**

1. Finds the value of the Loop start row, to loop over the set of test steps for the specified number of times.
2. Supports nested loop.

#### 3.2.5.11. Func\_goto

**Description:** This function is used in 'JumpTo' keyword, to jump to the specified label.

**Parameter:** none

**Visibility:** private

**Variable:**

Checkrow: Row object variable, used to store the row object of Keyword Test Case.

Aflag: Used to store the value of 'automate' column in Keyword Test Case.

strKword: Used to store the value of the Keyword in the specified row of Keyword Test Case.

islabel: Used to store the name of the label in the specified row of Keyword Test Case.

**Function:**

1. Makes the control to jump to the required predefined label (test step) of Keyword Test Script.
2. Throws an error message if a test case doesn't have the required label.

#### 3.2.5.12. Func\_perform

**Description:** This function is used to perform the required operation on the objects of AUT.

**Parameter:** None

**Visibility:** private

**Variable:**

Obj: General Object 'Obj' is used to hold the object returned by 'FindObj' method.

**Function:**

1. Identifies the object by calling the function 'Func\_FindObj' and by using the parameters of the 'Object' column.
2. Performs the required action on the identified object by calling the function 'Func\_Action' and by using the parameters of the 'Action' column.

#### 3.2.5.13. Func\_StoreCheck

**Description:** This function is used to either check the value of the required object attribute or to store the value of the required object attribute in the provided variable.

**Parameter:** None

**Visibility:** private

**Variable:**

Obj: General Object 'Obj' is used to hold the object returned by 'Func\_FindObj' method.

Actual: String variable 'actual' is used to store the actual value of the attribute returned by 'Func\_attrVal'.

Expval: String variable 'expval' is used to store the expected value of the attribute, fetched from keyword test case.

Varname: String variable 'varname' is used to store the variable name provided by the user.

**Function:**

1. Identifies the object by calling the function 'Func\_FindObj' and by using the parameters of the 'Object' column.
2. Retrieves the required attribute value of the object using 'Func\_attrVal' function.
3. In case of 'Check' keyword, compares the retrieved attribute value with the provided expected value and logs the result.
4. In case of 'StoreValue' keyword, stores the retrieved attribute value in the required variable.

#### 3.2.5.14. Func\_FindObj

**Description:** This function is used to identify the Web objects using DOM Path of the corresponding element and returns the identified object to the calling function. Elements Browser, AlertDialog, ConfirmDialog, and dialog are found using the text displayed on it.

**Parameter:** none

**Visibility:** private

**Variable:**

strPath: String variable used to store the path of the Object Library in the required format.

Bpath: String variable used to identify the elements based on the text displayed on it.

**Function:**

The following Web elements can be identified using Func\_FindObj method:

1. Browser: Identified using the text displayed on the 'Title' of the browser. Supports wild card pattern.
2. Following elements are identified using the text displayed on the respective dialog boxes: AlertDialog, Dialog, and ConfirmDialog.
3. Following Web elements are identified using DOM Path of the respective elements:

Button, Checkbox, element, document, image, link, select box, radio button, textbox, table.

Note: Of the above mentioned elements, 'element' can be used to find any Web element (button, textbox, tree view, HTML Frame etc).

4. Performs following SQL Operations - SQLConnect, SQLDefine, SQLDisconnect, SQLExecute.

Note: a. Open2Test supports 'ODBC' connectivity.

b. Required Data Source needs to be defined in Control Panel  
-> Administrative Tools -> Data Sources (ODBC) -> System  
DSN.

5. Generates Random Number

6. Splits the variable value based on the delimiter.

### 3.2.5.15. Func\_Action

**Description:** This function is used to perform actions on the objects identified using 'Func\_FindObj' function.

**Parameter:** Generic Object Obj

**Visibility:** private

**Variables:** none

**Function:**

Func\_Action method performs following function on the objects identified using 'Func\_FindObj' method:

1. Click - Simulates the click event in the object.
2. DoubleClick - Simulates the double click event in the object.
3. RightClick - Simulates the right click event in the object.
4. Keypress - Simulates the key press event in the object.  
Note: Keys must be formatted as: <CTRL-SHIFT-A><ENTER><ALT-F5>
5. MouseOver - Simulates mouse over action on the object.
6. AlertOk - Clicks on the OK button of the Alert Dialog box.
7. Navigate - Navigates the browser object to the required url.
8. Waitforpage - System halts the test execution until required Web page is loaded on the browser.
9. Maximize - Maximizes the browser object.
10. Minimize - Minimizes the browser object.
11. Back - Navigates back to the previous page of the browser object.
12. Forward - Simulates the forward action on the browser object.
13. Refresh - Simulates the refresh action in browser object.
14. SetLocation - Sets the location and size of the browser as specified.
15. Close - Closes the browser object.
16. ConfirmCancel - Clicks on the Cancel button of the Confirm Dialog box.
17. ConfirmOk - Clicks on the OK button of the Confirm Dialog box.
18. ClkDialogBtn - Clicks on the mentioned button index of the Dialog box
19. LgnDialogOk - Clicks on the OK button of the Login Dialog box.
20. Check - Checks/unchecks the checkbox object.
21. Settext - Sets the specified text in the textbox object.

22. Setpassword - Sets the specified text as password in the textbox object.
23. Cleartext - Clears the text present in the textbox object.
24. Setcurrentdate - Sets the current date in the specified textbox object.
25. Setcurrenttime - Sets the current time in the specified textbox object.
26. Setd - Sets the text box with the date value of the current date (+/-) required value.
27. Setm - Sets the text box with the date value of the current month (+/-) required value.
28. Sety - Sets the text box with the date value of the current year (+/-) required value.
29. SelectText: Selects the specified text in the dropdown/combobox/listbox object.
30. SelectIndex: Selects the specified index in the dropdown/combobox/listbox object.
31. Multiselecttext: Multi selects the specified text values in the list box object.
32. Multiselectindex: Multi selects the specified index values in the list box object.
33. ShowContextMenu: Shows the context menu on the required Web element.
34. ClickContextMenu: Clicks on the required item of the Context Menu on the Web element.

#### 3.2.5.16. Func\_attrVal

**Description:** This function is used to find the value of the required object attribute and returns the value.

**Parameter:** Generic Object Obj

**Visibility:** private

**Variable:**

actVal: String variable used to store the actual value of the object's attribute, which the system returns to the calling function.

**Function:**

Func\_attrVal method finds the value of the following object attribute and returns it to the calling function:

Disabled: Captures the value of the disabled property of the object.

Checked: Captures the value of the checked attribute of the checkbox.

Readonly: Captures the value of the object's read only attribute.

Name: Captures the name of the object.

Value: Captures the value of the text box.

SelectedText: Captures the value of the selected text in the dropdown/combobox/listbox object.

SelectedMultiText: Captures the value of the selected texts in the Listbox object.

**Note:** Captures the values selected in Listbox and concatenates the values with the delimiter ';'.

StatusBarText: Captures the text available in the status bar of the browser.

Ddownlen: Captures the number of elements of the dropdown/combobox/listbox object.

Radiooption: Captures the status of the radio button (whether it's selected or not) and stores the result.

innertext: Verifies whether the provided text is present on the browser at the specified DOM path or not.

Getcellcnt: Returns the count of cells in the Web table object.

Getrowcnt: Returns the count of rows in the Web table object.

Getcolcnt: Returns the count of columns in the Web table object.

### 3.2.5.17. Func\_Convert

**Description:** This function converts a provided value from one form to another and stores the resulting value in a provided variable.

**Parameter:** None

**Visibility:** Private

**Function:**

1. Rounds the provided decimal value to the nearest integer.
2. Converts the provided variable value to lower case.
3. Converts the provided variable value to upper case.
4. Trims the variable value.
5. Converts a decimal value to a long value.
6. Converts a string variable to integer variable.
7. Converts a date to the required format.

## 4. Common Functions

### 4.1. CommonFn Class

**Name of the class:** CommonFn

**Description:** CommonFn class contains the functions used to perform FSO related operations.

**Extends:** IteratingVUserScript Abstract Class

**Visibility:** Public

**Constructor Parameters:**

ISvu - IscriptVUser of calling class, usually of Web\_Framework class.

iConn - Connection object of calling class, usually of Web\_Framework class.

Numrow - Stores the calling row number of the Keyword Test Case.

**Constructor Function:**

CommonFn constructor assigns the IscriptVUser and Connection Object values of the calling object with that of CommonFn object and derives the values of various objects like Utilities Service, Browser Service, etc. from the value of IteratingVUser object.

**Assumption:**

Instance of CommonFn should be created from Web\_Framework class.

**Variables:**

Retval: String Variable used to store the return value.

Cmn\_Ut: Instance of UtilitiesService object, will be assigned with the value of calling object.

Cmn\_Br: Instance of Browser Service object, will be assigned with the value of calling object.

Cmn\_fts: Instance of Functional Test Service object, will be assigned with the value of calling object.

Cmn\_Wds: Instance of WebdomService object, will be assigned with the value of calling object.

Cmn\_Ph: Instance of Playback Handler object, will be assigned with the value of calling object.

Cmn\_Iwds: Instance of Internal Web DOM Service

FS: Instance of File Service object, used to perform FSO related operations.

Rownum: Integer variable, used to store the value of calling row number of Keyword Test Case.

**Enum:**

Operations: Enum used to store the keywords of various FSO related operations.

#### 4.1.1.1. Methods

##### 4.1.1.1.1. Execute

**Description:** Execute method is used to perform various File System Object related functions like create, delete, and copy Files and Folders.

**Visibility:** Public

**Parameters:**

arrObj[]: String array variable that describes the type of object and the kind of action to be performed. Fetched from the 'Objects' column of Keyword Test Case.

arrAct[]: String array variable that describes the attributes values, such as drive path of folder/file, overwrite flag, etc. - fetched from Action Value to be used while performing FSO Operations.

**Function:**

The following FSO functions can be performed using this Web\_Framework:

1. Creating a folder in the specified path.
2. Deleting the folder and its contents from the specified path.
3. Copying the contents of the folder from the specified source path to the destination path.
4. Creating a file in the specified path.
5. Deleting a file from the specified path.
6. Copying the contents of the file from the specified source path to the destination path.
7. Appending a value to the file.
8. Reading a value from the file.
9. Measuring the line count of the file.

**Exception Handling:** Throws exception to the calling function.

## 5. User-Defined Functions

### 5.1. UDF Class

**Name of the class:** UDF

**Description:** UDF Class contains the User-Defined Functions that can be called from Web\_Framework Class.

**Extends:** IteratingVUserScript Abstract Class

**Constructor Parameters:**

ISvu - IscriptVUser of calling class, usually of Web\_Framework class.

iConn - Connection object of calling class, usually of Web\_Framework class.

Numrow - Stores the calling row number of the Keyword Test Case.

**Constructor Function:**

UDF constructor assigns the IscriptVUser and Connection Object values of the calling object with that of CommonFn object and derives the values of various objects like Utilities Service, Browser Service, etc. from the value of IteratingVUser object.

**Assumption:**

Instance of UDF Class should be created from Web\_Framework class.

**Variables:**

Retval: String Variable used to store the return value.

UDF\_Ut: Instance of UtilitiesService object, will be assigned with the value of calling object.

UDF\_Br: Instance of Browser Service object, will be assigned with the value of calling object.

UDF\_fts: Instance of Functional Test Service object, will be assigned with the value of calling object.

UDF\_Wds: Instance of WebdomService object, will be assigned with the value of calling object.

UDF\_Ph: Instance of Playback Handler object, will be assigned with the value of calling object.

UDF\_Iwds: Instance of Internal Web DOM Service

Rownum: Integer variable, used to store the value of the calling row number of the Keyword Test Case.

**Enum:**

Functions: Stores the keyword names of the user-defined functions.

#### 5.1.1. Methods:

##### 5.1.1.1. CallUDF:

**Description:** CallUDF method calls the required user-defined function as per the Keyword Test Case.

**Visibility:** Public

**Parameters:**

FunName: String variable stores the name of the function.

FunPar: String variable stores the parameters of the functions, delimited with ':'.

**Function:**

1. Calls the required function as per the Keyword Test Case
2. If user-defined function returns a value, store the value in 'retval' variable.

**Exception Handling:** Throws exception to the calling function.

---

**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.*