Utility Objects in UFT

The **Utility Objects**section describes the global UFT One reserved objects. Reserved objects are not learned or stored in the object repository. Reserved objects enable you to retrieve or control UFT One setting or to modify UFT One behavior during a run session.

# Some of the frequently used Utility Objects

1. Crypt Object                       🡪 Encrypts/Decrypts the string
2. DataTable Object                🡪 All datatable operations
3. Description Object               🡪 We can do dynamic descriptive programming
4. Environment Object             🡪  We can create and access environment values
5. RandomNumber Object        🡪 Create random number
6. Recovery Object                   🡪 Recovery Scenario handling
7. Reporter Object                    🡪 Used to report the test results
8. TextUtil Object                      🡪 TextUtil is used to process text
9. XMLUtil Object                    🡪xmlUtil is used to process xml data.
10. WebUtil Object 🡪 Enables you to set or modify Web related settings.

## Crypt object

This object is used to encrypt strings.

Syntax:

**Crypt.Encrypt(**Data**)**

Its purpose is only to ensure that passwords are not be readily visible on the screen while recording, editing, or running a test or component.

## DataTable Object

Creates a runtime datatable

All methods performed on this object apply to the run-time data table object only. Changes to the run-time DataTable object are reflected in the run results, but the design-time data table is not affected.

Some of the methods available in datatable object:

|  |  |
| --- | --- |
| AddSheet | Adds the specified sheet to the run-time data table and returns the sheet so that you can directly set properties of the new sheet in the same statement. |
| DeleteSheet | Deletes the specified sheet from the run-time data table. |
| Export | Saves a copy of the run-time data table in the specified location. |
| ExportSheet | Exports a specified sheet of the run-time data table to the specified sheet in the specified file. |
| GetRowCount | Returns the total number of rows in the longest column in the first sheet in the run-time data table (global sheet). |
| GetSheet | Returns the specified sheet from the run-time data table. |
| Import | Imports the specified Microsoft Excel file to the run-time data table. |
| ImportSheet | Imports a sheet of a specified file to a specified sheet in the run-time data table. |

Example:

username = "john"  
password = "hp"  
  
DataTable.AddSheet("newsheet")  
DataTable.GetSheet("newsheet").AddParameter "UserName",username  
DataTable.GetSheet("newsheet").AddParameter "Password",password

## Environment Object

Enables you to work with environment variables.

You can set or retrieve the value of environment variables using the Environment object. You can retrieve the value of any environment variable. You can set the value of only user-defined, environment variables.

Environment.LoadFromFile "D:\Training\Mphasis Training\UFTOne\Env.xml"  
**msgbox** Environment("username")  
**msgbox** Environment("password")

## RandomNumber Object

Enables you to work with random number parameters.

You can generate a value for the specified random number parameter using the RandomNumber object.

Syntax:

**RandomNumber(***ParameterNameOrStartNumber*, [*EndNumber*]**)**

x=RandomNumber (0,100)

## TextUtil, WebUtil and XMLUtil Object

The object used to recognize text within a specified window handle.

MsgBox TextUtil.GetText(0, 20, 20, 200, 200)

Other alternative methods like GetRoProperty and text checkpoints are recommended than using a TextUtil

### WebUtil

**AddCookie**

WebUtil.AddCookie "www.dummy.com", "MyCookie=12345678; expires=Wed, 22-Dec-2001 00:00:00 GMT; domain=.dummy.com; path=/ ;secure"

**DeleteCookie**

WebUtil.DeleteCookie "www.dummy.com", "SecondCookie"

**DeleteCookies**

WebUtil.DeleteCookies

**GetCookies**

CookiesList = WebUtil.GetCookies "www.dummy.com"

**Launch Browser**

webutil.LaunchBrowser "CHROME",[www.google.com](http://www.google.com)

### XML Util

The object used to access and return XML objects.

Set XMLObj = XMLUtil.CreateXML()

XMLObj.LoadFile("C:\XML\BookStore.xml")

Set XMLObj = XMLUtil.CreateXMLFromFile("C:\XML\BookStore.xml")