
SSISTProbe – A Data Extraction Test Automation Framework

SSIS Test Automation Library Reference Manual

Filename: Library Reference Manual.doc
Revision: V1.01
Last Save Date: Sunday, March 02, 2008
Author(s): Sathyan S Nair
Author: satnair@thoughtworks.com (OR) Sathyan.s.nair@gmail.com
Read Only (public)
Link:

Table of Contents

SSISTProbe – A Data Extraction Test Automation	1
Framework.....	1
1. SSISTProbe – Automation Library Reference Manual.....	1
1.1. DBParent:: ConnectDB.....	1
1.2. DBParent:: ExecuteSql.....	1
1.3. UtilityParent:: GetConfigValue.....	2
1.4. XMLParent:: XMLDiff	2
1.5. XMLParent:: XMLDiff	2
1.6. DBParent:: DBXMLConv	3
1.7. UtilityParent:: ExecuteTestCases	3
1.8. UtilityParent:: GetSysUserName	3
1.9. DBParent:: CopyTable.....	4
1.10. DBParent:: CopyTableData	4
1.11. SSISParent:: SSISExecuteContainer.....	5
1.12. SSISParent:: SSISExecuteTask.....	5
1.13. SSISParent:: SSISTaskIDGen	5
1.14. SSISParent:: SSISExecutePackage.....	6
1.15. UtilityParent:: TestMethodLog	6
1.16. DBParent:: TableExist	6
1.17. DBParent:: GetRowCount.....	7
1.18. DBParent:: VerifyNull.....	7
1.19. DBParent:: VerifyIndex	7
1.20. DBParent:: GetDuplicates.....	8
1.21. UtilityParent:: TableDiff.....	8

1. SSISTProbe – Automation Library Reference Manual

Library design involves identifying requirements from multiple areas. At a high level, this includes (not limited to):

- Identification of necessary actions related to SSIS and Report Builder related to application functionalities
- Communication between the utilities/components (for example: data check-point components communicating to the logger etc)
- Customized user defined logs

1.1. DBParent:: ConnectDB

Prototype

```
SqlConnection ConnectDB(string serverName, string dbName, string userName, string password)
```

Description

Function to connect to Database and it returns SqlConnection object

Parameters

ServerName → Servername to connect to database
 DatabaseName → databasename to connect to database
 UserName → username to connect to database
 Password → password to connect to database

1.2 DBParent:: ExecuteSql

Prototype

```
ExecuteSql(string serverName, string dbName, string userName, string password, string queryFilePath, string functionName)
```

Description

Function to execute dml statements and .sql files. Returns # of rows affected.

Parameters

serverName → servername to connect to database
 dbName → databasename to connect to particular database
 userName → username to connect to database
 password → password to connect to database
 queryFilePath or QueryFile Name Path → sqlquery filepath or sqlquery statement
 functionName → function name to log if exception occurs

1.3 UtilityParent:: GetConfigValue

Prototype

```
GetConfigValue()
```

Description

Function reads config file and its returns collections of key and value for future reference

Parameters

No parameters

1.4 XMLParent:: XMLDiff

Prototype

```
XMLDiff(string sourceFile, string changedFile, string sourceNodeName, string  
changedNodeName, string diffFilePath)
```

Description

Function takes two xml files as input parameters, common node name of the two xml file and stores the difference in output XML file

Parameters

sourceFile → source xml file path
changedFile → changed xml file path
sourceNodeName → nodenames of sourcefilepath
changedNodeName → nodenames of changedfilepath
diffFilePath → difference to store in xmlfile

1.5 XMLParent:: XMLDiff

Prototype

```
XMLDiff(string sourceFile, string changedFile, string diffFile)
```

Description

Function takes two xml files as input parameters and store difference in a output xml file

Parameters

sourceFile → source xml file path
changedFile → changed xml file path
diffFile → XML file to store the diff

1.6 DBParent:: DBXMLConv

Prototype

```
DBXmlConv(string serverName, string dbName, string userName, string password, string
tableName, string fileSave)
```

Description

Function executes a SQL statement and stores the output in a XML file

Parameters

servername→ servername to connect to database
 dbName→ databasename to connect
 userName→username to connect to database
 password →password to connect to database
 tableName → tablename to get data
 fileSave→filepath to save tabledata information

1.7 UtilityParent:: ExecuteTestCases

Prototype

```
ExecuteTestCases(string filePath)
```

Description

Function takes XML testcase as input and executes all testmethods.

Parameters

filePath→ Testcase file path

1.8 UtilityParent:: GetSysUserName

Prototype

```
GetSysUserName()
```

Description

Function returns current user login

Parameters

No Parameter

1.9 DBParent:: CopyTable

Prototype

```
CopyTable(string srcServerName, string srcDatabaseName, string srcTableName, string
destServerName, string destDatabaseName,
          string destTableName, string functionName)
```

Description

Function takes two DB table as parameters. Creates target table in different DB including the data.

Parameters

srcServerName→source server name
srcDatabaseName→ source database name to connect
srcTableName→ source table name
destServerName→destination server name
destDatabaseName→destination database name
destTableName →destination table name
functionName→function name to log

1.10 DBParent:: CopyTableData

Prototype

```
CopyTableData(string srcServerName, string srcDatabaseName, string srcTableName, string
destServerName, string destDatabaseName, string destTableName)
```

Description

Function takes source and destination table details. Copies data from source to target.

Parameters

srcServerName→ source servername
srcDatabaseName→source databasename
srcTableName → source tablename
destServerName→destination servername
destDatabaseName→destination name
destTableName→destination table to copy all rows

1.11 SSISParent:: SSISExecuteContainer

Prototype

```
SSISExecuteContainer(string dtsFilePath, string xmlfilepath, string seqId)
```

Description

Function to execute a sequence container in a SSIS package

Parameters

dtsFilePath→SSIS File path to execute
xmlfilepath→xmlfile path it contains sequence id and seq name
seqId→sequence id to execute sequence

1.12 SSISParent:: SSISExecuteTask

Prototype

```
SSISExecuteTask(string dtsFilePath, string xmlfilepath, string TaskId, string functionName)
```

Description

Function to execute a SSIS task inside a sequence container

Parameters

dtsFilePath→SSIS File path to execute
xmlfilepath→xml file path it contains task id and task name
TaskId→ task id to excute
functionName→function name to log

1.13 SSISParent:: SSISTaskIDGen

Prototype

```
SSISTaskIDGen(string pathname, string outFileName)
```

Description

Function to generate a ID for each SSIS container and task

Parameters

pathname→ SSIS package file path
outFileName → xmlfilepath to save generate id

1.14 SSISParent:: SSISExecutePackage

Prototype

```
SSISExecutePackage(string dtsFilePath)
```

Description

Function to execute a .dtsx package

Parameters

dtsFilePath → SSIS package file path to execute

1.15 UtilityParent:: TestMethodLog

Prototype

```
TestMethodLog(string exceptMsg, string testCase, string filePath, int flag)
```

Description

Function to log system expectations and user-defined messages

Parameters

exceptMsg → exception message to save in logfile
 testCase → testcases name to save in logfile
 filePath → filpath to save logfile
 flag → exception is user or error msg

1.16 DBParent:: TableExist

Prototype

```
TableExist(string dbName, string serverName, string userName, string tableName, string  
functionName)
```

Description

Function to verify the table existence

Parameters

dbName → database name
 serverName → server name to connect
 userName → user name to connect to database
 tableName → tablename to verify table exist or truncate
 functionName → function name to log

1.17 DBParent:: GetRowCount

Prototype

```
GetRowCount(string serverName, string dbName, string tableName, string functionName)
```

Description

Function to fetch the rowcount information of a DB table

Parameters

serverName →servername to connect database
 dbName → database name to connect
 tableName → tablename to get row count
 functionName→ function name to log

1.18 DBParent:: VerifyNull

Prototype

```
VerifyNull(string serverName, string dbName, string tableName, string columnNames, string  
functionName, Hashtable configTable)
```

Description

Function to verify the value "NULL" existence for a particular column in a table

Parameters

serverName → servername to connect
 dbName → database name to connect
 tableName → tablename to verify null
 columnNames →columnnames to verify
 functionName →function name to log
 configTable → configtable contains filepath to store.

1.19 DBParent:: VerifyIndex

Prototype

```
VerifyIndexes(string serverName, string dbName, string tableName, string columnNames,
```

```
string functionName, Hashtable configTable)
```

Description

```
Function to verify DB table index existence
```

Parameters

```
serverName → servername to connect
dbName → database name
tableName →tablename to verify index
columnNames →columnnames verify column exist or not
functionName → function name to log
configTable → configtable contains filepath to store.
```

1.20 DBParent:: GetDuplicates

Prototype

```
GetDuplicates(string serverName, string dbName, string tableName, string columnNames,
string functionName, Hashtable configTable)
```

Description

```
Function to fetch the duplicate information for a specific columns in a table
```

Parameters

```
serverName → server name to connect database
dbName → database name
tableName → tablename to get duplicate column
columnNames →column names
functionName → function name to log
configTable → configtable contains file path to store
```

1.21 UtilityParent:: TableDiff

Prototype

```
TableDiff(string srcServerName, string srcDataBase, string srcUserName, string
srcPassword,
string srcTableName, string destServerName, string destDatabase,
string destUsername,
string destPassword, string destTableName, string outputFile, string
columnFlag, string functionName, Hashtable configTable)
```

Description

Function to compare two DB tables and log the difference to a output file

Parameters

srcServerName → source server name to connect
srcDataBase → source database name to connect
srcUserName → source database user name to connect
srcPassword → source server database password
srcTableName → source tablename to compare
destServerName → destination server name to connect database
destDatabase → destination database name to connect
destUsername → destination database user name to connect
destPassword → destination database password
destTableName → destination tablename to compare
outputFile → difference of comparsion table store in different file path

columnFlag → columnflag to make compare based on column
functionName → function name to log
configTable → config contain compare table exefilepath