SCTM Integration

# 1. Problem:

1. Executing python script: SCTM supports different types of scripts like VB script, JScripts, Perl, Python etc. But in executing python script, it totally depends on Windows Script Host (WSH) and it is not working as it is explained in the user guide. WSH generates error in executing the python script.
2. Logging Result: If a test suite/test fixture has many test cases, after executing the test suite, if some of tests passed and some of them failed, SCTM logs all as failed under the same test. It is true for any type of tests: NUnit, JUnit, Python and others.

# 2. Requirement of SCTM for Logging result:

SCTM requires that the test code generate an XML file (named output.xml) describing the result of the test execution. In case of NUnit or JUnit, the corresponding test framework creates the XML file, however both NUnit and JUnit create only one XMl file for all the test cases under the same test suite/Fixture. If there is any error in the result XML file, SCTM logs all the test cases under the suite as failed, not the individual test cases actually failed. In case of other scripting languages like perl or python, the test driver has to create the XML file. The format of the XML file is explained in the SCTM user guide.

# 3. Solution:

## 3.1 Test Driver/Suite/Fixture:

1. Create a temporary log file: The test driver will create a log file that contains information about the test cases within the script. Whether the test has passed or failed and any error information. For example, the following excerpt of a log file, indicates that the test case “Test Case1” has passed (1=passed) and the test case “Test Case44” has failed (0=failed) due to “Network not found”.

Test Case1 1

Test Case44 0

Error: Network not found

End

## 3.2 VBWrapper:

1. Execute the python script or any other exe for the test driver:

SCTM can execute VB Script (Please see the [sample](#_Sample_VB_Wrapper:) below). Call the python script or the exe of the test driver from the VB Script. For a particular test run, it will first look for the temporary log file. If it does not exist, it will execute the test script/test code or the exe which will create the log and it will execute the test script/test code only once as from the next time it will get the log file created in the first place.

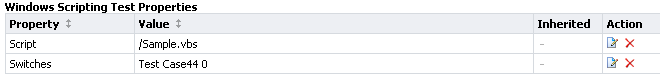
1. Parse the temporary log file created by the test driver

For each test case, parse the log file by calling the VB Script from SCTM (Please see SCTM [Configuration](#_3.3_SCTM_Configuration:) below). It is possible to identify the test case by specifying the test case name or ID in the Switch.

* 1. Create output.xml: Create the output.xml according to the requirement of SCTM containing the result of the test case got parsing the log.
  2. Log message: For error messages, log the message in the “Message” section of SCTM including them in the output.xml as above or using the following in the VB Script:

WScript.Echo “Network Not Found”

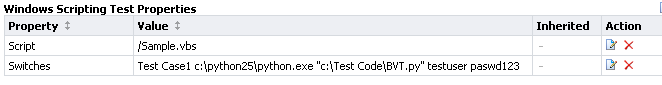
* 1. Delete the temporary log: When executing the last test case in the test suite, delete the log file. It is possible to identify the last test case passing a switch from SCTM (SCTM [Configuration](#_3.3_SCTM_Configuration:)) and interpreting that in the VB script. For example, during execution, SCTM will call the VB Script “Sample.vbs” and it will pass two parameters “Test Case44” – the name of the test case and “0” to indicate that it is the last test case.



## SCTM Configuration:

3.3.1 Test type:For each test case, choose the type as “Windows Scripting Test”

3.3.2 Specify the script and switches**:** Specify the test script and the necessary switches which will be passed as command line arguments in the VB script. For example, the following is the configuration for the test case “Test Case1”. It will call the VB Script “Sample.vbs” and will pass five command line parameters for the script: “Test Case1” the name of the test case (which will be used to parse the log to get the result), path of the python interpreter, python script including the path which contains the test cases (python script must exist in that path in the SCTM execution server), user name and password, the required parameters to execute the test script.

****

# Sample VB Wrapper:

|  |
| --- |
| dim outFile, errorCount, warningCount  errorCount = 0  Set oShell = CreateObject("Wscript.Shell")  Set WshSysEnv = oShell.Environment("System")  'Create a File System Object  Set FSO = CreateObject("Scripting.FileSystemObject")  'Name of the temporary file including path which will be created by the Test code  fileName = "c:\temp\Borland\BVTLog.txt"  /\*  Test Script : BVT.py  Location of the script in execution server: c:\Test Code  Required arguments for this Script: username and password  Path of the python interpreter in execution server: c:\python25\python.exe  Command to execute the above test code from commandline:  c:\python25\python.exe "c:\Test Code\BVT.py" <username> <password>"  To execute the above code, user has to pass five parameters through the switch of SCTM:  0. Test case identifier (say, name of the test case)  1. pythonPath (c:\python25\python.exe)  2. path of the test code ("c:\Test Code\BVT.py")  3. username  4. password    \*/  args = WSCript.Arguments.Count  /\* Get the required parameters to construct the command to execute the test code  Parameters are passed through the switch of SCTM  \*/  if (args = 5) Then  pythonPath = WScript.Arguments.Item(1)  WSCript.Echo pythonPath  pythonScript = WScript.Arguments.Item(2)  WSCript.Echo pythonScript  username = WScript.Arguments.Item(3)  WSCript.Echo username  password = WScript.Arguments.Item(4)  WSCript.Echo password  end if  /\*  If the log file does not exist, execute the test code that will create the log file.  \*/  if (Not FSO.fileExists(fileName)) Then  /\*Execute the test code which will create the temporary log file containing the test  results \*/  WSCript.Echo "Executing the test code"  /\* Create the command to execute the test code combining the arguments  For example: c:\python25\python.exe "c:\Test Code\BVT.py" <UserName> <Password>  Execute the command using the VB Script Run command  execute the command synchronously passing True as an argument to VB Script Run  command \*/    oShell.Run  pythonPath&Chr(32)&Chr(34)&pythonScript&Chr(34)&Chr(32)&username&Chr(32)&password, ,True  end if  If args < 1 then  WScript.Echo "usage: args.vbs argument [argument] [argument] [argument]"  WScript.Quit  end If  outFile = "output.xml"  warningCount = 0  'Get the test case identifier passed from the SCTM through switch  t1 = WScript.Arguments.Item(0)  ' Extract the result of test execution parsing the log file  if (FSO.fileExists(fileName)) Then  'Open the log file  Set oFile = FSO.OpenTextFile(fileName, 1)  'Parse for result of the test case under execution  Do While Not oFile.AtEndofStream  str = oFile.ReadLine  a = Split(str)    if (StrComp(a(0), t1) = 0) Then  'Check whether the test case has been failed  if (StrComp(a(1), "0") = 0) Then  errorCount = 1    'Extract the error messages when failed.  Do While Not oFile.AtEndofStream  'Read error message line  msg = oFile.ReadLine  'Check if reached the End of the error message  if (StrComp(msg, "End" ) = 0) Then  Exit Do  end if  'Write the error message in the Message section of SCTM  WSCript.Echo msg  Loop  end if  /\* End parsing the log file once the result for the test case under execution  has been found.  \*/  Exit Do    end if    Loop  'Close the log file  oFile.Close  end if  /\*Create the output.xml file according to requirement of SCTM embeding the results got as  above \*/  Set oTX = FSO.OpenTextFile(outFile, 2, True, -1)  oTX.WriteLine ("<?xml version=""1.0"" encoding=""UTF-16""?>")  oTX.WriteLine ("<ResultElement TestItem=""pythontest"">")  oTX.WriteLine (" <ErrorCount>" & errorCount & "</ErrorCount>")  oTX.WriteLine (" <WarningCount>" & warningCount & "</WarningCount>")  oTX.WriteLine (" <RunCount></RunCount>")  oTX.WriteLine ("<Incident>")  oTX.WriteLine ("<Message></Message>")  oTX.WriteLine (" <Severity>Error</Severity>")  oTX.WriteLine (" <Detail>")  oTX.WriteLine ("<TestName>pythontest</TestName>")  oTX.WriteLine (" <Info>Automation from VBS Script</Info>")  oTX.WriteLine (" </Detail>")  oTX.WriteLine (" </Incident>")  oTX.WriteLine ("</ResultElement>")  oTx.Close  /\*  The last test case in the test suite under execution will have an argument "0" to mark  the test case as the last one.  \*/  if (args = 2) Then  t2 = WScript.Arguments.Item(1)  'Delete the log file if it is the last test case  if (StrComp(t2, "0") = 0) Then  Set textFile = FSO.GetFile(fileName)  textFile.Delete  WSCript.Echo "Deleting temp log"  end if  end if |