Running Regression Tests From Command Line

You can run regression test from command line. This is needed, if you want to e.g. integrate the regression tests into a continuous integration server such as Jenkins.

Call the Regression Test Command Line Tool like:

```
java -jar RegTestRunner.jar -project <path to the Builder project folder>
[-suite <name of the test suite to be executed>] [-logfile <filename
including path>] [<target Bridge>]
```

e.g. **java -jar RegTestRunner.jar -project "C:\E2E Documentation\Advanced Modeling\PState" -suite "Build" -logfile C:\Temp\testlog.xml**.

Example File (Builder project Advanced Modeling/PState):

<your example path>\Advanced Modeling\PState\uml\pstatePurchaseOrder.xml

Related Pages:

- Regression Test File and Folder Structure
- Setting a Request Timeout for the Analyzer

Parameters

Regression Test Parameters

To select the tests to perform, you can specify the following parameters:

Parameter	Mandatory	Description	Example
-project	✔	Specify the path to the Builder project that contains the regression tests.	`-project "C:\E2E Documentation\Advanced Modeling\PState"`
-suite		Specify the test suite you want to run. The test suite must exist within the project given with `-project`. If you specify no test suite, all tests of the project will be executed.	`-suite "QA Tests/PurchaseOrderExample"`
-file		Deprecated since Analyzer 6.0.27Specify the path to a testsuite.xml that contains the tests you want to perform. For more information on where to find such a file, refer to Regression Test File and Folder Structure.	`-file "C:\E2E Documentation\Advanced Modeling\PState\regressiontest\QA Tests\testsuite.xml"`
-list		Lists all available test suites in the project specified by `-project`. No tests executed.	`-project "C:\E2E Documentation\Advanced Modeling\PState" -list`
-logfile		Optionally specify a filename (including extension of your choice) and a path. The RegTestRunner will generate a file with the given name to the given path. If you specify a filename only, the logfile will be generated to the location of the test suite. This logfile is not to be confused with the logfile that is created if you have added a logfile option in the Regression Test Tool (see Logging Test Runs). For more information on the contents of the logfile, see Format of the Test Logfile further below.	`-logfile C:\Temp\myLogFile.txt`

Target Bridge Parameters

Optionally specify a target Bridge to run the tests against. Specify all or none of the following parameters:

Parameter	Description	Example
-host	Host name of the Bridge. Used for SOAP, deploy, start and stop tests.	`-host testbridge.e2e.ch`
-port	Bridge port. Used for deploy, start and stop tests.	`-port 8080`
-username	Bridge user to authenticate for deploy, start and stop tests.	`-username admin`
-password	Password of the Bridge user. Used for deploy, start and stop tests.	`-password secure1234`
-node	Bridge node name. Used for deploy, start and stop tests. If it is the same as **-host**, you can skip this parameter.	`-node localhost`

> Note, that the target Bridge parameter values will only be applied to the tests, if the tests do not have these values specified directly in the test themselves. Tests that have e.g. specified a dedicated host in the test case properties will not get overwritten by the target Bridge parameter values.

To make your tests compliant to using the target Bridge parameters, you have to specify the related data (host, port, user credentials and node) via test case options. How to do that is explained in detail on Adding Options to a Test Suite pp.

Setting a Request Timeout

Analyzer 7.2.0 Regression test actions that do a network request have a timeout. You can change the timeout defaults via Java system properties:

Action	Default Timeout	Java System Property	Value
Establish connection	1 minute	`sun.net.client.defaultConnectTimeout`	Specify a value in milliseconds. A value of **-1** means: no timeout.
Wait for response	10 minutes	`sun.net.client.defaultReadTimeout`	Specify a value in milliseconds. A value of **-1** means: no timeout.

The following test actions do support timeouts:

- Deploy Bridge Repository (Deploy Test)
- Start Bridge Service
- Stop Bridge Service
- Send a Request to the Bridge

Set the changed timeouts via the Regression Test Command Line Tool call like

```
java -Dsun.net.client.defaultReadTimeout=5000 -jar RegTestRunner.jar ...
```

The command above starts tests with a timeout of 5000 ms for waiting for a response of network requests.

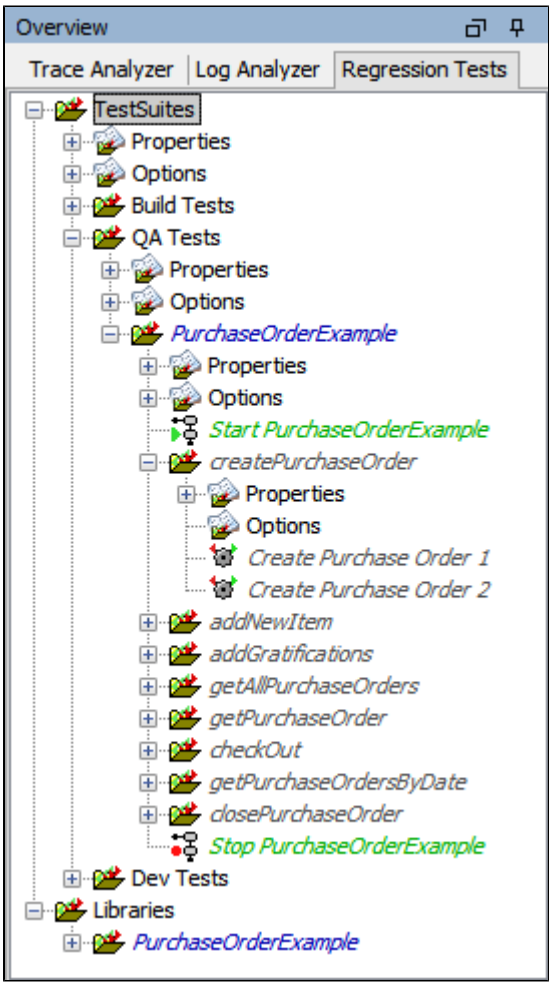
To set the same timeouts for the Analyzer refer to Setting a Request Timeout for the Analyzer.

Examples

To try out the Regression Test Command Line Tool, you can use the PState Builder project that is delivered with the E2E Examples.

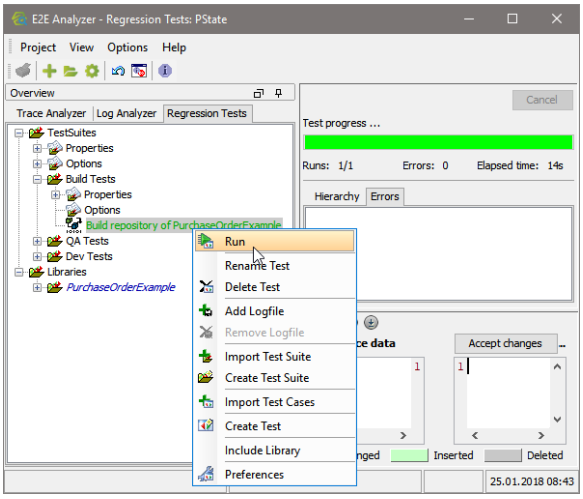
Example File (Builder project Advanced Modeling/PState):

	<your example path>\Advanced Modeling\PState\uml\pstatePurchaseOrder.xml

Analyzer	Regression Test Command Line Tool
	C:\E2E Documentation\Advanced Modeling>java -jar RegTestRunner.jar -project PState -list /Build Tests /QA Tests /QA Tests /PurchaseOrderExample /QA Tests /PurchaseOrderExample /createPurchaseOrder /QA Tests /PurchaseOrderExample /addNewItem /QA Tests /PurchaseOrderExample /addGratifications /QA Tests /PurchaseOrderExample /getAllPurchaseOrders /QA Tests /PurchaseOrderExample /getPurchaseOrder /QA Tests /PurchaseOrderExample

/checkOut
/QA Tests
/Purchase
OrderExam
ple
/getPurch
aseOrders
ByDate
/QA Tests
/Purchase
OrderExam
ple
/closePur
chaseOrde
r
/Dev
Tests
/Dev
Tests
/Purchase
OrderExam
ple
/Dev
Tests
/Purchase
OrderExam
ple
/createPu
rchaseOrd
er
/Dev
Tests
/Purchase
OrderExam
ple
/addNewIt
em
/Dev
Tests
/Purchase
OrderExam
ple
/addGrati
fications
/Dev
Tests
/Purchase
OrderExam
ple
/getAllPu
rchaseOrd
ers
/Dev
Tests
/Purchase
OrderExam
ple
/getPurch
aseOrder
/Dev
Tests
/Purchase
OrderExam
ple
/checkOut
/Dev
Tests
/Purchase
OrderExam
ple
/getPurch

	aseOrders ByDate /Dev Tests /Purchase OrderExam ple /closePur chaseOrder



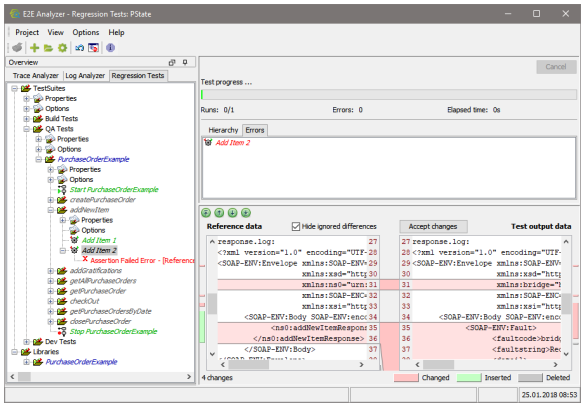

```
C:\E2E Documentation\Advanced Modeling>java -jar RegTestRunner.jar -project PState -suite "Build Tests"
Running Test '<buildRepositoryTest>Build repository of PurchaseOrderExample'.
Finished: 19.094 seconds

<testsuites>

<testsuite name="TestSuites.Build Tests" tests="1" >

<testcase errors="0" name="Build repository of PurchaseOrderExample" time="19.091"/>
   <
/testsuite>
<
/testsuites>
```

```
C:\E2E Documentation\Advanced Modeling>java -jar RegTestRunner.jar -project
```



PState -
suite
"QA Tests
/Purchase
OrderExam
ple"
Running
Test
'<startse
rvice>Sta
rt
PurchaseO
rderExamp
le'.
Running
Test
'<Testcas
eTest>Cre
ate
Purchase
Order 1'.
Running
Test
'<Testcas
eTest>Cre
ate
Purchase
Order 2'.
Running
Test
'<Testcas
eTest>Add
Item 1'.
Running
Test
'<Testcas
eTest>Add
Item 2'.
Running
Test
'<Testcas
eTest>Add
Gratifica
tions
for
Wishes
Unlimited
'.
Running
Test
'<Testcas
eTest>get
AllPurcha
seOrders'
.
Running
Test
'<Testcas
eTest>Get
Purchase
Order 1'.
Running
Test
'<Testcas
eTest>Get
Purchase
Order 4'.
Running
Test
'<Testcas
eTest>Che

ck Out
Order 1'.
Running
Test
'<Testcas
eTest>Che
ck Out
Order 2'.
Running
Test
'<Testcas
eTest>Get
Purchase
Orders
from
1970-11-
01'.
Running
Test
'<Testcas
eTest>Clo
se
Purchase
Order 1'.
Running
Test
'<Testcas
eTest>Clo
se
Purchase
Order 2'.
Running
Test
'<stopser
vice>Stop
PurchaseO
rderExamp
le'.
Finished:
8.261
seconds
<testsuit
es>

<testsuit
e name="
TestSuite
s.QA
Tests.
PurchaseO
rderExamp
le"
tests="
15">

<testcase
errors="
0" name="
Start
PurchaseO
rderExamp
le"
time="
1.198"/>

<testcase
errors="
0" name="
Stop
PurchaseO
rderExamp

```
le"
time="
5.694"/>
   <
/testsuit
e>

<testsuit
e name="
TestSuite
s.QA
Tests.
PurchaseO
rderExamp
le.
createPur
chaseOrde
r"
tests="2"
>

<testcase
errors="
0" name="
Create
Purchase
Order 1"
time="
0.316"/>

<testcase
errors="
0" name="
Create
Purchase
Order 2"
time="
0.116"/>
   <
/testsuit
e>

<testsuit
e name="
TestSuite
s.QA
Tests.
PurchaseO
rderExamp
le.
addNewIte
m"
tests="2"
>

<testcase
errors="
0" name="
Add Item
1" time="
0.119"/>

<testcase
errors="
1" name="
Add Item
2" time="
0.111">

<error
message="
```

Assertion Failed Error - [Reference = D:\E2E Documentation 18\Advanced Modeling\PState\regressiontest\reference\libraries\PurchaseOrderExample\addNewItem\PurchaseOrderService.PurchaseOrderPort.addNewItem\AddItem2\request.log]

 [different] Expected text value '1' but was '99' - comparing <purchaseOrderID ...>1</purchaseOrderID> at /env:Envelope[1]/env:Body[1]/ns0:addNewItem[1]/purchaseOrderID[1]/text()[1] to <purchaseOrderID ...>99</purchaseOrderID> at /env:Envelope[1]/env:

```
Body[1]
/ns0:
addNewIte
m[1]
/purchase
OrderID
[1]
/text()
[1]&#xA;
&#xA;
[Referenc
e = D:
\E2E
Documenta
tion
18\Advanc
ed
Modeling\
PState\re
gressiont
est\refer
ence\libr
aries\Pur
chaseOrde
rExample\
addNewIte
m\Purchas
eOrderSer
vice.
PurchaseO
rderPort.
addNewIte
m\Add
Item
2\respons
e.log]
&#xA;
&#xA;
[differen
t]
Expected
attribute
name
'xmlns:
ns0' but
was
'null' -
comparing

   [...]
<
/testsuit
es>
```

Format of the Test Logfile

The logfile generated by the `-logfile` parameter is formatted in a junit compatible xml format that is used by Jenkins and others:

```
<testsuites>
   <testsuite name="TestSuites" tests="10"/>
   <testsuite name="TestSuites.Dev Tests" tests="10"/>
      <testsuite name="TestSuites.Dev Tests.PurchaseOrderExample" tests="
10">
         <testcase errors="0" name="Create Purchase Order 1" time="0.854"/>
         <testcase errors="1" name="Create Purchase Order 2" time="0.01">
            <error message="Assertion Failed Error - [...]"/>
         </testcase>
         [...]
   </testsuite>
</testsuites>
```

The XML structure of the logfile reflects the test case structure in the Regression Test Tool and gives some additional information on the test execution:

XML Element	XML Attribute	Description
testsuites	-	Root element.
testsuite	name	Name of the test suite including test suite path in a test suite hierarchy.
	tests	Number of tests in the test suite.
testcase	name	Name of the test case.
	errors	Number of errors during test execution.
	time	Elapsed test time in seconds.
error	message	Text of the test error message.

Troubleshooting: Bridge Certificate Exception

Problem

Running tests with the Regression Test Command Line Tool, Bridge calls fail and you are getting this error: `javax.net.ssl.SSLHandshakeException: java.security.cert.CertificateException: Certificates does not conform to algorithm constraints`.

Solution

Check the certificate on your Bridge (see Managing the Bridge Certificate). MD5 certificates are deprecated as insecure. If your Bridge still uses MD algorithm (**MD5withRSA**), you should create a new certificate and restart the Bridge. The new certificate should have algorithm **SHA256withRSA**.

Workaround:
If you can not fix the Bridge as described above (for whatever reason), you can change file `<jre>/lib/security/java.security` of the JRE you use to run the Regression Test Command Line Tool. Remove the "MD5" parts in **jdk.certpath.disabledAlgorithms** and **jdk.tls.disabledAlgorithms** to enable the usage of MD5 certificates.