1. ANNEX B: SOAPUI DOCUMENTATION

1.1. Abbreviations

IN: Initialization

JRE: Java Runtime Environment

MS: Message

SC: Scenario

TC: Test Case

UC: Use Case

1.2. Test tool setup

1.2.1. Program

- Download and install from the web site the latest free version of soapUI (http://www.soapui.org/)

1.2.2. Add-ons and plug-ins

Additional resources that are needed to work with soapUI can be found and taken from the OSOR repository:

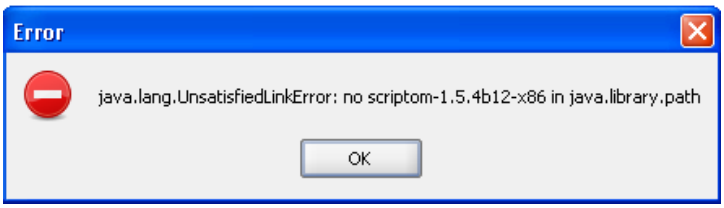
https://svn.forge.osor.eu/svn/openeprior/trunk/Test/Test%20Environment/soapUI/resources/

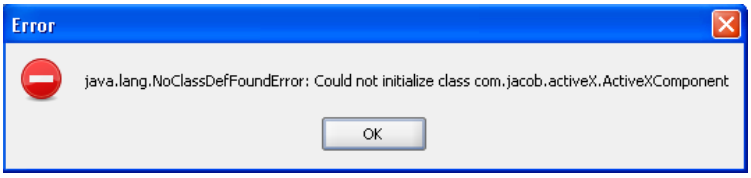
- Copy all .jar files in the additional resources folder to $SOAP_UI_HOME/bin/ext
- jacob-1.14-x86.dll or jacob-1.14-x64.dll - Copy the file that matches your processor type to a folder included in your path (for example C:\WINDOWS\system32)
- scriptom-1.5.4b11-x86.dll or scriptom-1.5.4b11-x64.dll - Copy the file that matches your processor type to a folder included in your PATH (for example C:\WINDOWS\system32)
- Default location for $SOAP_UI_HOME is C:\Program Files\ eviware\soapUI-3.0.1

Important Notes

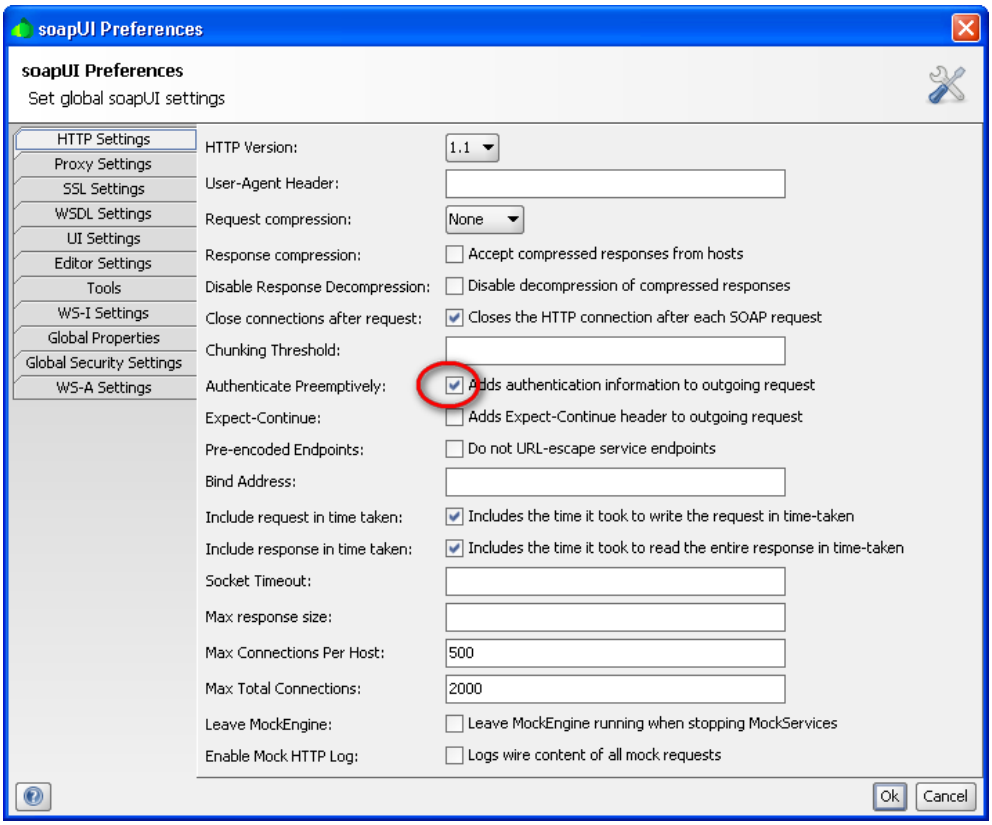
- Due to the attachments used for the some of the projects, e.g. UC7 Submit Attached Document, soapUI needs sufficient memory to function successfully. You should modify the file soapui.bat in order to increase the program's memory:
  - go to C:\Program Files\eviware\soapUI-3.0.1\bin
  - right click to edit the bat file; you will find the following line where you should change the memory to be at least -Xms128m -Xmx512m
  - set JAVA_OPTS=%JAVA_OPTS% -Xms128m –Xmx1024m -Dsoapui.properties=soapui.properties
- You should run the application by using the soapui.bat file, and also change the shortcut to point to the bat file

- In case you come up with one of the following error messages while running tests which contain pop up windows (e.g. Delete_entire_database, or during the special test case scenarios §1.7), please remove from the soapui.bat file the sentence `set JAVA_OPTS=%JAVA_OPTS% -Djava.library.path="%SOAPUI_HOME%\"`





- Run soapUI and go to File\Preferences; you should tick the box "Authenticate Preemptively"



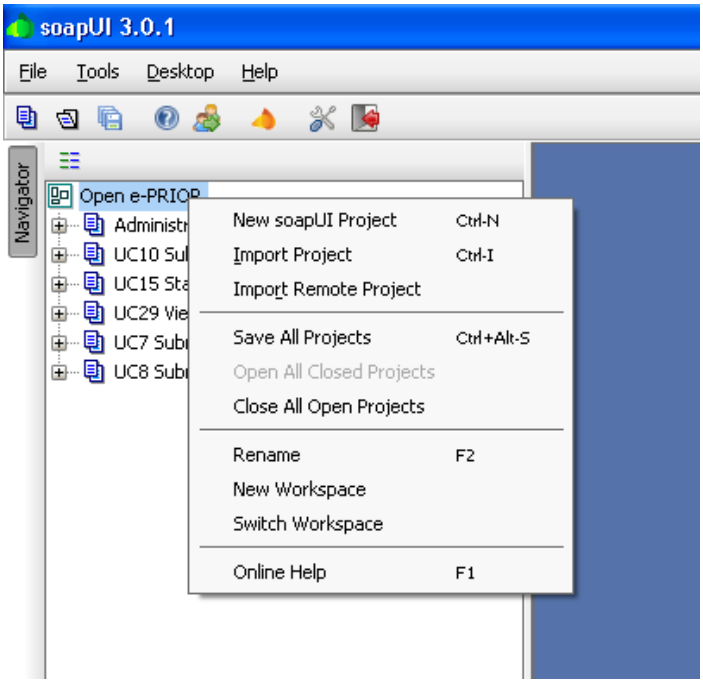
Once installation of all the software components is finished, you can refer to official soapUI documentations to start with:

- Getting started: http://www.soapui.org/gettingstarted/index.html
- User guide: http://www.soapui.org/userguide/index.html (also available in pdf format)

1.2.3. Create or import a project

To create a new project you should:

- Right click on the workspace node in the left navigation pane and select "New soapUI Project" (see picture below)
- You will be prompted for a project name and then for a local file where the project should be saved
- If all is OK, an empty project will be created in the workspace



To import an existing project you should:

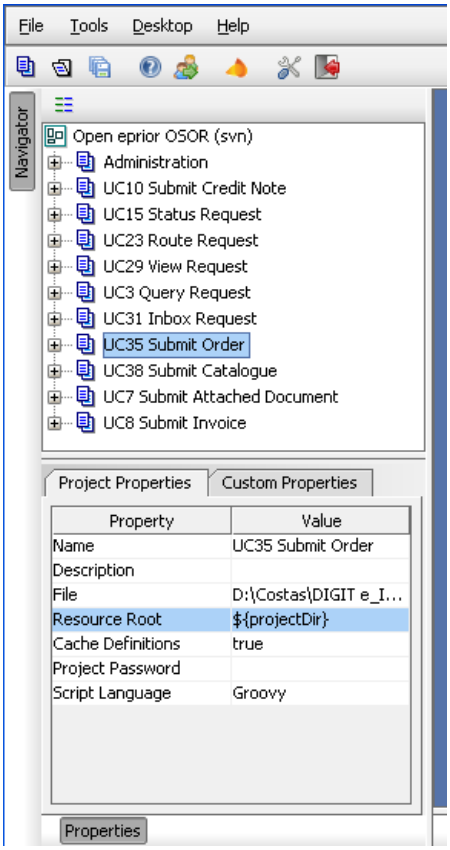
- Right click on the workspace node in the left navigation pane and select "Import Project" (see picture above)
- You will be prompted to select a soapUI project file
- If all is OK, the project will be imported in the workspace

The projects you can import in soapUI for Open e-PRIOR, can be found in the OSOR repository:

https://svn.forge.osor.eu/svn/openeprior/trunk/Test/Test%20Environment/soapUI/OSOR/

As soon as you prepare the environment, you should add to the following projects the value "${projectDir}" in "Resource Root":

- UC3 Query Request
- UC15 Submit Request
- UC23 Route Request
- UC29 View Request
- UC31 Inbox Request
- UC35 Submit Order
- UC38 Submit Catalogue

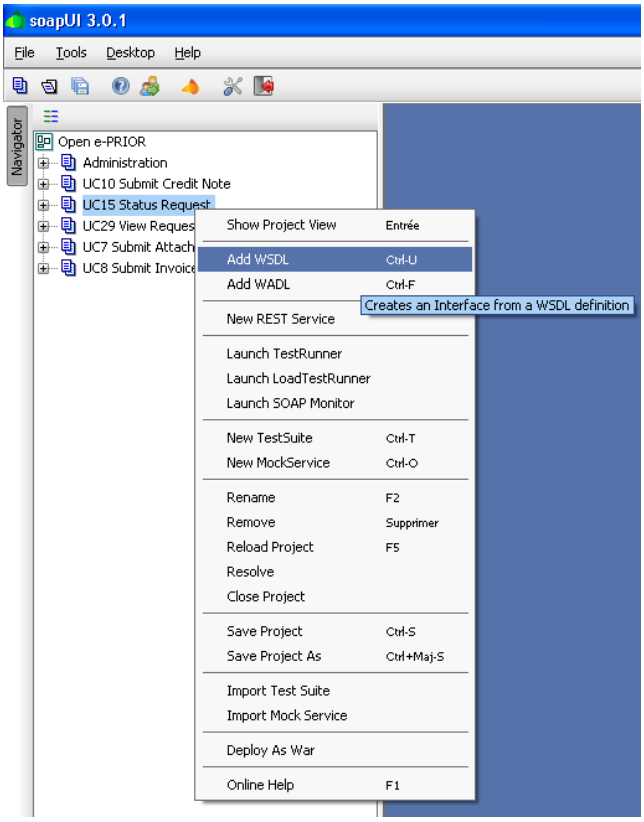


1.2.4. Adding a WSDL to a project

Once a project is created or imported, you can start adding interfaces to it:

- Right click on the project node and select "Add WSDL from URL" (see picture below)
- You will be prompted for the URL to the Interfaces WSDL; there you enter the WSDL for each of the above projects, for example: http://localhost:8080/OpenEPriorIntegration/wsdl/Invoice-0.1.wsdl (UC8 Submit Invoice)

- soapUI will prompt you if you want to create default requests for each operation, simply select "Yes"
- soapUI will now add the SOAP/Http Binding for the invoice



Once the projects and the bindings for each UC are added, the test suites can be created.

1.3. Test Design

1.3.1. Naming rules

Each UC contains test suites which were named after the use case, the scenario and the test case. For example UC7_SC8, is for the project UC7 Submit Attached Document, the 8th scenario.

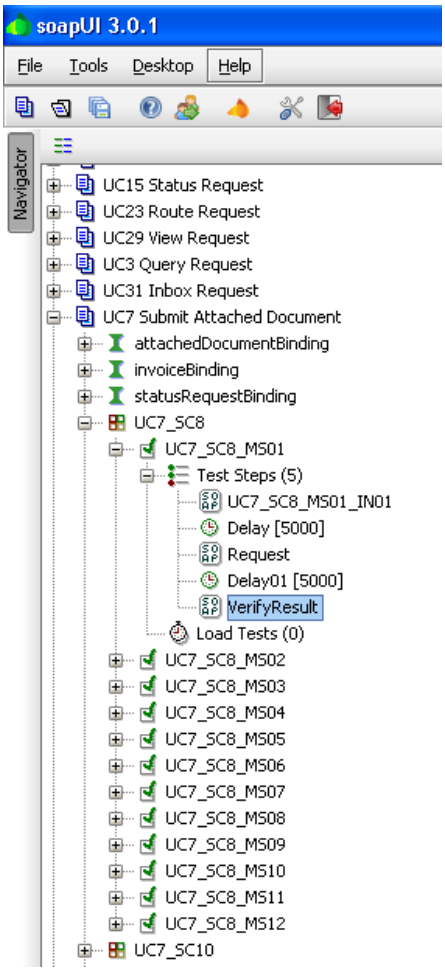
- UC*_SC*; test suite which contains a soapUI test case for each separate test message defined in the test case
- UC*_SC*_MS*; soapUI test case to test a specific test message defined in the corresponding excel test case
- UC*_SC*_MS*_IN*; message that will be sent to initialize the system in order to be able to send the main test request
- Request; main test message of each test case

1.3.2. Test case structure

A test case consists out of the following items:

- Imported project in soapUI (e.g. UC7 Submit Attached Document)
- Test suite (e.g. UC7_SC8)
- Test message (e.g. UC7_SC8_MS01)
- Test body:
    - initialisation message; where applicable (e.g. UC7_SC8_MS01_IN01)
    - delay of ~6000ms between steps to let the system process
    - another initialisation message; where applicable (e.g. UC7_SC8_MS01_IN02)
    - delay of ~6000ms between steps to let the system process; where applicable
    - request; the main test message for all the TCs
    - another request; where applicable
    - Verify Result; always appeals to the "Request" step of each TC, i.e. the invoice. It is a status request which verifies the objectives of the test case.

The following picture shows the complete structure of a test case.



Test case example

UC7 Submit Attached Document is used as reference.

Scenario 8

Description: Verify the Basic Flow of the Submit Attachment Use Case

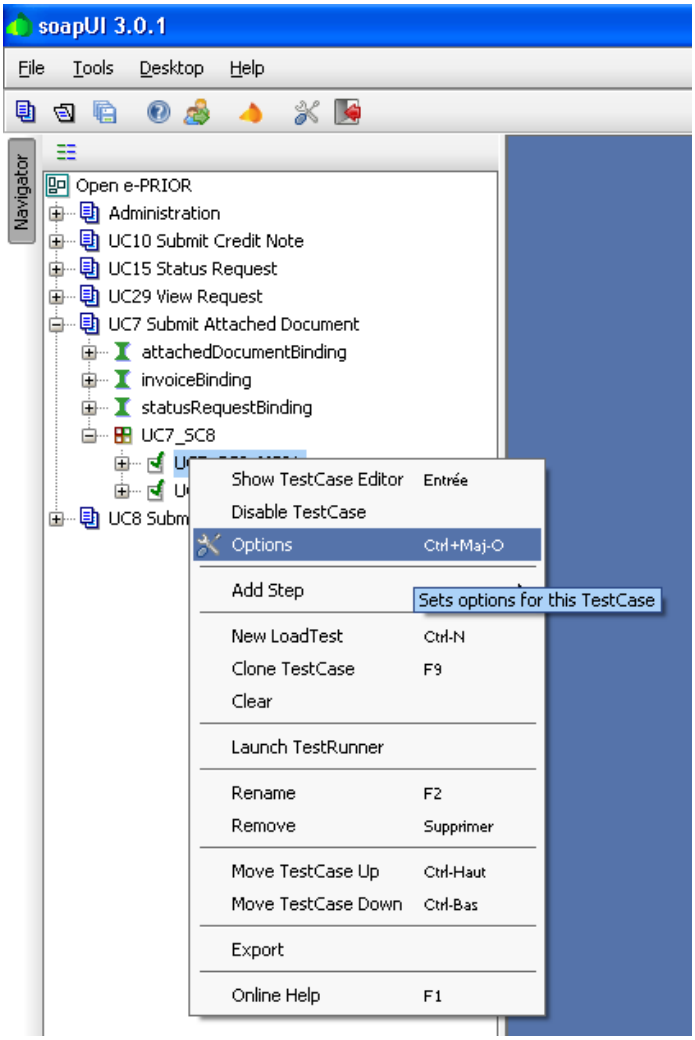
Test as it appears in soapUI:

- UC7_SC8; test suite
- UC7_SC8_MS01; test case
- UC7_SC8_MS01_IN01; the user submits a valid invoice to the system
- Request; the user submits a valid attachment to the system
- VerifyResult; the user submits the status request for the invoice and any related document

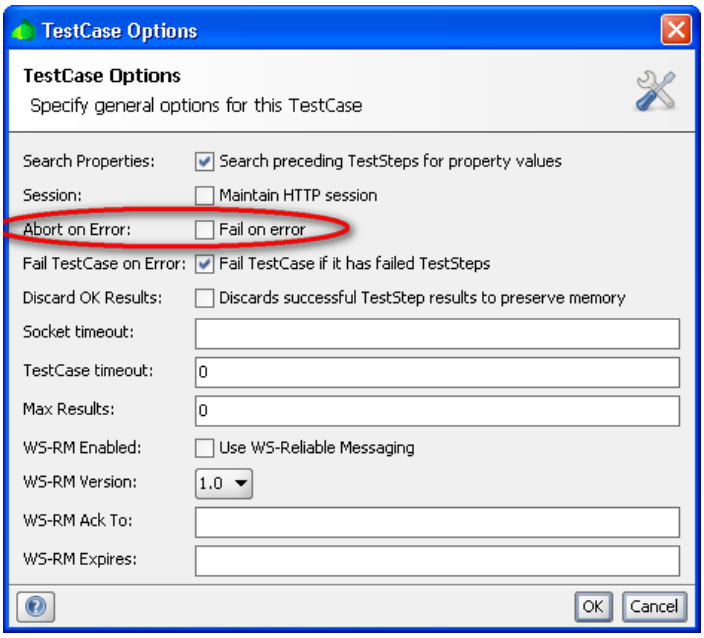
Outcome: All assertions (refer to §1.3.5) should be OK, indicated by a green light for the TCs

1.3.3. Test case options

Once you created a test case you should right click on it and select "Options".

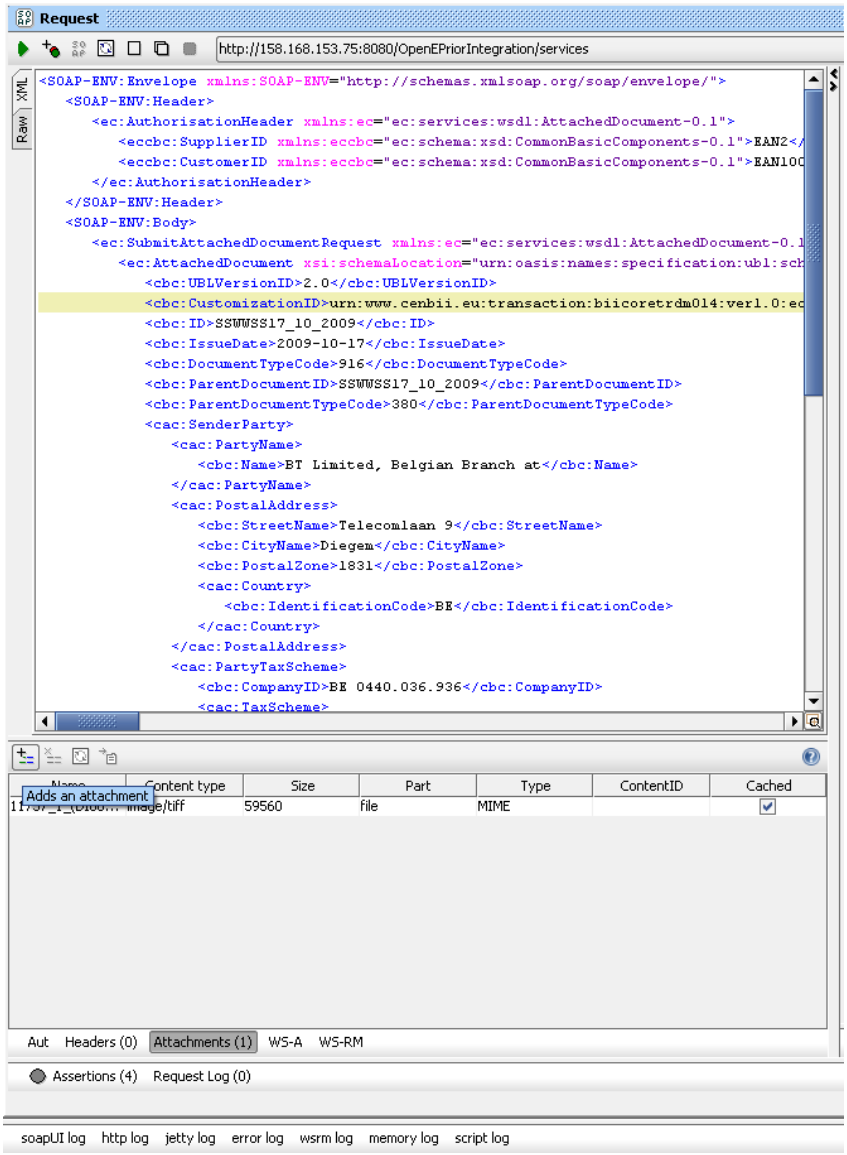


The "TestCase Options" window will be prompted, where you should un-tick the option "Abort in Error". Then, the "Fail TestCase on Error" option will automatically become active.



1.3.4. Attachments

To add an attachment to a request message, select the attachments button ("Adds an attachment to this request") at the bottom of each message editor

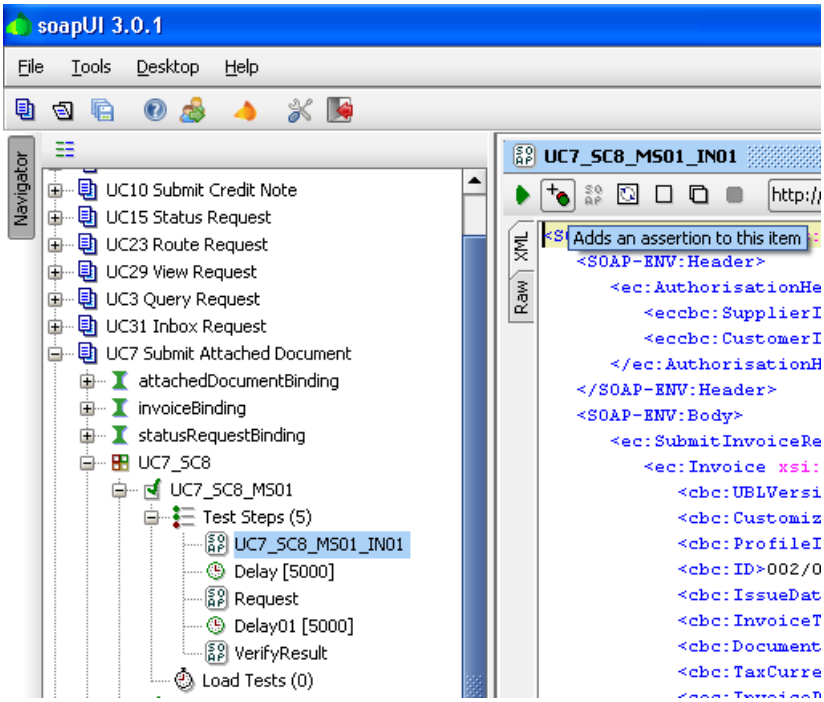


Once attached, you should modify some of its settings from the attachment table:

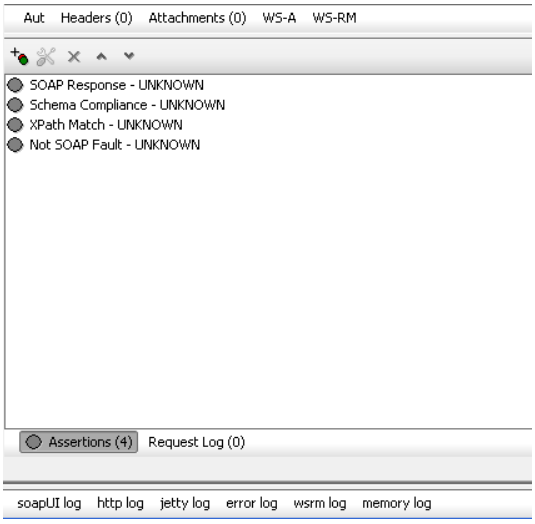
- "Part"; the part this attachment should be associated with. Scroll down and choose "file"
- "Content type"; the MIME content-type of the attachment. You should rename it according to the type you are using, for example "image/tiff" (refer to §1.6 Valid Attachments Types).

1.3.5. Assertions

For each test request you should add some assertions to verify that the response from the system is in line with the predefined test criteria. Select the second toolbar button in the request editor window ("Adds an assertion to this item") and add:



- "Schema Compliance" (*) assertion; this will check that the response is compliant with the associated WSDL/Schema definition. The assertion will be shown in the assertion list under the request/response editors (see image below)



- "SOAP Response" (*) assertion; this will validate that the response is a valid SOAP Message. This is the minimal assertion that should be added to catch empty responses or HTTP error pages. This assertion has no configuration parameters.
- "XPath Match" assertion; this will allow specification of an XPath expression to be evaluated against the received response message and compare its result to a predefined value.

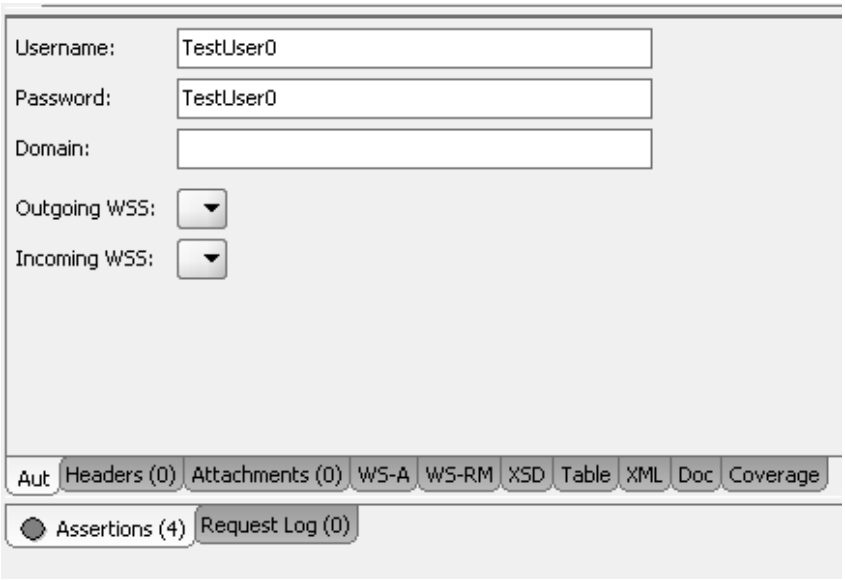
- "Not SOAP Fault" (*) assertion; checks that the response is not a SOAP Fault. This should be used in conjunction with the Schema Compliance since a SOAP Fault does not get validated against any schema (unless there is a Fault Part defined in the WSDL and that Fault Part is present in the response). This assertion has no configuration parameters.

Assertions marked with a (*) are "singular assertions", meaning that they can be added only once to a TestRequest.

Post the request with the green submit-button, soapUI will run the request and validate the response. If all goes well, the test-request should be shown with a green background in the navigation tree.

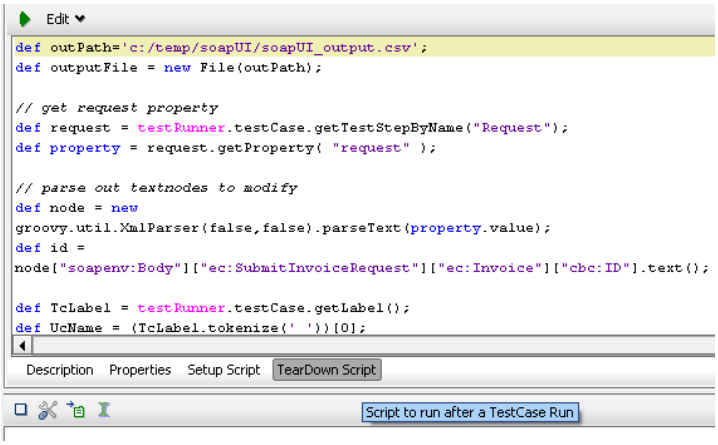
1.3.6. Authentication

To add an authentication to a request message, select the Aut button ("Authentication and Security-related settings") at the bottom of each message editor and add a username and password.



1.3.7. TearDown script

A TearDown script is a groovy script used to run automatically when a test suite finishes. Select the button TearDown Script at the bottom of each test case.



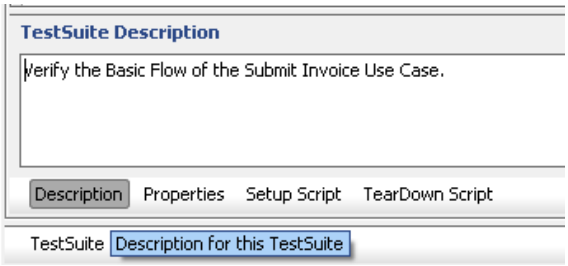
The TearDown script is currently used to record data in the comma separated values (CSV) files soapUI_output.csv:

- Message Name
- UC Name
- SC Name
- Message ID
- Date
- Start Time
- End Time
- Status
- Reason

(Refer to § 1.4 Administration create folders)

1.3.8. Descriptions

Select the button Description at the bottom of each test suite or test case. A test suite description of the "Test Objective" according to the TC (e.g. the description of UC8_SC1 in SoapUI should contain "Verify the Basic Flow of the Submit Invoice Use Case.") and the test case description contains information about the specific messages in that TC (e.g. the description of UC8_SC1_MS01 in SoapUI should contain "UC8_SC1_MS01.xml: valid file").

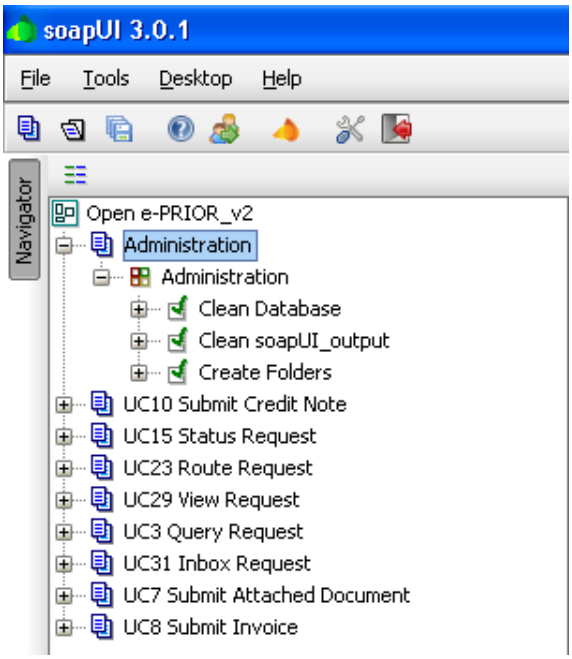


1.4. Administration

The Administration project contains Groovy scripts that can be executed upon request of the tester:

- Clean Database; removes all test messages from Open e-PRIOR's database to restore it to a clean state before initiating tests
    - Please note that ID or parameters used inside of the SQL statements might need to be change if you do not test with the default users from the set up scripts
    - Please note that to clean the Catalogue Tables from the database, the following script needs to be executed: <<Distribution-Package>>/scripts/OpenePRIOR/CentralCatalogue/hsqlRunDeleteScript.bat

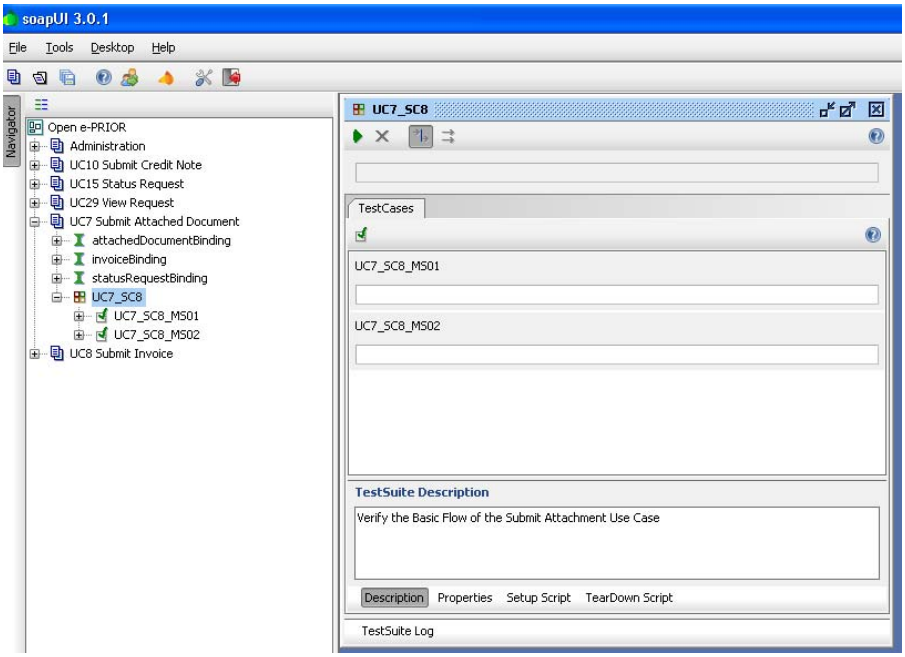
- Create Folders; it will create C:\temp\soapUI where the CSV files will be generated.
- Clean soapUI_output; removes all soapUI output data recorded in the excel file soapUI_output.csv.



1.5. Test execution

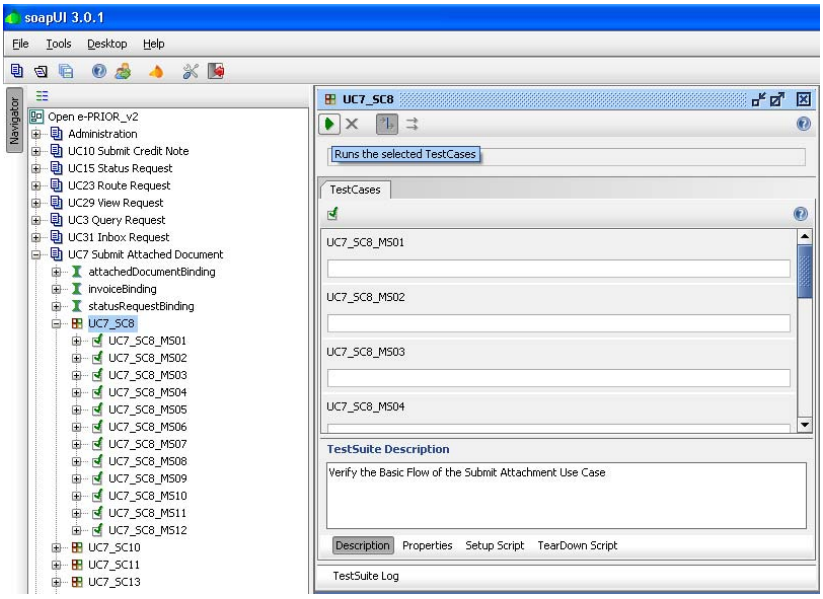
Before executing the tests make sure that the database and the soapUI log files were deleted by running the administration clean test cases (refer to §1.4 Administration).

Double click on one test suite; a new window will pop up with all the TC contained messages.



Now you can simply press the run button to run the tests. A test is considered successful when the bar is green. In case of an error, i.e. red bar, click the "TestSuite Log" button to see where and what the occurred error was.

Note that there is also the possibility to run only one test case message by double clicking on it and press run.



1.6. Valid Attachments Types

Below is a sample list of valid attachment types:

- image/x-png
- text/richtext
- text/plain
- application/msword
- image/jpg
- application/pdf
- application/vnd.ms-excel
- image/pjpeg
- text/xml
- image/tiff
- image/bmp
- image/gif
- application/vnd.ms-powerpoint
- text/html
- application/msoutlook

1.7. Special Test Case Scenarios

1.7.1. How to block JMS queue insertion

Related to: UC7_SC12, UC8_SC4, UC10_SC73, UC35_SC142

Requirement: The tester must simulate a technical error in incoming queue by blocking the JMS queue insertion.

Solution:

- Go to the JBoss sever console of the Test Environment. The URL to use is the following: http://158.168.153.52:8080/jmx-console/
- On the menu on the left, click the entry: "jboss.messaging.destination"
  A new sub-menu will be shown on the center of the page.
- On this menu, click the entry: "name=documentQueue,service=Queue"
  This link will open the list of the properties and the possible actions on the JMS queue named documentQueue.

- In the operations table (the one at the bottom of the page), look for the "stop" operation (third row) and click on the "Invoke" button.

This will stop the insertion on the queue.

After testing the system, the queue can be restarted by clicking the "Invoke" button of the "start" operation (second row of the operations table) on the same page.

1.7.2. How to simulate a database error

Related to: UC3_SC89, UC7_SC11

Requirement: The tester must verify the user receives the correct error when the system can not correctly process a request due to a database error.

Solution: Use a DB tool (ex. DBVisualizer) to access the correct database for the environment the tests are being executed on.
Execute the following statement to alter a column name:

ALTER TABLE EPR_TB_MESSAGE ALTER COLUMN MSG_DOCUMENTID RENAME TO MSG_DOCUMENTIDs;

The above statement will alter the name of the column MSG_DOCUMENTID of the table EPR_TB_MESSAGE, which is used by the system for processing a request, by adding another character at the end of the name. This will generate an error when trying to insert a document into the table.

Execute the following statement to restore the working situation after the tests:

ALTER TABLE EPR_TB_MESSAGE ALTER COLUMN MSG_DOCUMENTIDs RENAME TO MSG_DOCUMENTID;