NEMSIS TAC Whitepaper

NEMSIS v3 Web Services Guide

Date

November 2, 2011

November 14, 2011 (FINAL)

April 24, 2012 (Updated)

May 09, 2012 (Updated)

August 27, 2012 (updated)

Authors

Su Shaoyu - NEMSIS Lead Developer
N. Clay Mann – NEMSIS P.I.

Contributors

Grant Dittmer – Digital Innovations
Jerome Soller – Cognitech
Josh Legler – Utah Department of Health
EMSPIC Team – University of North Carolina-Chapel Hill

NEMSIS v3 Web Services Guide

Background

In NEMSIS V2, the dominant data submission mechanisms are ftp and web submission using Java Applet. Such procedures are manual and difficult to automate. The NEMSIS TAC has implemented a simple Web Services API for V2 data submission to the national EMS data warehouse. Although used only by a few states, it has proved to be an efficient alternative.

For software applications seeking V3 compliance, data submission via Web Services (WS) will be required. We hope the EMS software development community will take advantage of this mature technology and expand its usage to facilitate business workflow control and data exchange. If implemented properly, standardized NEMSIS WS API will provide the following advantages:

1. Fully automated electronic data exchange (including submission and retrieval) between the WS Consumer's system and the WS Provider's system. Human intervention could be minimized.

2. Better transaction management: The result of data submission / retrieval could be easily monitored without worrying about timeout. In NEMSIS systems, XML and business validation report could be available in real time, from the same communication channel.
3. Possibility to move to real-time / surveillance.
4. Some benefits are intrinsic characteristics of WS, such as vendor independence, scalability, and wide acceptance/adoption.

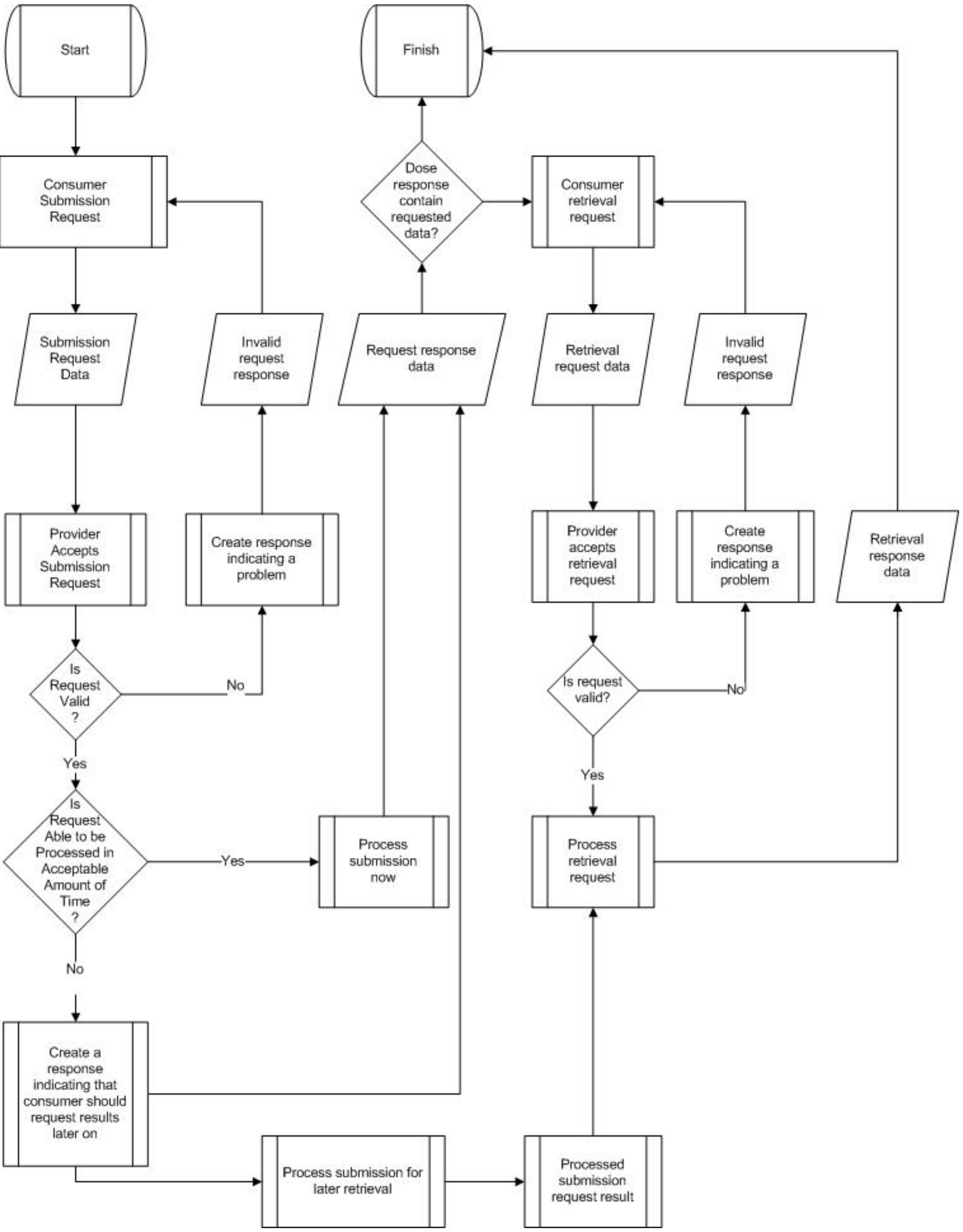
Purpose of NEMSIS v3 Web Services API

A unified interface to support data exchange between distributed NEMSIS v3 systems

Major design goals

1. Flexibility: no constraints on how to implement, which programming language to use, or what platform to deploy.
2. Efficiency: instant response, quick turn-around.
3. Unity: minimize confusion over information exchange.

Work Flow Diagram (Data Submission Request Only):


Start
Consumer Submission Request
Submission Request Data
Provider Accepts Submission Request
Is Request Valid ?
No
Create response indicating a problem
Invalid request response
Yes
Is Request Able to be Processed in Acceptable Amount of Time ?
Yes
Process submission now
No
Create a response indicating that consumer should request results later on
Process submission for later retrieval
Processed submission request result
Request response data
Dose response contain requested data?
Finish
Consumer retrieval request
Retrieval request data
Provider accepts retrieval request
Is request valid?
No
Create response indicating a problem
Invalid request response
Yes
Process retrieval request
Retrieval response data

Detail Requirements

Protocol

1. SOAP Web Service is required
2. RESTful Web Services will continue to be supported for version 2 data submission to NEMSIS TAC.

Security

HTTPS is the required communication channel. WS messages might include sensitive information, such as login credentials and Protected Health Information (PHI). These types of data must be protected at the transport-level. Further discussion about security is included in a later section.

Status Codes

1. Common errors (defined by WSDL) should be included in the Response Data and use the appropriate Status Code as indicators. It is up to the vendor to decide whether to utilize the SOAP header for extra error information which is solely used for their own purpose.
2. NEMSIS WSDL defines a set of standard response codes. All WS response messages include a status code.
    a. All error codes are negative integers.
    b. All success codes are positive integers.
    c. Integer “0” means that the requested action is not completed yet. The client should query the server later to retrieve final processing status.
    d. States or vendors could design their own error codes, with integers smaller than -100. Similarly, custom success codes should use integers greater than 100. It is hard to distinguish state custom codes and vendor custom codes. There are many cases in which state and vendor means the same thing.

3. For data exchange, if a state/vendor uses codes from -100 to 100, it is required to adopt the definition from the NEMSIS WSDL. For example, it is invalid to use “-1” as a status code for “Failed import of a file, because of the [FATAL] level Schematron rule violation” since this has been defined in the WSDL as “Invalid username and/or password”.

Important Common Objects

DataPayload

The Payload object, consisting of three data types (string, XML element, or Base64 encoded binary), is required for submitting data and returning reports. The table below gives an example of each payload.

PayloadOfString	`<DataPayload>`
	  `<PayloadOfString>`
	    `<encoding>text/xml</encoding>`
	    `<payload><![CDATA[<?xml version="1.0" encoding="UTF-8"?>`
	    `<product><name>pen</name>`

	<price>$1</price> </product>]]></payload> </PayloadOfString> </DataPayload>
PayloadOfBinary	<DataPayload> <PayloadOfBinary> <encoding>base64Binary</encoding> <compressed>false</ compressed> <payload>67542893472937492</payload> </PayloadOfBinary> </DataPayload>
PayloadOfXmlElement	<DataPayload> <PayloadOfXmlElement> <encoding>xml</encoding> <product> <name>pen</name> <price>$1</price> </product> </PayloadOfXmlElement> </DataPayload>

For PayloadOfBinary, the compression algorithm is ZIP as defined in http://www.pkware.com/documents/casestudies/APPNOTE.TXT (see http://en.wikipedia.org/wiki/ZIP_file_format for background reading.) To use PayloadOfBinary, the content of original XML file, or zipped XML file must be encoded using base64 alphabet (see http://en.wikipedia.org/wiki/Base64, http://www.w3.org/TR/xmlschema-2/#base64Binary for background reading and http://www.w3.org/2002/ws/databinding/examples/6/09/Base64BinaryElement/) into clear text.

On the client side, the workflow is

Original NEMSIS XML file --> Base64 encoding to a base64 string --> embed into <payload> element of <PayloadOfBinary>. An example could be:

```
<PayloadOfBinary>
        <encoding>base64Binary</encoding>
        <compressed>false</ compressed>
        <payload>67542893472937492</payload>
</PayloadOfBinary>
```

Or

Original NEMSIS XML file --> zip --> Base64 encoding the zip file to a base64 string --> embed into <payload> element of <PayloadOfBinary>. An example could be:

```
<PayloadOfBinary>
        <encoding>base64Binary</encoding>
        <compressed>true</ compressed>
```

```
                    <payload>eqioweuqiow</payload>
                </PayloadOfBinary>
```

On the server side, after the server receives the payload, the string value needs to be base64-decoded. Depending on the value of element <compressed>, the decoded data can be directly save to a XML file (when compressed=false), or a zip file (when compressed=true).

It is also necessary to emphasize that on client side, the zip should be executed on ONE SINGLE NEMSIS XML file since v3 Web Service is designed to transmit ONE SINGLE file. Servers should reject the submission if the unzipped payload contains directory structure or/and multiple files.

DataSchema and DataSchemaVersion

Four NEMSIS schemas are predefined. States/vendors could define custom schema used in data exchange. "NEMSIS EMS" and "NEMSIS Demographics" are required.

Combined with DataSchema, DataSchemaVersion helps to identify the exact schema used for the payload. For example, DataSchema=”NEMSIS EMS” and DataSchemaVersion=”2.2.1” is a valid combination. It points to current v2 schema. However, there is no version 2.5.6 for NEMSIS EMS data. So DataSchema=”NEMSIS EMS” and DataSchemaVersion=”2.5.6” is not a valid combination.

SubmitDataReport

SubmitDataReport is for the report of data submission processing. It should include at least one XmlValidationErrorReport element for the XML validation report. The format implementation is as follows:

1. If the submission fails XML validation, SubmitDataReport is required to contain one XmlValidationErrorReport. Since the submission is rejected at that step, SchematronReport must not be included.
2. If the submission passes XML validation, SubmitDataReport must contain
    a. XmlValidationErrorReport, with XmlValidationErrorReport’s TotalErrorCount set to zero.
    b. SchematronReport , which contains either complete Schematron result file(s), or digested Schematron result file(s), or both of them. The decision is up to the state/vendor.
3. SubmitDataReport may include optional CustomReport element(s). It is designed to help the state/vendor handle their specific requirements.

XmlElementInfo

This is used to identify the offending XML element reported in XML validation or Schematron validation. Line/column numbers, or XPATH location, could be used to identify the position of an XML element. Usually, DOM parsers (validators) report XPATH information and SAX/StAX parsers (validators) report line/column numbers. Some special XML processors, like SAXON EE/PE version, can report both. The design is to allow line/column numbers and XPATH location to be reported together: but it is not required to do so. The design also allows position information to be “unknown”.

XmlValidationError

There are two major types of XML validation errors. (For complete list of XML validation errors, check http://svn.apache.org/viewvc/xerces/java/trunk/src/org/apache/xerces/impl/msg/ .) One could be pinpointed to a particular element: for example, if the element is defined as an integer but the value "ABC" is submitted. For this kind of error, use XmlElementInfo to report the offending element. Another type of error is not focused on one element: for example, if a CSV file is submitted for XML validation. In this case, use XmlGeneralErrorList to include a list of error messages.

Main Functions

Communication in Web Services is defined by the request message and response message. There are four required interfaces (functions): SubmitData, RetrieveStatus, and QueryLimit. Data structure for the request and response messages corresponding to these functions are defined.

Username, password, and organization are always required for any WS request. This information could be included in the request's SOAP header, although we fail to find any advantage. For the proposed functions, values for element requestType are predefined. The state/vendor could develop other functions with a custom value for "requestType". All functions also include an element of "additionalInfo" to allow for custom input.

NEMSIS WS response messages all include a status code (discussed above) and an echoing "requestType", set to the same value as in the request. Except for the function of QueryLimit, they also include a unique identifier "requestHandle": a system-assigned unique identifier at the server side for the transaction that takes place as a result of the request. In multiple query response situations, the "requestHandle" (or simply referred to as "handle") is used as a common point of reference.

SubmitData

To submit data, the client needs to specify the data payload, data schema name, and schema version. As discussed above, the combination of schema name and version will decide the standard for the data submitted. After the data are submitted, it is subject to XML validation and Schematron rule checking. The submission is to be rejected due to:

1. XML validation fails
2. National Schematron rule(s) violated
3. Critical state/vendor Schematron rule(s) violated
4. Other critical business rule(s) violated

The response for data submission could be synchronous or asynchronous: in the synchronous situation, an object of "SubmitDataReport" is included in the response, together with status code, handle, and echoing requestType. In the asynchronous situation, the server is not able to process the submitted data in time. Then "SubmitDataReport" is not included in the response. The client should use the assigned requestHandle in the response message to query the server later.

RetrieveStatus

RetrieveStatus function is used to retrieve results from the previous submission. This is most important for an asynchronous response from the submission. However, even with a synchronous response, the server might save the submission processing status result for a limited time period. In this case, if the client needs to retrieve the result later, it can use the returned requestHandle.

The response to RetrieveStatus could be:

1. If the process is still not finished (pending), the server should return the same requestHandle and status code of pending (0).
2. If the process is completed and status is available, the server should return the same requestHandle, proper status code, and a SubmitDataReport, depending on the original request.
3. The server should return with proper status code if the status is not available because the status has expired, been deleted, or the requestHandle is not a valid identifier.

QueryLimit

Different web servers and WS implementations could apply a unique constraint on the size of the whole Web Service message. WS consumers can use this interface to query WS server's configuration for this limit.

The response to QueryLimit could be:

1. A positive integer to indicate the size limit on data payload, expressed in KB (1024 bytes).
2. A negative integer (-1) and an error status code.

Use cases:

Case 1 – Submit Data Request, Synchronous scenario

Step 1. Agency “Elmo” wants to send a set of EMS records to state “Sesame Street”. A Submit Request is sent with “Request Type” = “SubmitData”. The SOAP message looks like this:

SOAP Message 1

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ws="http://ws.nemsis.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <ws:SubmitDataRequest>
      <ws:username>sutest</ws:username>
      <ws:password>openthedoor</ws:password>
      <ws:organization>sesame</ws:organization>
      <ws:requestType>SubmitData</ws:requestType>
      <ws:SubmitPayload>
        <!--You have a CHOICE of the next 3 items at this level-->
<!--
        <ws:PayloadOfString>
          <ws:encoding>text/xml</ws:encoding>
          <ws:payload>?</ws:payload>
        </ws:PayloadOfString>
-->
        <ws:PayloadOfBinary>
          <ws:encoding>base64Binary</ws:encoding>
          <ws:compressed>true</ws:compressed>
          <ws:payload>cid:103713444083</ws:payload>
        </ws:PayloadOfBinary>
<!--
        <ws:PayloadOfXmlElement>
          <ws:encoding>xml</ws:encoding>-->
          <!--You may enter ANY elements at this point-->
<!--        </ws:PayloadOfXmlElement>
-->
      </ws:SubmitPayload>
      <ws:requestDataSchema>bbbb</ws:requestDataSchema>
      <ws:schemaVersion>3456</ws:schemaVersion>
      <ws:additionalInfo>fun</ws:additionalInfo>
    </ws:SubmitDataRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Step 2. The State receives the message and successfully processes it within a reasonable amount of time. It replies with a “requestHandle”, a system-assigned unique identifier for the transaction of handling Elmo’s WS request. Then the response message looks like this:

SOAP Message 2

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <SubmitDataResponse xmlns="http://ws.nemsis.org/">
      <requestType>SubmitData</requestType>
      <requestHandle>12345678</requestHandle>
      <statusCode>1</statusCode>
      <reports>
        <XmlValidationErrorReport>
          <TotalErrorCount>0</TotalErrorCount>
        </XmlValidationErrorReport>
        <SchematronReport>
          <CompleteSchematronReport>
            <CompleteReport>
              <PayloadOfString>
                <payload>test string payload. should be XML file content.</payload>
              </PayloadOfString>
            </CompleteReport>
          </CompleteSchematronReport>
        </SchematronReport>
      </reports>
    </SubmitDataResponse>
  </S:Body>
</S:Envelope>
```

Case 2. Submit Data Request, Asynchronous scenario

Step 1. Same as step 1 in case 1, Agency “Elmo” sends XML to state “Sesame Street”.

Step 2. The State receives the xml file. Because the system is too busy handling other agencies’ requests, the response message looks like this:

SOAP Message 3

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <SubmitDataResponse xmlns="http://ws.nemsis.org/">
      <requestType>SubmitData</requestType>
      <requestHandle>12345678</requestHandle>
      <statusCode>0</statusCode>
    </SubmitDataResponse>
  </S:Body>
</S:Envelope>
```

Step 3. RetrieveStatus Request for previous submission

After 15 minutes, as a responsible agent, Elmo thinks it has given the state system enough time to process the data. Using “requestHandle”, Elmo can query the state system to get the result of its previous WS request. The SOAP message for this kind of “RetrieveStatus” request looks like this:

SOAP Message 4

```
    <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ws="http://ws.nemsis.org/">
      <soapenv:Header/>
      <soapenv:Body>
        <ws:RetrieveStatusRequest>
          <ws:username>sutest</ws:username>
          <ws:password>openthedoor</ws:password>
          <ws:organization>sesame</ws:organization>
          <ws:requestType>RetrieveStatus</ws:requestType>
          <ws:requestHandle>12345678</ws:requestHandle>
          <!--Optional:-->
          <ws:originalRequestType>SubmitData</ws:originalRequestType>
          <ws:additionalInfo>fun</ws:additionalInfo>
        </ws:RetrieveStatusRequest>
      </soapenv:Body>
</soapenv:Envelope>
```

Step 4. Response to Step 3’s Request

Output 1: If the state has finished processing the submission request, then the response message looks like “SOAP Message 2”.

Output 2: If the state still hasn’t finished processing the submission request, the response message looks like “SOAP Message 3”. So Elmo needs to come back later to check the status (repeat step 3).

Note: if for any reason Elmo forgets the status of his submission, even after he has received notification of a successful submission, Elmo should be able to send a “RetrieveStatus” request to the state again. As a good bookkeeper, the state should be able to response back with “SOAP Message 2”. Certainly, if the requestHandle is actually for a submission 10 years ago, it is very possible that the state will tell Elmo the status has expired.

Discussions:

1. Authentication/Security Implementation
    a. Some vendors might prefer to utilize the security element in the SOAP message's header. Web Service Security Specification, published by OASIS (http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SOAPMessageSecurity.pdf), provides a set of mechanisms to enforce message integrity and confidentiality. However, due to the vast differences among EMS vendors/deployments, the NEMSIS TAC will not specify a particular SOAP security header structure. It is very possible, and legitimate, that a vendor may choose to include the authentication information in the SOAP header. In such instances, both ends of the WS communication have to reach a mutual understanding about the underlining security mechanism (and ignore the username/password parameters in the proposed object nemsisV3WsRequest). For example, they can choose to use (a)symmetric keys to encrypt username/password.
    b. It was stated in the draft NEMSIS WS Guide that "passwords should never be sent as clear-text". We should revise this statement to be "passwords should never be sent as clear-text over a non-secured channel". Again, we require that all WS communications use HTTPS since we need to protect not only the header section, but also the embedded data payload. Hashing, or password digestion, is not more secure than a password in clear text, if it is not sent on a secure channel or token not encrypted. (See http://www.oasis-open.org/committees/download.php/16782/wss-v1.1-spec-os-UsernameTokenProfile.pdf, line 159-162. On 10/19/2011, some researchers demonstrated at the ACM conference how to break W3C standard XML encryption.)
    c. Password hashing might create unique challenges for implementation. For example, when using Microsoft Active Directory (AD) authentication, the WS recipient has to have a way to figure out the clear text password, then it can be sent to AD for authentication. OASIS documents have detailed discussions about other security concerns. (See references on page 27.)

2. Web Services is not a perfect solution for transmitting large amounts of data in one WS message. Most web servers and WS implementations apply a limit on the size of one Web Service message. Oversized WS messages could be rejected by a web server, even before reaching the WS receiving codes. Also, due to performance considerations, WS recipients might prefer small WS messages. Researchers at IBM have published a study on this issue. (See reference 4 on page 27) We strongly encourage software vendors to develop appropriate strategies that fit their own software/hardware environments.

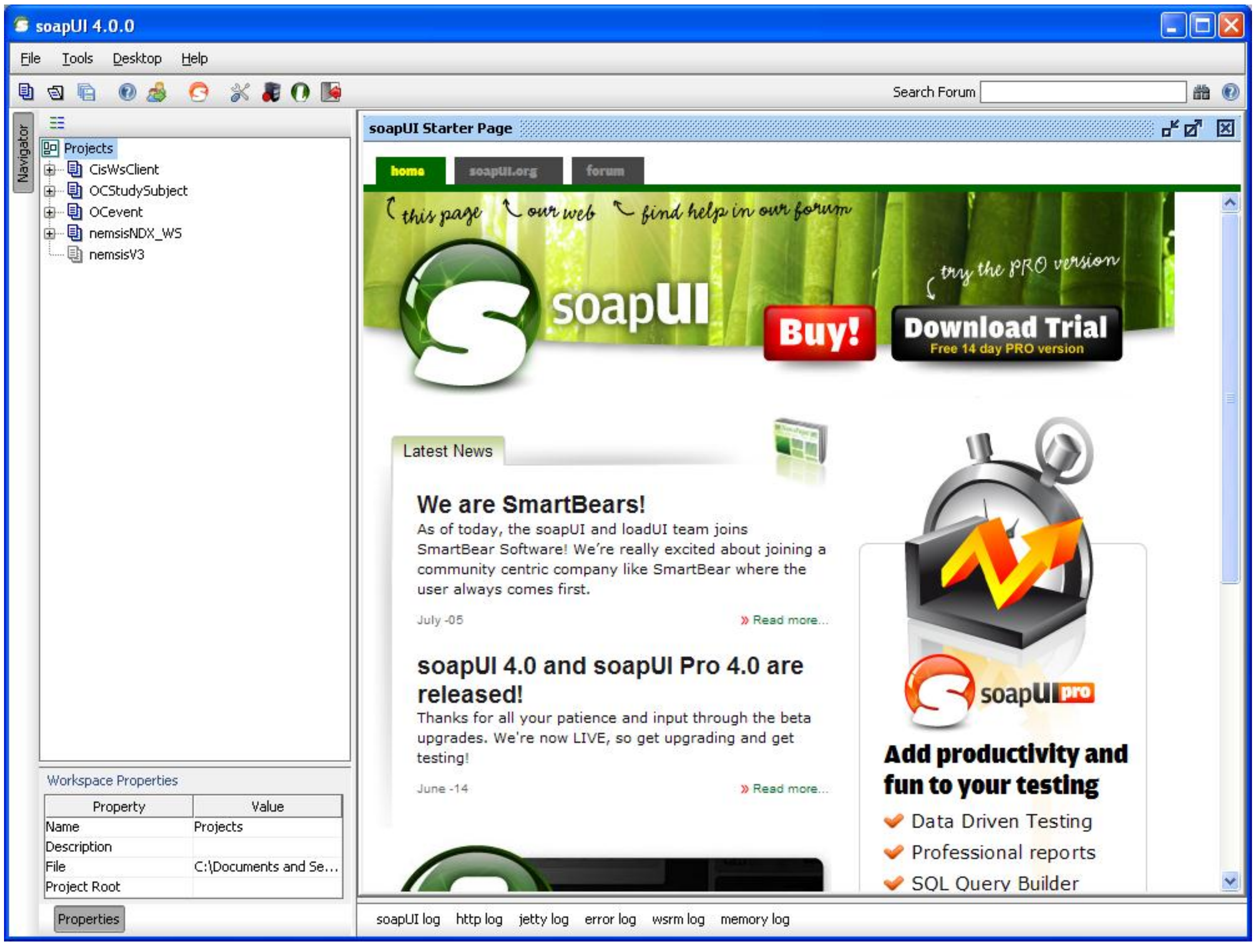
3. Unfortunately, it is hard to predict the size of one NEMSIS EMS record, especially when medical image files might be embedded. This is why the proposed interface "QueryLimit" should return the limit on the size of the string, instead of how many EMS records are contained.

4. Many web servers and web clients (like most web browsers) have built-in HTTP compression capability to facilitate faster transmission speeds. But, in the case where the WS client/server doesn't have HTTP compression built-in, it is necessary to consider compressing the NEMSIS XML File. In most cases, NEMSIS XML files could be compressed easily. We often see the compression rates of 30:1. So a 30MB NEMSIS file could be compressed to 1MB. After Base64 encoding, which increases the size of binary data by 33%, the final string to transmit is about 1.4MB.

A Quick Guide for Using soapUI

This is just a simple guide to use soapUI, one of the most popular open source applications for testing Web Services. I believe that most readers are already familiar with it. Feel free to skip this section. The following is for Windows users. Check http://www.soapui.org/Getting-Started/getting-started.html for additional information.

1. Download soapUI from http://www.soapui.org/. (I prefer to download the standalone version.)
2. Install soapUI to your system, if you download the installer. If you download the standalone version, simply unzip it to your hard drive; then go into its bin folder, and run “soapui.bat” to start soapUI.
3. Start soapUI from Programs, or run “soapui.bat”. You should see a starter page like this:

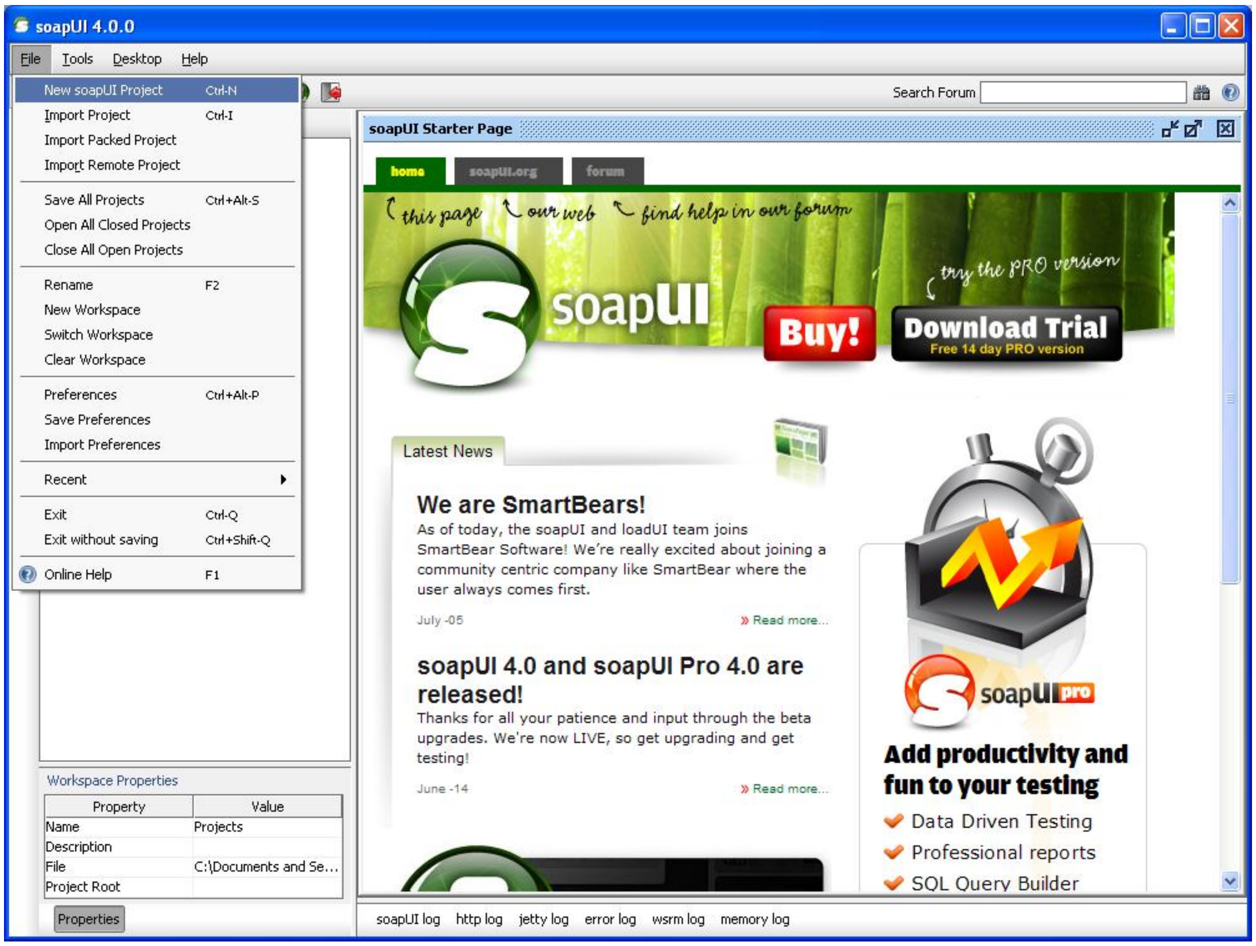


We are SmartBears!
As of today, the soapUI and loadUI team joins SmartBear Software! We're really excited about joining a community centric company like SmartBear where the user always comes first.
July -05 » Read more...

soapUI 4.0 and soapUI Pro 4.0 are released!
Thanks for all your patience and input through the beta upgrades. We're now LIVE, so get upgrading and get testing!
June -14 » Read more...

Property	Value
Name	Projects
Description	
File	C:\Documents and Se...
Project Root	

4. Click on File→"new soapUI Project" to start a new project.



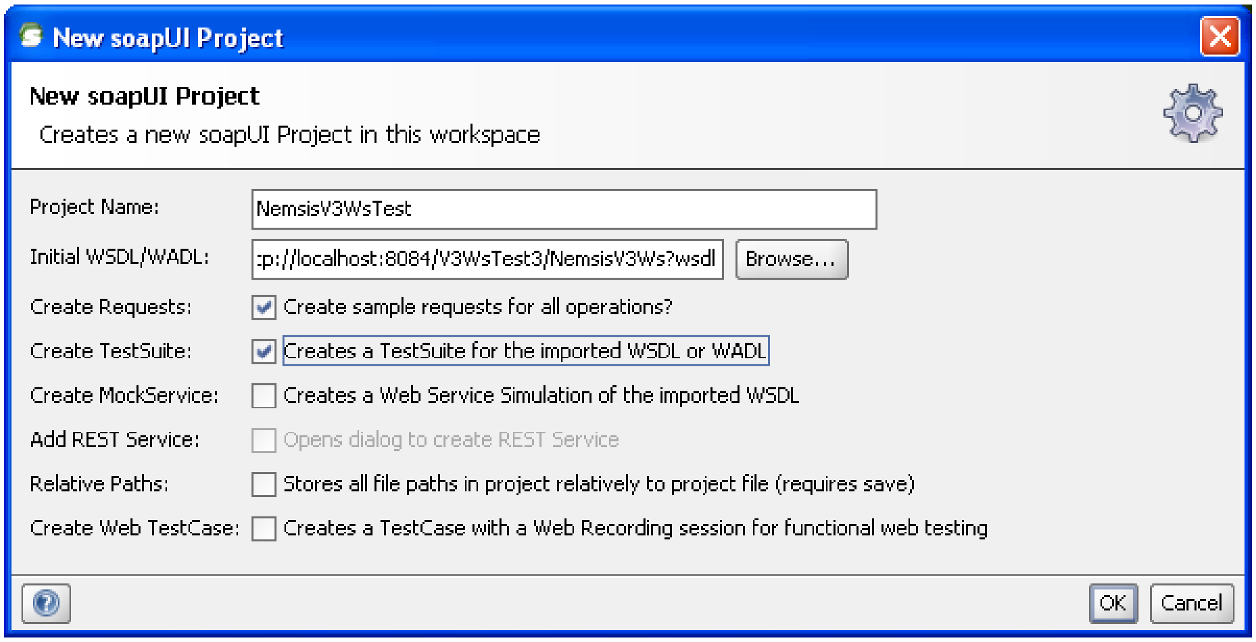
We are SmartBears!
As of today, the soapUI and loadUI team joins SmartBear Software! We're really excited about joining a community centric company like SmartBear where the user always comes first.
July -05 » Read more...

soapUI 4.0 and soapUI Pro 4.0 are released!
Thanks for all your patience and input through the beta upgrades. We're now LIVE, so get upgrading and get testing!
June -14 » Read more...

Property	Value
Name	Projects
Description	
File	C:\Documents and Se...
Project Root	

5. Name your project. Since I deployed my NEMSIS WS on my developing desktop, the WSDL could be accessed at http://nemsis.org/v3/NEMSIS_V3_8.wsdl. Copy this to the text box next to "Initial WSDL/WADL". In many cases, you might save the published WSDL file to your hard drive – then you can use the Browse button to locate the WSDL file.
It would be wise to investigate the WSDL file first: you have to make sure all referenced files are accessible. A funny thing about a WSDL file published by Spring Framework is that it usually doesn't include the full path in the SOAP address. For example, the line
 <soap:address location=" http://nemsis.org/v3/NEMSIS_V3_8.wsdl " />
could simply be
 <soap:address location="/v3/NEMSIS_V3_8.wsdl " />
In that case, you'd better save the WSDL and edit it. (You can also change the soapUI test property to force it point to the real server address.)

Remember to create TestSuite.



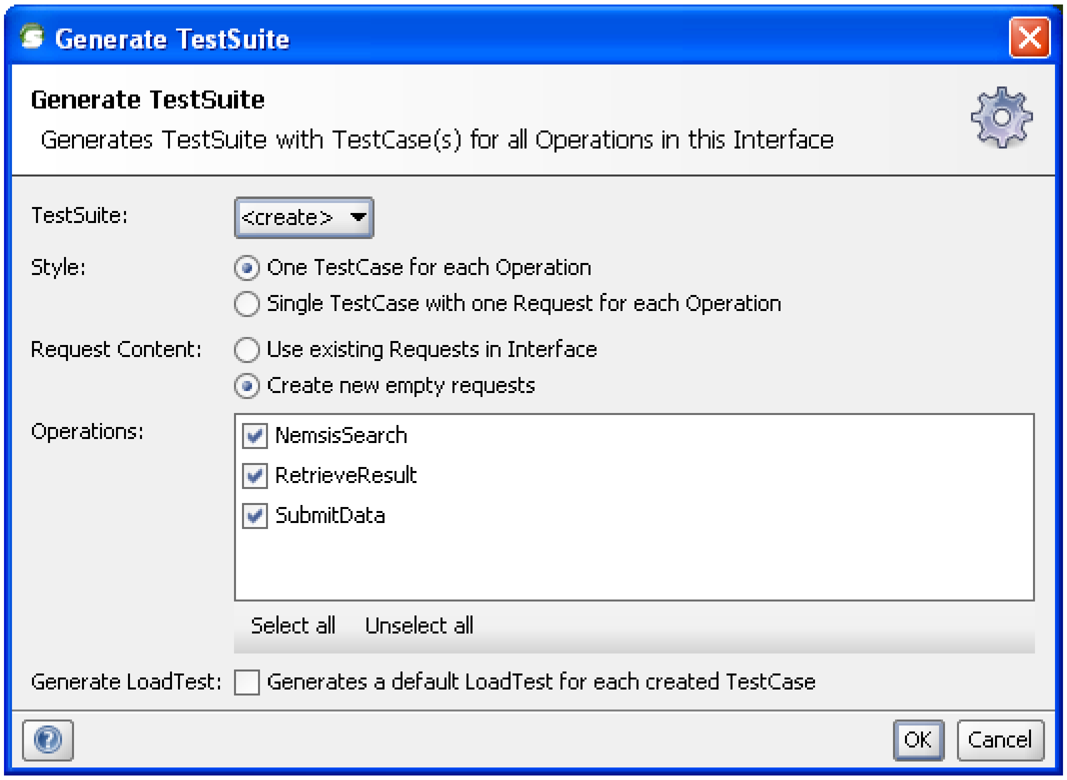
New soapUI Project

New soapUI Project

Creates a new soapUI Project in this workspace

Project Name:	NemsisV3WsTest
Initial WSDL/WADL:	:p://localhost:8084/V3WsTest3/NemsisV3Ws?wsdl
Create Requests:	☑ Create sample requests for all operations?
Create TestSuite:	☑ Creates a TestSuite for the imported WSDL or WADL
Create MockService:	☐ Creates a Web Service Simulation of the imported WSDL
Add REST Service:	☐ Opens dialog to create REST Service
Relative Paths:	☐ Stores all file paths in project relatively to project file (requires save)
Create Web TestCase:	☐ Creates a TestCase with a Web Recording session for functional web testing

6. Now, soapUI will parse the WSDL and generate TestSuite for all functions defined in WSDL file.



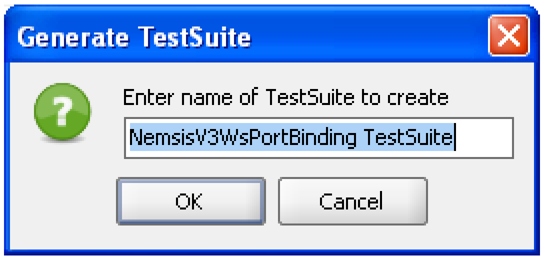
Generate TestSuite

Generate TestSuite

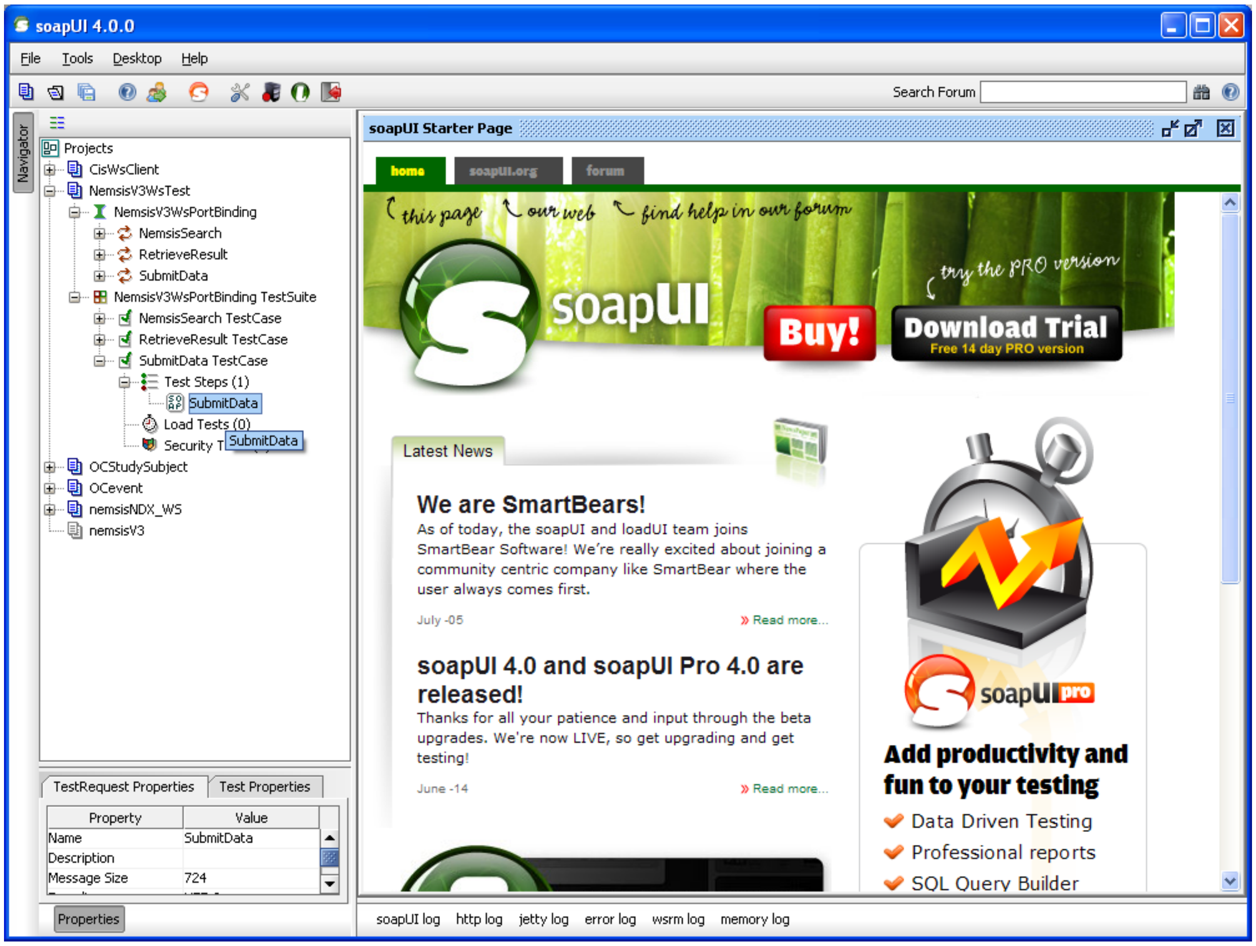
Generates TestSuite with TestCase(s) for all Operations in this Interface

TestSuite:	<create>
Style:	● One TestCase for each Operation ○ Single TestCase with one Request for each Operation
Request Content:	○ Use existing Requests in Interface ● Create new empty requests
Operations:	☑ NemsisSearch ☑ RetrieveResult ☑ SubmitData
Generate LoadTest:	☐ Generates a default LoadTest for each created TestCase

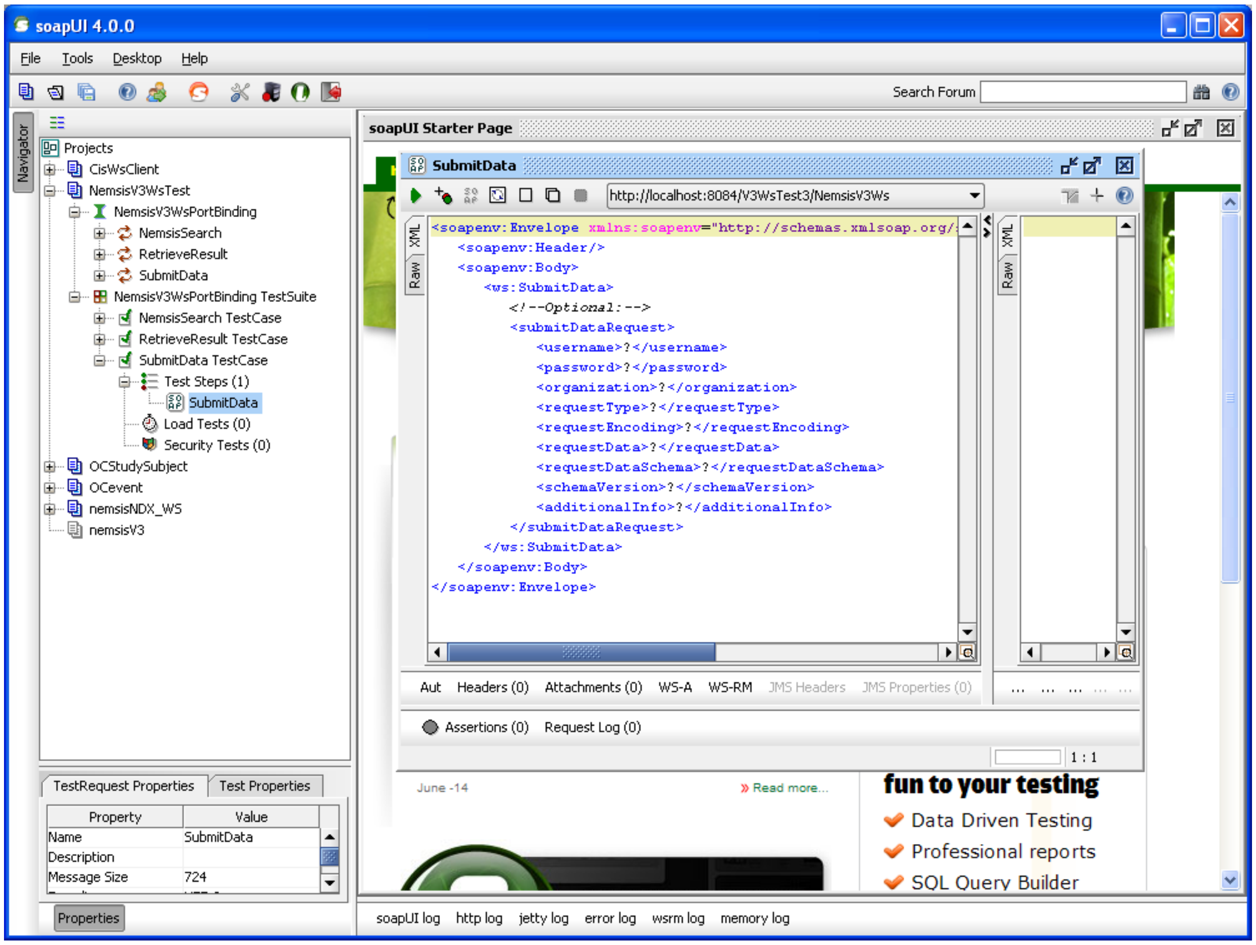
7. Give the TestSuite a name


Generate TestSuite
Enter name of TestSuite to create
NemsisV3WsPortBinding TestSuite
OK
Cancel

8. In project NemsisV3WsTest, go into "NemsisV3WsPortBinding TestSuite", "SubmitData TestCase", "Test Steps(1)", "SubmitData":



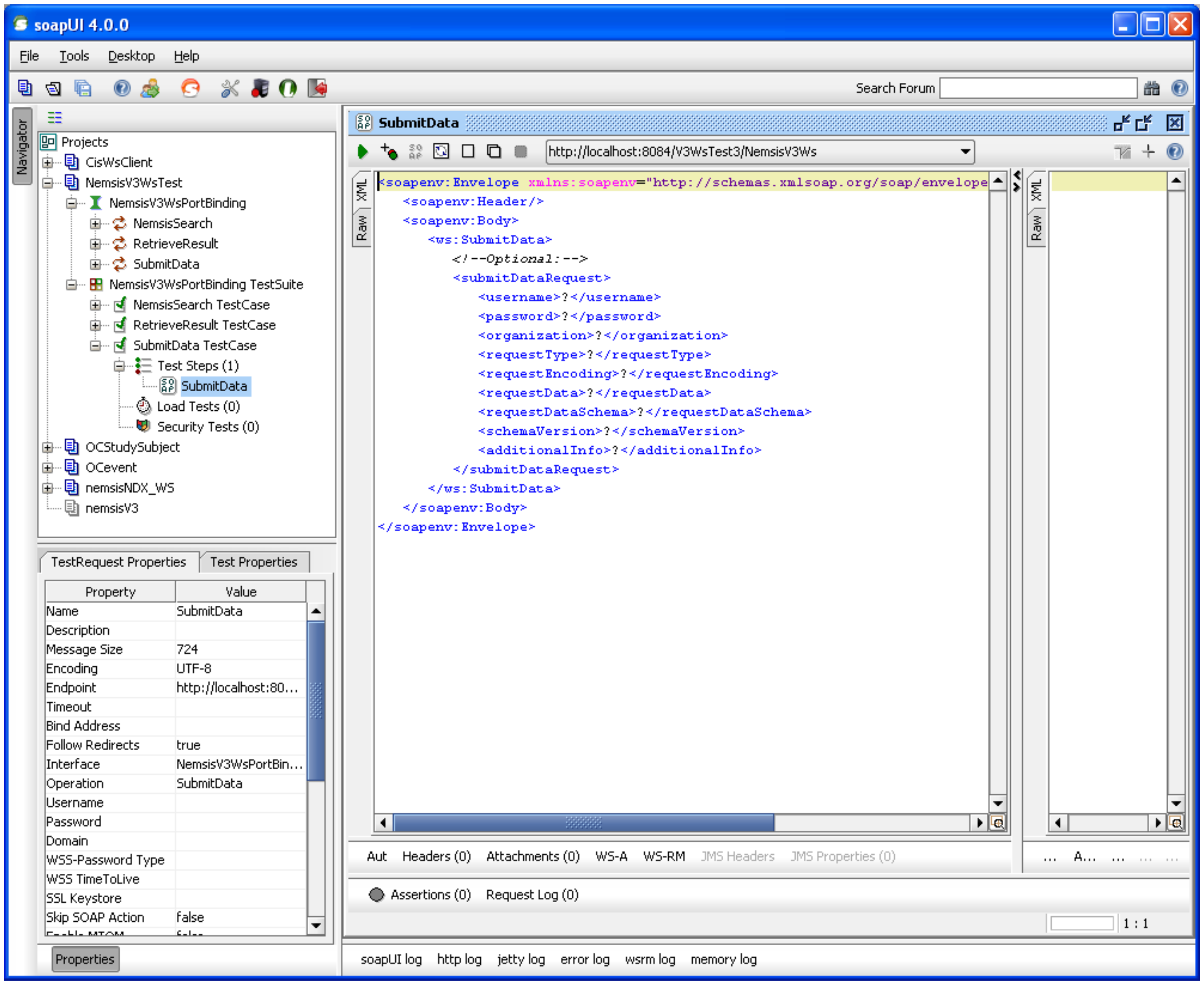
Double click on "SubmitData",



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
   <soapenv:Header/>
   <soapenv:Body>
      <ws:SubmitData>
         <!--Optional:-->
         <submitDataRequest>
            <username>?</username>
            <password>?</password>
            <organization>?</organization>
            <requestType>?</requestType>
            <requestEncoding>?</requestEncoding>
            <requestData>?</requestData>
            <requestDataSchema>?</requestDataSchema>
            <schemaVersion>?</schemaVersion>
            <additionalInfo>?</additionalInfo>
         </submitDataRequest>
      </ws:SubmitData>
   </soapenv:Body>
</soapenv:Envelope>
```

Property	Value
Name	SubmitData
Description	
Message Size	724

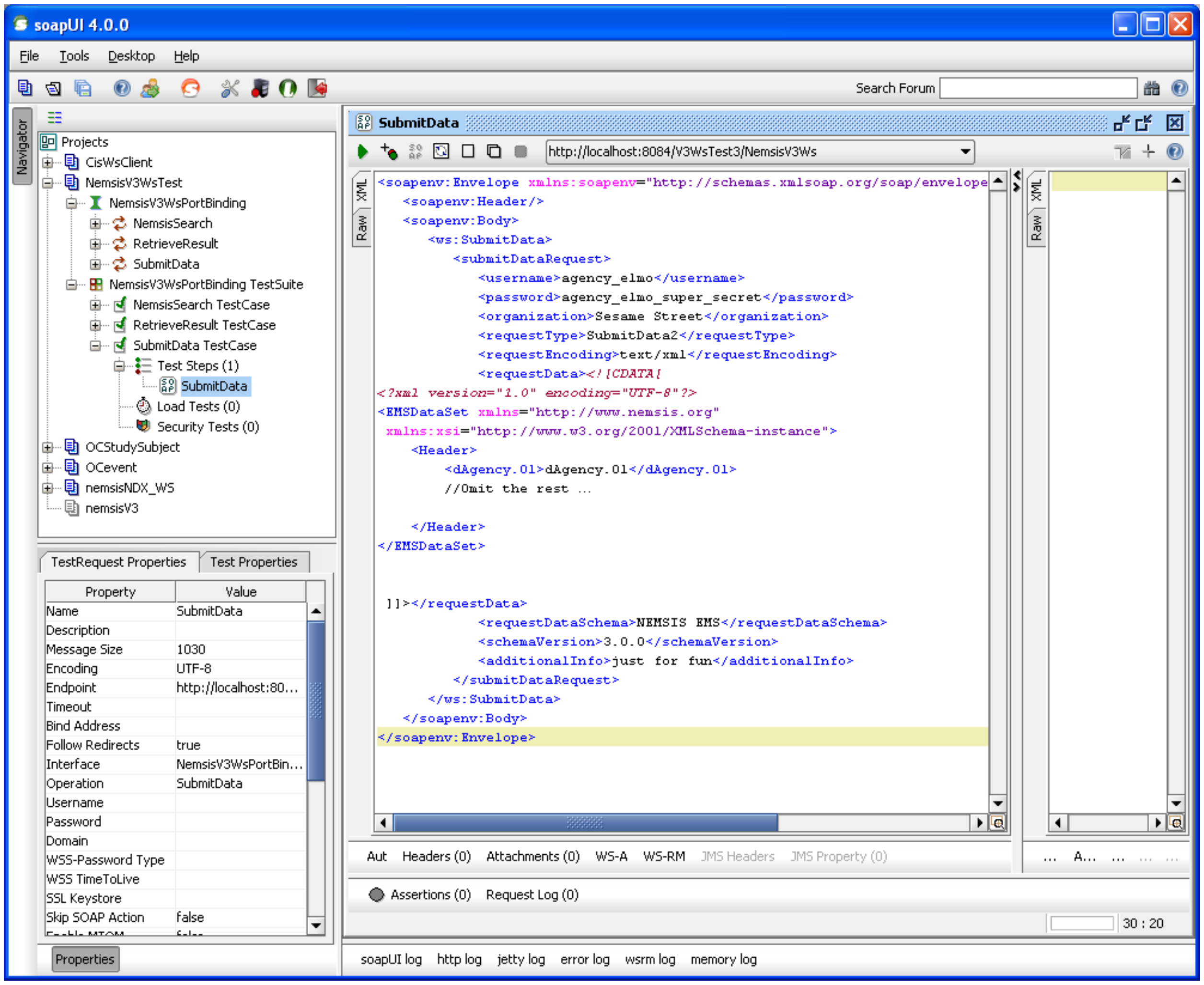
Now is a good time to close “Starter Page” and maximize the new “SubmitData” window.



Property	Value
Name	SubmitData
Description	
Message Size	724
Encoding	UTF-8
Endpoint	http://localhost:80…
Timeout	
Bind Address	
Follow Redirects	true
Interface	NemsisV3WsPortBin…
Operation	SubmitData
Username	
Password	
Domain	
WSS-Password Type	
WSS TimeToLive	
SSL Keystore	
Skip SOAP Action	false

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope
   <soapenv:Header/>
   <soapenv:Body>
      <ws:SubmitData>
         <!--Optional:-->
         <submitDataRequest>
            <username>?</username>
            <password>?</password>
            <organization>?</organization>
            <requestType>?</requestType>
            <requestEncoding>?</requestEncoding>
            <requestData>?</requestData>
            <requestDataSchema>?</requestDataSchema>
            <schemaVersion>?</schemaVersion>
            <additionalInfo>?</additionalInfo>
         </submitDataRequest>
      </ws:SubmitData>
   </soapenv:Body>
</soapenv:Envelope>
```

9. SoapUI has already created a template for you to fill in real data. You can copy and paste the content of “SOAP Message 1” in this file’s “User Cases” section.


soapUI 4.0.0
File Tools Desktop Help
Search Forum
Navigator
SubmitData
http://localhost:8084/V3WsTest3/NemsisV3Ws
XML
Raw
Aut Headers (0) Attachments (0) WS-A WS-RM JMS Headers JMS Property (0)
Assertions (0) Request Log (0)
30 : 20
TestRequest Properties Test Properties
Properties
soapUI log http log jetty log error log wsrm log memory log

```
Projects
  CisWsClient
  NemsisV3WsTest
    NemsisV3WsPortBinding
      NemsisSearch
      RetrieveResult
      SubmitData
    NemsisV3WsPortBinding TestSuite
      NemsisSearch TestCase
      RetrieveResult TestCase
      SubmitData TestCase
        Test Steps (1)
          SubmitData
        Load Tests (0)
        Security Tests (0)
  OCStudySubject
  OCevent
  nemsisNDX_WS
  nemsisV3
```

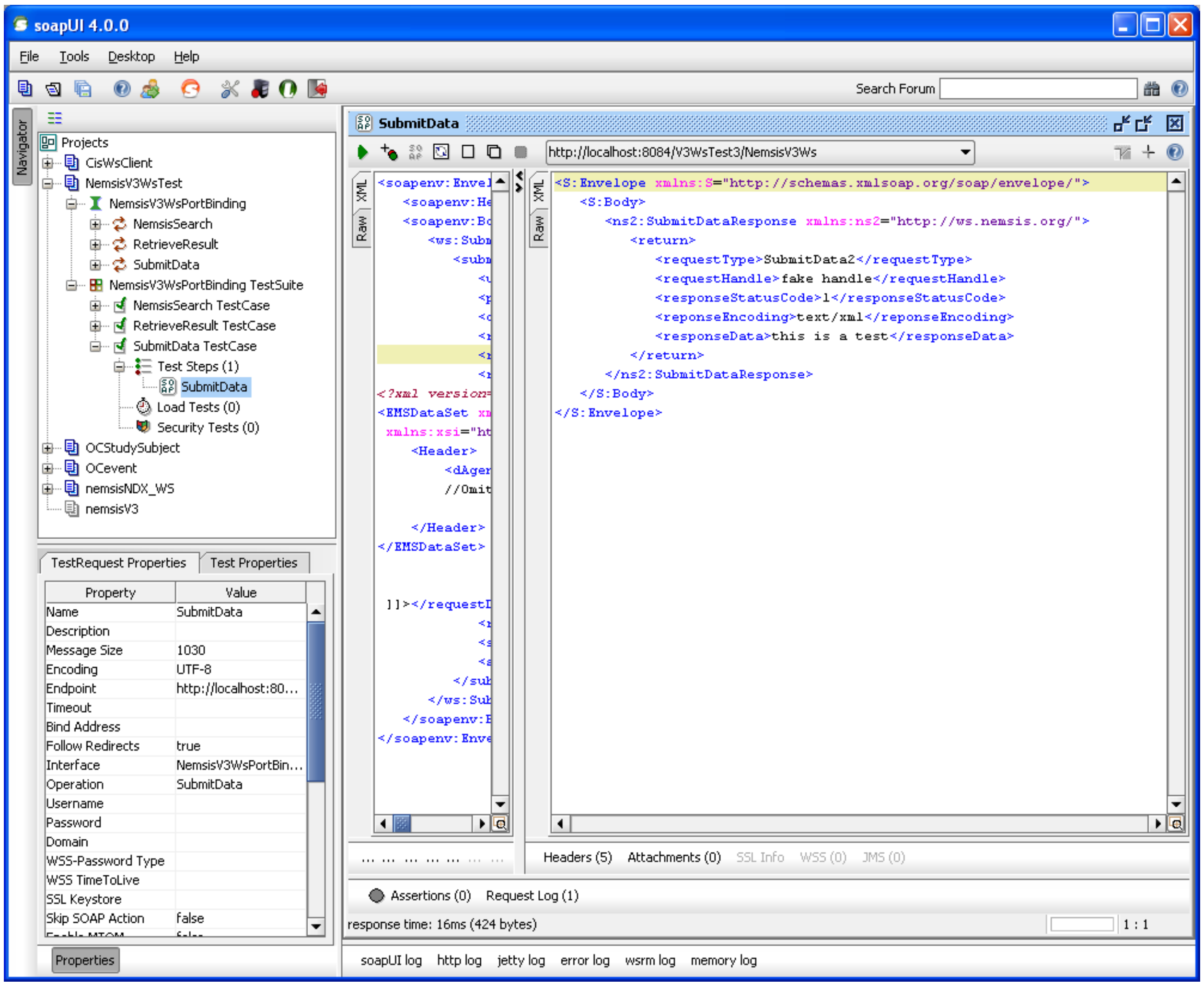
Property	Value
Name	SubmitData
Description	
Message Size	1030
Encoding	UTF-8
Endpoint	http://localhost:80...
Timeout	
Bind Address	
Follow Redirects	true
Interface	NemsisV3WsPortBin...
Operation	SubmitData
Username	
Password	
Domain	
WSS-Password Type	
WSS TimeToLive	
SSL Keystore	
Skip SOAP Action	false

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope
   <soapenv:Header/>
   <soapenv:Body>
      <ws:SubmitData>
         <submitDataRequest>
            <username>agency_elmo</username>
            <password>agency_elmo_super_secret</password>
            <organization>Sesame Street</organization>
            <requestType>SubmitData2</requestType>
            <requestEncoding>text/xml</requestEncoding>
            <requestData><![CDATA[
<?xml version="1.0" encoding="UTF-8"?>
<EMSDataSet xmlns="http://www.nemsis.org"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <Header>
       <dAgency.01>dAgency.01</dAgency.01>
       //Omit the rest ...

   </Header>
</EMSDataSet>


 ]]></requestData>
            <requestDataSchema>NEMSIS EMS</requestDataSchema>
            <schemaVersion>3.0.0</schemaVersion>
            <additionalInfo>just for fun</additionalInfo>
         </submitDataRequest>
      </ws:SubmitData>
   </soapenv:Body>
</soapenv:Envelope>
```

10. Magic moment: click on the green triangle button in left-upper corner of the “SubmitData” window. This will submit the SOAP message to the designated WS server (in Web Services terms, this is the “endpoint”. Make sure your WS server is up and running!). You can see the result in the right panel: since we are testing, my super-simple WS server returns a super-simple message.


soapUI 4.0.0
File Tools Desktop Help
Search Forum
Navigator
SubmitData
XML
Raw
Headers (5) Attachments (0) SSL Info WSS (0) JMS (0)
Assertions (0) Request Log (1)
soapUI log http log jetty log error log wsrm log memory log
TestRequest Properties Test Properties
Properties

Projects
- CisWsClient
- NemsisV3WsTest
  - NemsisV3WsPortBinding
    - NemsisSearch
    - RetrieveResult
    - SubmitData
  - NemsisV3WsPortBinding TestSuite
    - NemsisSearch TestCase
    - RetrieveResult TestCase
    - SubmitData TestCase
      - Test Steps (1)
        - SubmitData
      - Load Tests (0)
      - Security Tests (0)
- OCStudySubject
- OCevent
- nemsisNDX_WS
- nemsisV3

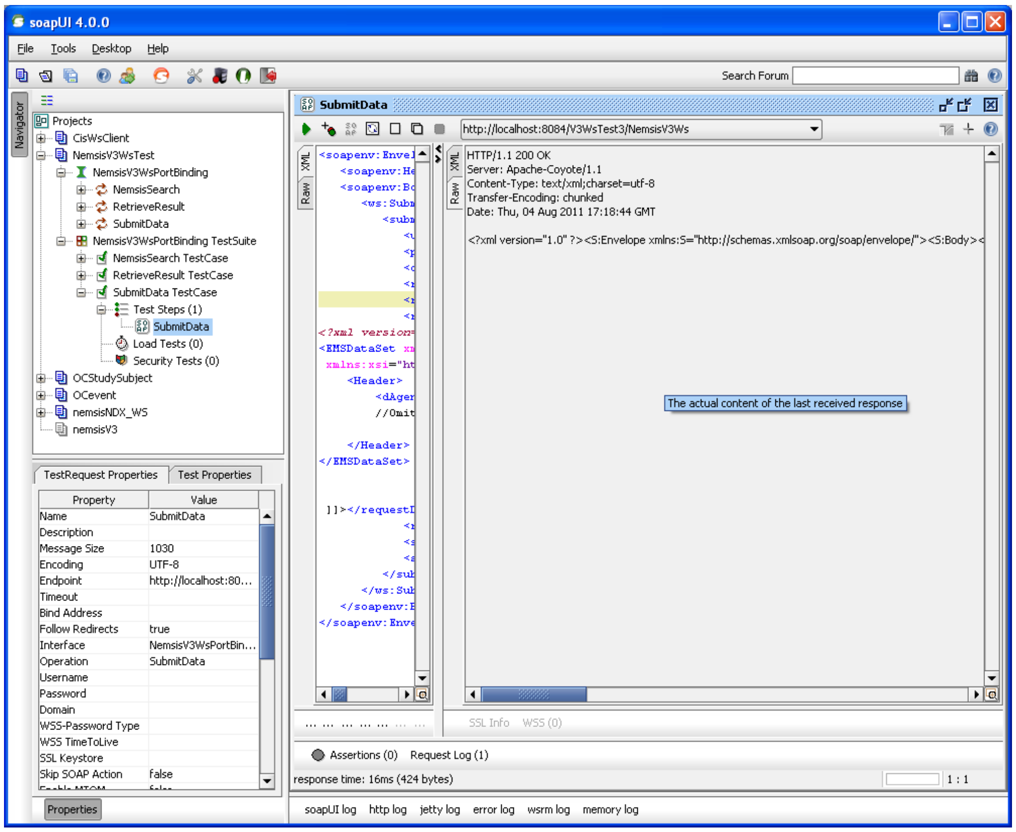
http://localhost:8084/V3WsTest3/NemsisV3Ws

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:SubmitDataResponse xmlns:ns2="http://ws.nemsis.org/">
         <return>
            <requestType>SubmitData2</requestType>
            <requestHandle>fake handle</requestHandle>
            <responseStatusCode>1</responseStatusCode>
            <reponseEncoding>text/xml</reponseEncoding>
            <responseData>this is a test</responseData>
         </return>
      </ns2:SubmitDataResponse>
   </S:Body>
</S:Envelope>
```

Property	Value
Name	SubmitData
Description	
Message Size	1030
Encoding	UTF-8
Endpoint	http://localhost:80…
Timeout	
Bind Address	
Follow Redirects	true
Interface	NemsisV3WsPortBin…
Operation	SubmitData
Username	
Password	
Domain	
WSS-Password Type	
WSS TimeToLive	
SSL Keystore	
Skip SOAP Action	false

response time: 16ms (424 bytes) 1 : 1

11. SoapUI provides a wide range of functionaries to help you debug a WS application. For example, you can check the raw request/response message, error log, http log. And you can change your endpoint, set up security, attach documents, etc. I attached some snapshots to provide further guidance. To explore the full power of soapUI, read its online document and the forum.

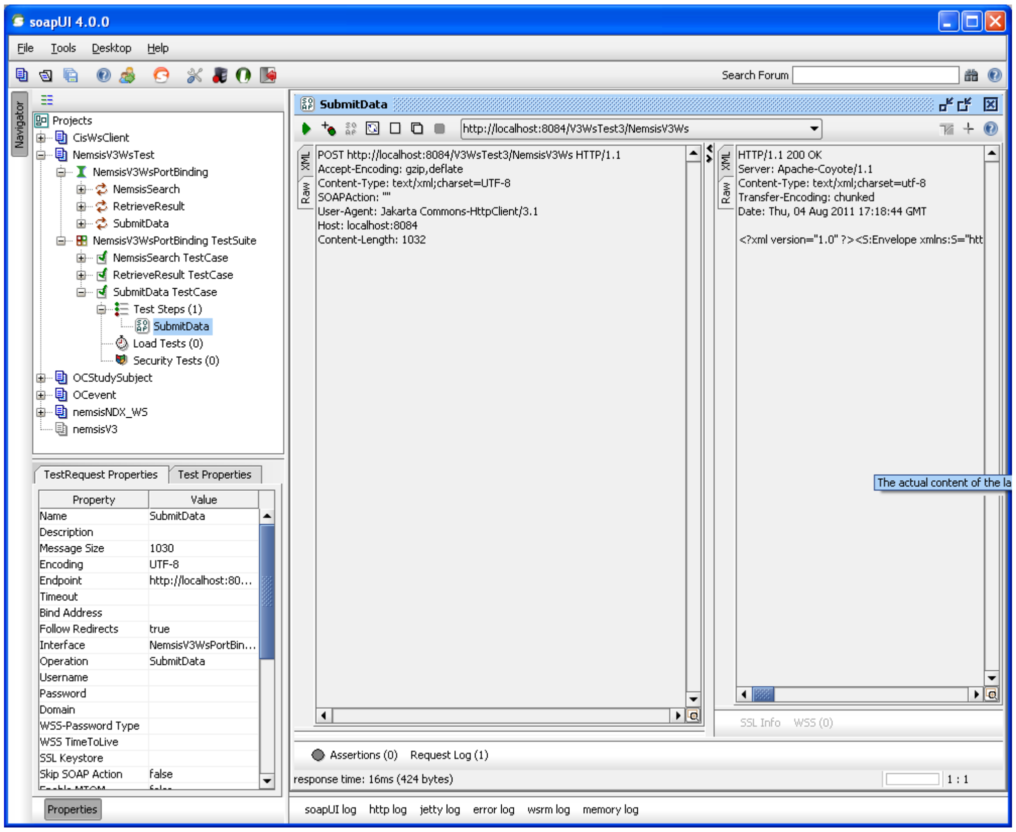


```
<soapenv:Envel
  <soapenv:He
  <soapenv:Bo
    <ws:Subm
      <subm
<?xml version=
<EMSDataSet xm
 xmlns:xsi="ht
   <Header>
     <dAger
     //Omit
   </Header>
</EMSDataSet>
 ]]></requestD
    </sub
   </ws:Sub
  </soapenv:B
</soapenv:Enve
```

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;charset=utf-8
Transfer-Encoding: chunked
Date: Thu, 04 Aug 2011 17:18:44 GMT

<?xml version="1.0" ?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Body><
```

Property	Value
Name	SubmitData
Description	
Message Size	1030
Encoding	UTF-8
Endpoint	http://localhost:80…
Timeout	
Bind Address	
Follow Redirects	true
Interface	NemsisV3WsPortBin…
Operation	SubmitData
Username	
Password	
Domain	
WSS-Password Type	
WSS TimeToLive	
SSL Keystore	
Skip SOAP Action	false

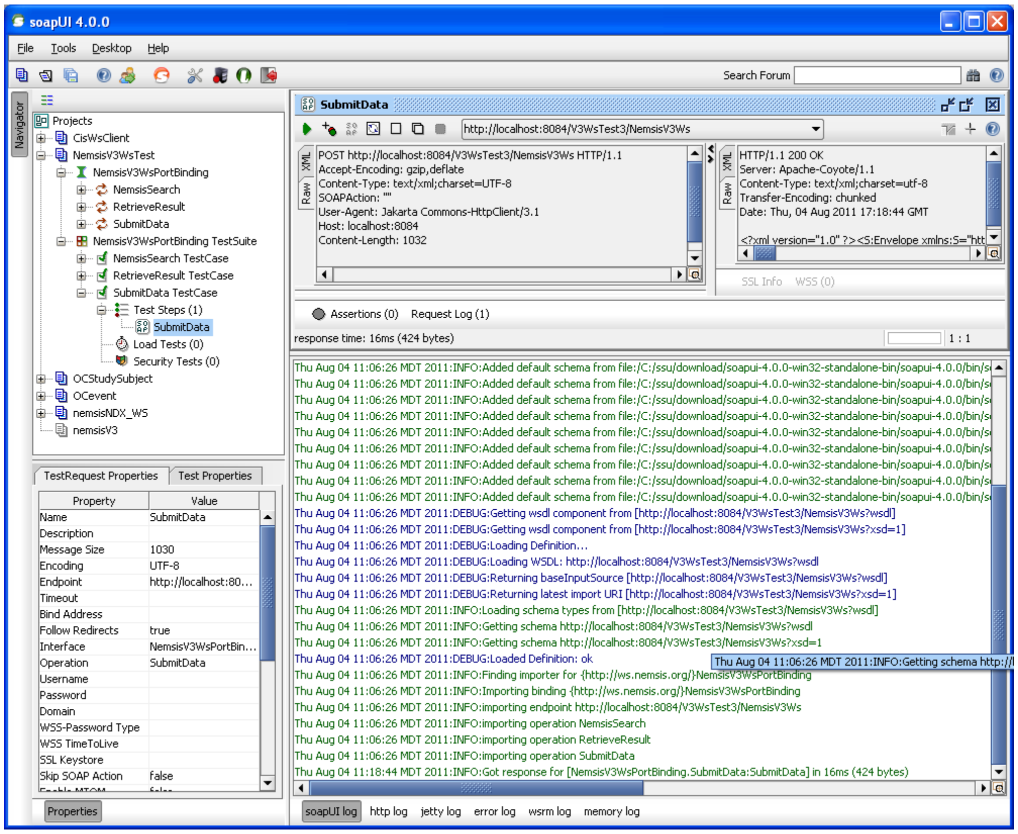


```
POST http://localhost:8084/V3WsTest3/NemsisV3Ws HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: ""
User-Agent: Jakarta Commons-HttpClient/3.1
Host: localhost:8084
Content-Length: 1032
```

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;charset=utf-8
Transfer-Encoding: chunked
Date: Thu, 04 Aug 2011 17:18:44 GMT

<?xml version="1.0" ?><S:Envelope xmlns:S="htt
```

Property	Value
Name	SubmitData
Description	
Message Size	1030
Encoding	UTF-8
Endpoint	http://localhost:80…
Timeout	
Bind Address	
Follow Redirects	true
Interface	NemsisV3WsPortBin…
Operation	SubmitData
Username	
Password	
Domain	
WSS-Password Type	
WSS TimeToLive	
SSL Keystore	
Skip SOAP Action	false



```
POST http://localhost:8084/V3WsTest3/NemsisV3Ws HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: ""
User-Agent: Jakarta Commons-HttpClient/3.1
Host: localhost:8084
Content-Length: 1032
```

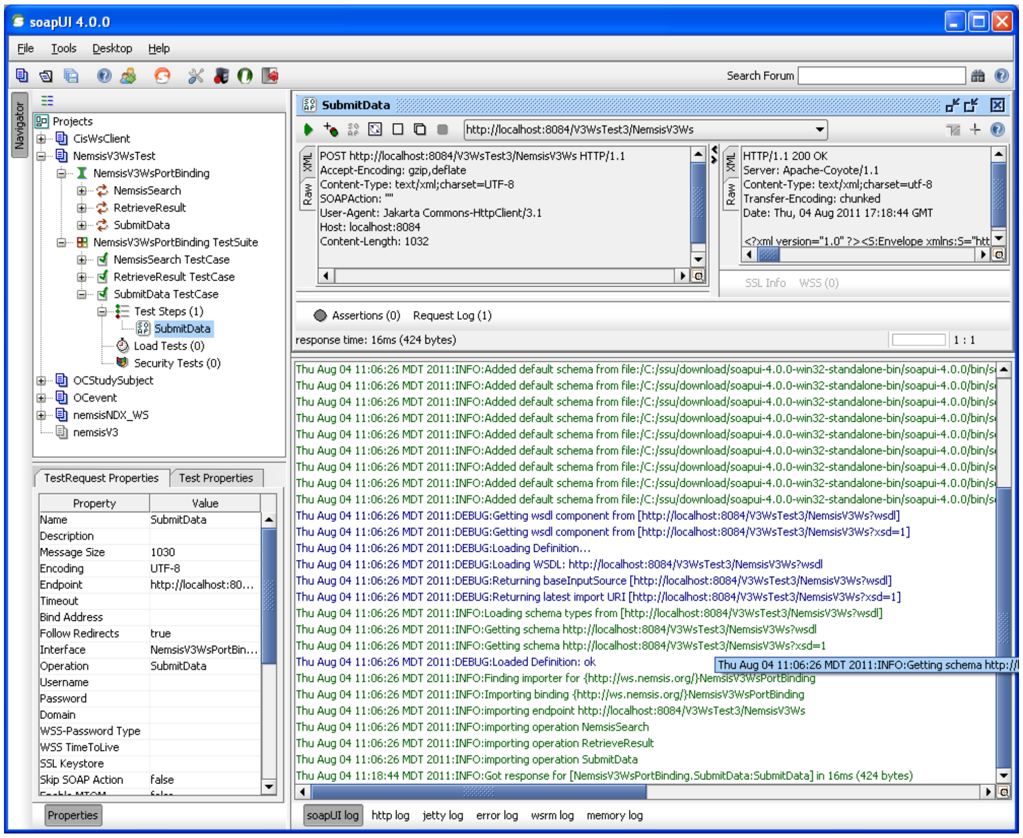
```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;charset=utf-8
Transfer-Encoding: chunked
Date: Thu, 04 Aug 2011 17:18:44 GMT

<?xml version="1.0" ?><S:Envelope xmlns:S="htt
```

response time: 16ms (424 bytes)

Property	Value
Name	SubmitData
Description	
Message Size	1030
Encoding	UTF-8
Endpoint	http://localhost:80...
Timeout	
Bind Address	
Follow Redirects	true
Interface	NemsisV3WsPortBin...
Operation	SubmitData
Username	
Password	
Domain	
WSS-Password Type	
WSS TimeToLive	
SSL Keystore	
Skip SOAP Action	false

```
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4.0.0-win32-standalone-bin/soapui-4.0.0/bin/s
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4.0.0-win32-standalone-bin/soapui-4.0.0/bin/s
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4.0.0-win32-standalone-bin/soapui-4.0.0/bin/s
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4.0.0-win32-standalone-bin/soapui-4.0.0/bin/s
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4.0.0-win32-standalone-bin/soapui-4.0.0/bin/s
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4.0.0-win32-standalone-bin/soapui-4.0.0/bin/s
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4.0.0-win32-standalone-bin/soapui-4.0.0/bin/s
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4.0.0-win32-standalone-bin/soapui-4.0.0/bin/s
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4.0.0-win32-standalone-bin/soapui-4.0.0/bin/s
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Getting wsdl component from [http://localhost:8084/V3WsTest3/NemsisV3Ws?wsdl]
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Getting wsdl component from [http://localhost:8084/V3WsTest3/NemsisV3Ws?xsd=1]
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Loading Definition...
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Loading WSDL: http://localhost:8084/V3WsTest3/NemsisV3Ws?wsdl
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Returning baseInputSource [http://localhost:8084/V3WsTest3/NemsisV3Ws?wsdl]
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Returning latest import URI [http://localhost:8084/V3WsTest3/NemsisV3Ws?xsd=1]
Thu Aug 04 11:06:26 MDT 2011:INFO:Loading schema types from [http://localhost:8084/V3WsTest3/NemsisV3Ws?wsdl]
Thu Aug 04 11:06:26 MDT 2011:INFO:Getting schema http://localhost:8084/V3WsTest3/NemsisV3Ws?wsdl
Thu Aug 04 11:06:26 MDT 2011:INFO:Getting schema http://localhost:8084/V3WsTest3/NemsisV3Ws?xsd=1
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Loaded Definition: ok
Thu Aug 04 11:06:26 MDT 2011:INFO:Finding importer for {http://ws.nemsis.org/}NemsisV3WsPortBinding
Thu Aug 04 11:06:26 MDT 2011:INFO:Importing binding {http://ws.nemsis.org/}NemsisV3WsPortBinding
Thu Aug 04 11:06:26 MDT 2011:INFO:importing endpoint http://localhost:8084/V3WsTest3/NemsisV3Ws
Thu Aug 04 11:06:26 MDT 2011:INFO:importing operation NemsisSearch
Thu Aug 04 11:06:26 MDT 2011:INFO:importing operation RetrieveResult
Thu Aug 04 11:06:26 MDT 2011:INFO:importing operation SubmitData
Thu Aug 04 11:18:44 MDT 2011:INFO:Got response for [NemsisV3WsPortBinding.SubmitData:SubmitData] in 16ms (424 bytes)
```



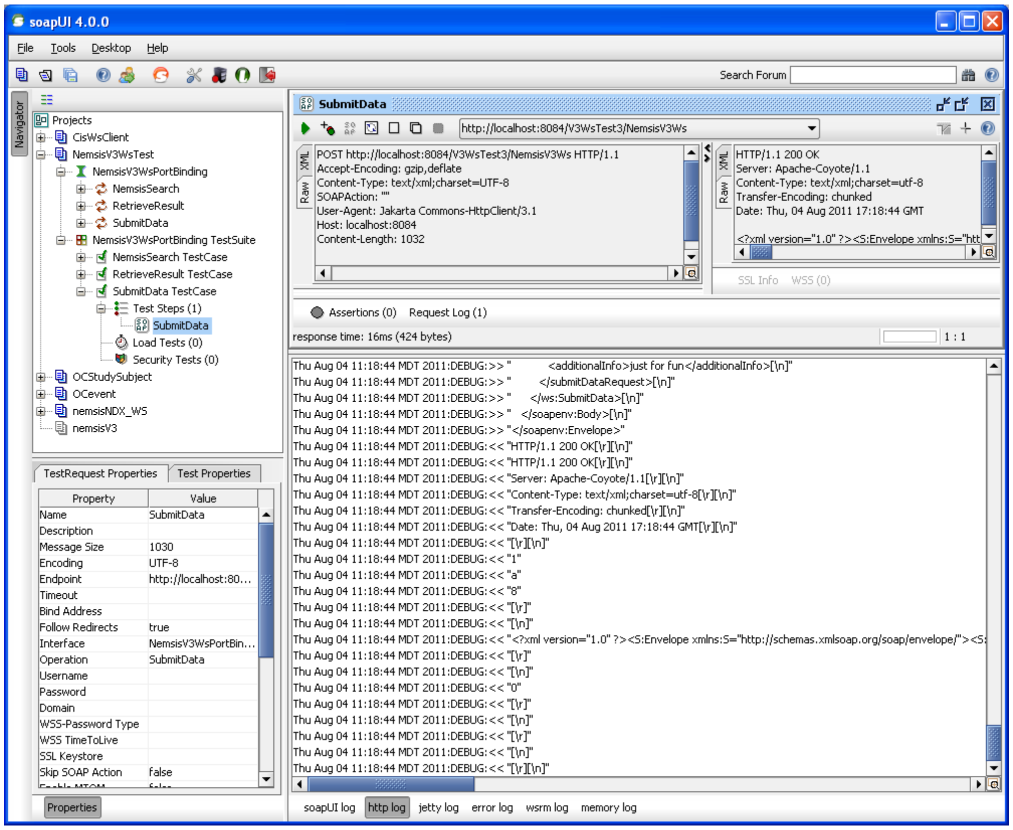
```
POST http://localhost:8084/V3WsTest3/NemsisV3Ws HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: ""
User-Agent: Jakarta Commons-HttpClient/3.1
Host: localhost:8084
Content-Length: 1032
```

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;charset=utf-8
Transfer-Encoding: chunked
Date: Thu, 04 Aug 2011 17:18:44 GMT

<?xml version="1.0" ?><S:Envelope xmlns:S="htt
```

response time: 16ms (424 bytes)

```
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Loaded Definition: ok
Thu Aug 04 11:06:26 MDT 2011:INFO:importing operation SubmitData
Thu Aug 04 11:18:44 MDT 2011:INFO:Got response for [NemsisV3WsPortBinding.SubmitData:SubmitData] in 16ms (424 bytes)
```



```
POST http://localhost:8084/V3WsTest3/NemsisV3Ws HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: ""
User-Agent: Jakarta Commons-HttpClient/3.1
Host: localhost:8084
Content-Length: 1032
```

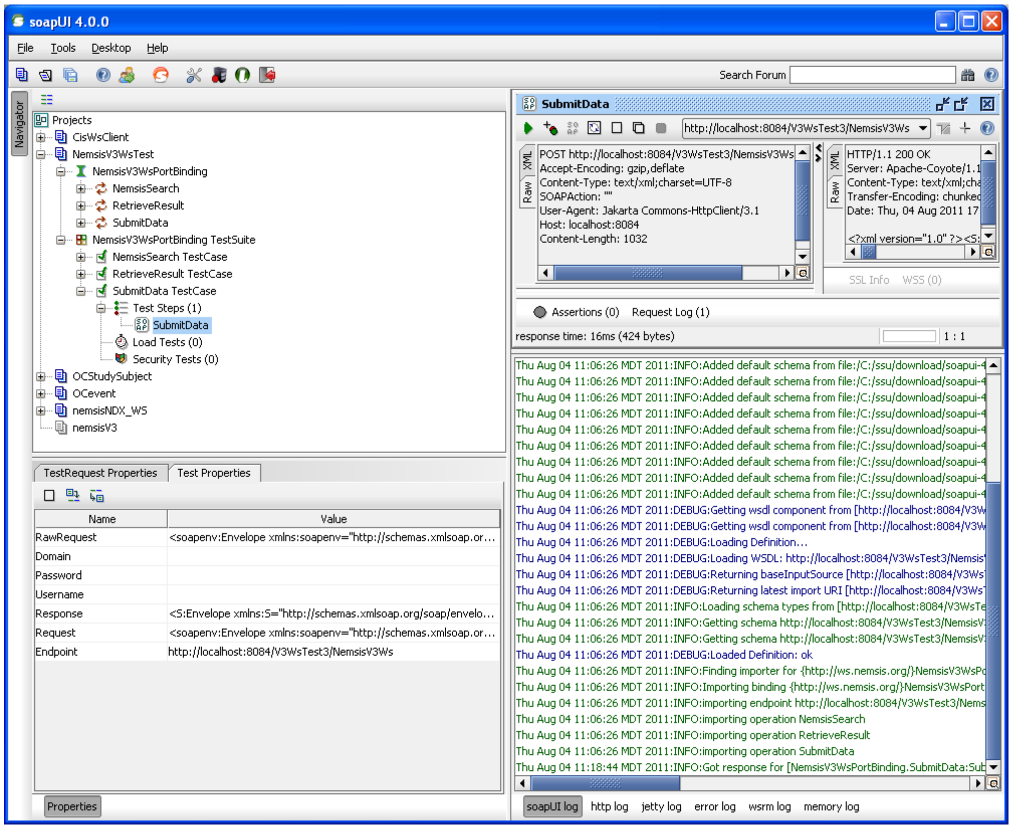
```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;charset=utf-8
Transfer-Encoding: chunked
Date: Thu, 04 Aug 2011 17:18:44 GMT

<?xml version="1.0" ?><S:Envelope xmlns:S="htt
```

response time: 16ms (424 bytes)   1 : 1

Property	Value
Name	SubmitData
Description	
Message Size	1030
Encoding	UTF-8
Endpoint	http://localhost:80…
Timeout	
Bind Address	
Follow Redirects	true
Interface	NemsisV3WsPortBin…
Operation	SubmitData
Username	
Password	
Domain	
WSS-Password Type	
WSS TimeToLive	
SSL Keystore	
Skip SOAP Action	false

```
Thu Aug 04 11:18:44 MDT 2011:DEBUG:>> "            <additionalInfo>just for fun</additionalInfo>[\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:>> "         </submitDataRequest>[\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:>> "      </ws:SubmitData>[\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:>> "   </soapenv:Body>[\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:>> "</soapenv:Envelope>"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "HTTP/1.1 200 OK[\r][\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "HTTP/1.1 200 OK[\r][\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "Server: Apache-Coyote/1.1[\r][\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "Content-Type: text/xml;charset=utf-8[\r][\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "Transfer-Encoding: chunked[\r][\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "Date: Thu, 04 Aug 2011 17:18:44 GMT[\r][\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "[\r][\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "1"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "a"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "8"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "[\r]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "[\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "<?xml version="1.0" ?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "[\r]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "[\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "0"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "[\r]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "[\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "[\r]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "[\n]"
Thu Aug 04 11:18:44 MDT 2011:DEBUG:<< "[\r][\n]"
```



```
POST http://localhost:8084/V3WsTest3/NemsisV3Ws
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: ""
User-Agent: Jakarta Commons-HttpClient/3.1
Host: localhost:8084
Content-Length: 1032
```

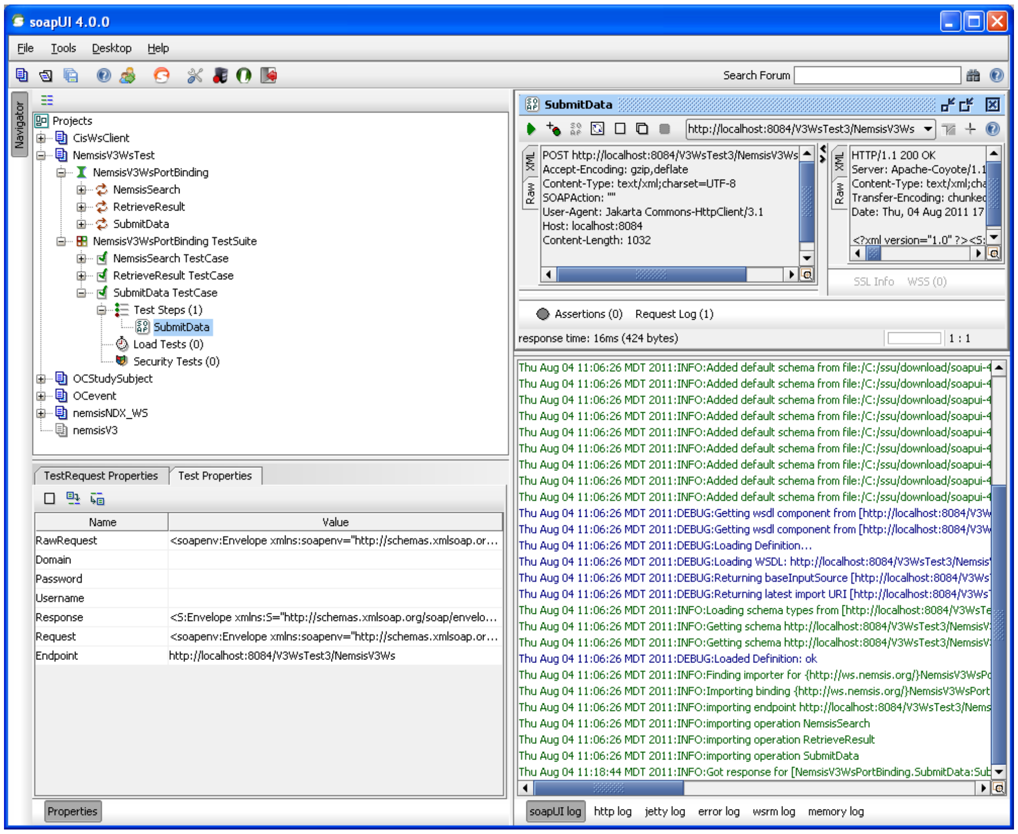
```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;cha
Transfer-Encoding: chunked
Date: Thu, 04 Aug 2011 17

<?xml version="1.0" ?><S:
```

response time: 16ms (424 bytes)   1 : 1

Name	Value
RawRequest	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.or…
Domain	
Password	
Username	
Response	<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelo…
Request	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.or…
Endpoint	http://localhost:8084/V3WsTest3/NemsisV3Ws

```
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Getting wsdl component from [http://localhost:8084/V3W
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Getting wsdl component from [http://localhost:8084/V3W
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Loading Definition...
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Loading WSDL: http://localhost:8084/V3WsTest3/Nemsis
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Returning baseInputSource [http://localhost:8084/V3Ws
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Returning latest import URI [http://localhost:8084/V3Ws
Thu Aug 04 11:06:26 MDT 2011:INFO:Loading schema types from [http://localhost:8084/V3WsTe
Thu Aug 04 11:06:26 MDT 2011:INFO:Getting schema http://localhost:8084/V3WsTest3/NemsisV
Thu Aug 04 11:06:26 MDT 2011:INFO:Getting schema http://localhost:8084/V3WsTest3/NemsisV
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Loaded Definition: ok
Thu Aug 04 11:06:26 MDT 2011:INFO:Finding importer for {http://ws.nemsis.org/}NemsisV3WsPo
Thu Aug 04 11:06:26 MDT 2011:INFO:Importing binding {http://ws.nemsis.org/}NemsisV3WsPort
Thu Aug 04 11:06:26 MDT 2011:INFO:importing endpoint http://localhost:8084/V3WsTest3/Nems
Thu Aug 04 11:06:26 MDT 2011:INFO:importing operation NemsisSearch
Thu Aug 04 11:06:26 MDT 2011:INFO:importing operation RetrieveResult
Thu Aug 04 11:06:26 MDT 2011:INFO:importing operation SubmitData
Thu Aug 04 11:18:44 MDT 2011:INFO:Got response for [NemsisV3WsPortBinding.SubmitData:Sub
```



```
Projects
  CisWsClient
  NemsisV3WsTest
    NemsisV3WsPortBinding
      NemsisSearch
      RetrieveResult
      SubmitData
    NemsisV3WsPortBinding TestSuite
      NemsisSearch TestCase
      RetrieveResult TestCase
      SubmitData TestCase
        Test Steps (1)
          SubmitData
        Load Tests (0)
        Security Tests (0)
  OCStudySubject
  OCevent
  nemsisNDX_WS
  nemsisV3
```

Name	Value
RawRequest	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.or…
Domain	
Password	
Username	
Response	<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelo…
Request	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.or…
Endpoint	http://localhost:8084/V3WsTest3/NemsisV3Ws

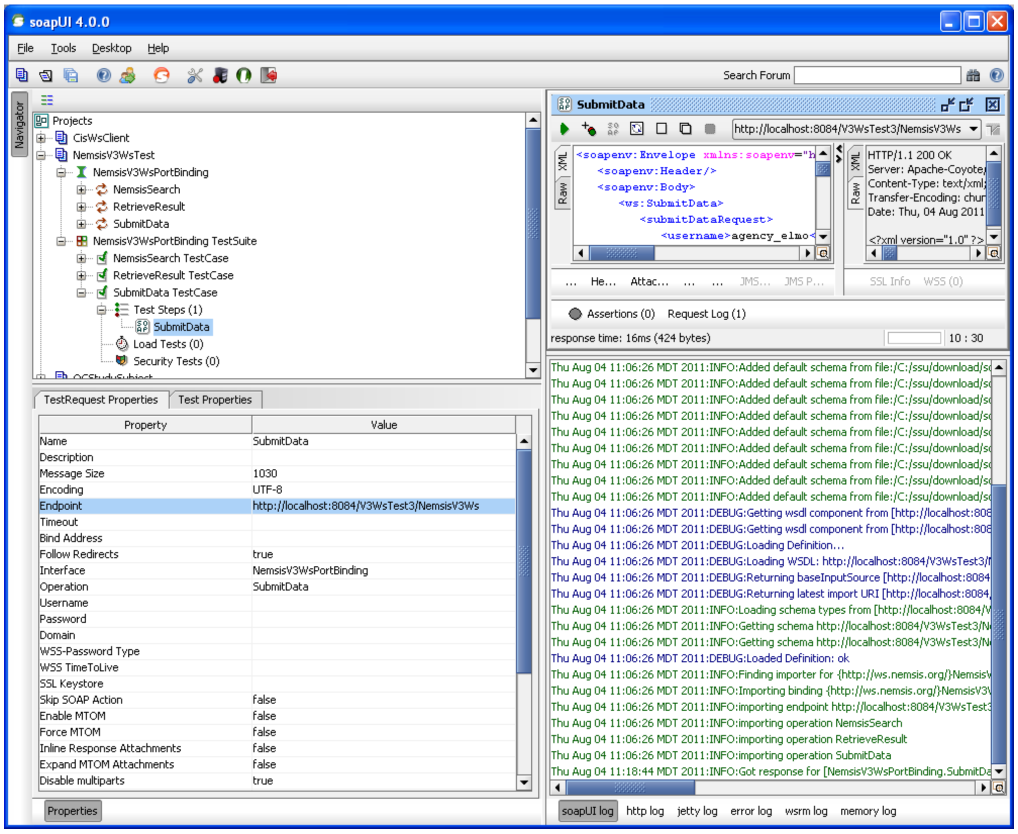
```
POST http://localhost:8084/V3WsTest3/NemsisV3Ws
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: ""
User-Agent: Jakarta Commons-HttpClient/3.1
Host: localhost:8084
Content-Length: 1032
```

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;cha
Transfer-Encoding: chunked
Date: Thu, 04 Aug 2011 17

<?xml version="1.0" ?><S:
```

response time: 16ms (424 bytes) 1 : 1

```
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/soapui-4
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Getting wsdl component from [http://localhost:8084/V3W
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Getting wsdl component from [http://localhost:8084/V3W
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Loading Definition...
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Loading WSDL: http://localhost:8084/V3WsTest3/Nemsis
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Returning baseInputSource [http://localhost:8084/V3Ws
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Returning latest import URI [http://localhost:8084/V3Ws
Thu Aug 04 11:06:26 MDT 2011:INFO:Loading schema types from [http://localhost:8084/V3WsTe
Thu Aug 04 11:06:26 MDT 2011:INFO:Getting schema http://localhost:8084/V3WsTest3/NemsisV
Thu Aug 04 11:06:26 MDT 2011:INFO:Getting schema http://localhost:8084/V3WsTest3/NemsisV
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Loaded Definition: ok
Thu Aug 04 11:06:26 MDT 2011:INFO:Finding importer for {http://ws.nemsis.org/}NemsisV3WsPo
Thu Aug 04 11:06:26 MDT 2011:INFO:Importing binding {http://ws.nemsis.org/}NemsisV3WsPort
Thu Aug 04 11:06:26 MDT 2011:INFO:importing endpoint http://localhost:8084/V3WsTest3/Nems
Thu Aug 04 11:06:26 MDT 2011:INFO:importing operation NemsisSearch
Thu Aug 04 11:06:26 MDT 2011:INFO:importing operation RetrieveResult
Thu Aug 04 11:06:26 MDT 2011:INFO:importing operation SubmitData
Thu Aug 04 11:18:44 MDT 2011:INFO:Got response for [NemsisV3WsPortBinding.SubmitData:Sub
```



```
Projects
  CisWsClient
  NemsisV3WsTest
    NemsisV3WsPortBinding
      NemsisSearch
      RetrieveResult
      SubmitData
    NemsisV3WsPortBinding TestSuite
      NemsisSearch TestCase
      RetrieveResult TestCase
      SubmitData TestCase
        Test Steps (1)
          SubmitData
        Load Tests (0)
        Security Tests (0)
  OCStudySubject
```

Property	Value
Name	SubmitData
Description	
Message Size	1030
Encoding	UTF-8
Endpoint	http://localhost:8084/V3WsTest3/NemsisV3Ws
Timeout	
Bind Address	
Follow Redirects	true
Interface	NemsisV3WsPortBinding
Operation	SubmitData
Username	
Password	
Domain	
WSS-Password Type	
WSS TimeToLive	
SSL Keystore	
Skip SOAP Action	false
Enable MTOM	false
Force MTOM	false
Inline Response Attachments	false
Expand MTOM Attachments	false
Disable multiparts	true

```
<soapenv:Envelope xmlns:soapenv="h
   <soapenv:Header/>
   <soapenv:Body>
      <ws:SubmitData>
         <submitDataRequest>
            <username>agency_elmo<
```

```
HTTP/1.1 200 OK
Server: Apache-Coyote/
Content-Type: text/xml;
Transfer-Encoding: chun
Date: Thu, 04 Aug 2011

<?xml version="1.0" ?>
```

response time: 16ms (424 bytes) 10 : 30

```
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/sc
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/sc
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/sc
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/sc
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/sc
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/sc
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/sc
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/sc
Thu Aug 04 11:06:26 MDT 2011:INFO:Added default schema from file:/C:/ssu/download/sc
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Getting wsdl component from [http://localhost:808
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Getting wsdl component from [http://localhost:808
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Loading Definition...
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Loading WSDL: http://localhost:8084/V3WsTest3/N
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Returning baseInputSource [http://localhost:8084
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Returning latest import URI [http://localhost:8084
Thu Aug 04 11:06:26 MDT 2011:INFO:Loading schema types from [http://localhost:8084/V
Thu Aug 04 11:06:26 MDT 2011:INFO:Getting schema http://localhost:8084/V3WsTest3/N
Thu Aug 04 11:06:26 MDT 2011:INFO:Getting schema http://localhost:8084/V3WsTest3/N
Thu Aug 04 11:06:26 MDT 2011:DEBUG:Loaded Definition: ok
Thu Aug 04 11:06:26 MDT 2011:INFO:Finding importer for {http://ws.nemsis.org/}Nemsis
Thu Aug 04 11:06:26 MDT 2011:INFO:Importing binding {http://ws.nemsis.org/}NemsisV3
Thu Aug 04 11:06:26 MDT 2011:INFO:importing endpoint http://localhost:8084/V3WsTest3
Thu Aug 04 11:06:26 MDT 2011:INFO:importing operation NemsisSearch
Thu Aug 04 11:06:26 MDT 2011:INFO:importing operation RetrieveResult
Thu Aug 04 11:06:26 MDT 2011:INFO:importing operation SubmitData
Thu Aug 04 11:18:44 MDT 2011:INFO:Got response for [NemsisV3WsPortBinding.SubmitDa
```

Reference:

1. WS-SecurityPolicy 1.2. OASIS Standard, 1 July 2007. http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-os.pdf
2. Web Services Security: SOAP Message Security 1.1. OASIS Standard Specification, 1 February 2006. http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SOAPMessageSecurity.pdf
3. Web Services Security: UsernameToken Profile 1.1. OASIS Standard Specification, 1 February 2006. http://www.oasis-open.org/committees/download.php/16782/wss-v1.1-spec-os-UsernameTokenProfile.pdf
4. SOAP Message Size Performance Considerations. http://www.redbooks.ibm.com/abstracts/redp4344.html