Installation and Usage of SBML 2.5

Installation

General - Information on the SBML Server can be accessed via the URL:

http://netlab.gmu.edu/OpenBML

SBML source code can be found in the SVN repository linked to by the OpenBML website. Version 2.5 is in the repository subdirectory SBMLWS/trunk. This is the most recent version of SBMLServer and is still in development. Previous stable versions of SBMLServer can be found in the repository subdirectory SBMLWS/branches.

The main enhancement for version 2.5 was the addition of a RESTful version of SBMLServer along with sample clients. This change requires configuring JBoss to use HornetQ for messaging. Another update was the processing of input MSDL and the use of MSDL information within orders and reports. The new repository items with version 2.5 are listed in bold below.

The main subdirectories of the repository are:

SBMLServer - Contains the SOAP based server code
SBMLServerREST – Contains the RESTful version of the server
SBMLServerLib – Contains the routines and setup files that are used by both the SOAP based and RESTful server
SBML_HOME - Contains scripts, configuration files, and schemas
SBMLHornetQREST – Contains the web application that provides the HornetQ REST interface

In addition, there is sample code on how to interact with the SOAP based and RESTful server. The subdirectories of the repository with this sample code are:

SOAP Based Sample Code:

SBMLClientLib – Contains a convenience class that can be used by java client applications to interact with the SOAP based server
SBMLClient - Contains a sample client application that uses the SBMLClientLib provided class
SBMLSubscriberHornetQ – Contains a sample PubSub subscriber application that uses the SBMLClientLib provided class (formally named SBMLSubscriber in version 2.4)
SBMLReplay – Contains a sample client application that replays the transactions in a replay log generated by the SOAP based SBMLServer (again by using the SBMLClientLib provided class)

RESTful Sample Code:

SBMLClientREST – Contains a sample client to the RESTful version of the SBML server
SBMLSubscriberJmsREST – Contains a sample PubSub subscriber application using the HornetQ JMS api
SBMLSubscriberHornetQREST – Contains a sample PubSub subscriber application using the HornetQ REST interface
SBMLReplayREST – Contains a sample client that replays the transactions in a replay log generated by SBMLServerREST

Other subdirectories in the repository are:

SBMLJars – Contains a build of SOAP based items: SBMLServer, SBMLClientLib, SBMLClient, SBMLReplay, and SBMLSubscriberHornetQ, as well as RESTful item: **SBMLServerREST**, **SBMLHornetQREST**, **SBMLClientREST**, **SBMLReplayREST**, **SBMLSubscriberJmsREST**, and **SBMLSubscriberHornetQREST**.
Translators – Contains source for the CSL (Condensed Scripting Language) compiler
SBMLSamples – Contains sample input files to test SBMLServer and SBMLServerREST

The repository also contains this document as well as a CSL programming guide for users that would like to write their own mapping scripts.

Note: As an alternative to setting up SBMLServer locally, the C4I center will installed SBML version 2.5 (both SOAP based and RESTful) on the machine bml-server3.c4i.gmu.edu for remote requests. Because of security concerns, bml-server3 will only be available on our VPN. Please contact help@c4i.gmu.edu to request VPN access. We will attempt but do not guarantee to keep this service up and current.

To build from source will require the installation of NetBeans with JBoss added as a server. The SOAP based items are built using Ant while the RESTful items are built using Maven. There are libraries included in several of the RESTful items that must be added to you to your local Maven repository before building.

INSTALLATION INSTRUCTIONS for SBML Server (both SOAP based and RESTful)

1) System and Software Requirements

These instructions assume that the installation will be done on a Linux system. We have successfully installed just the SOAP based SBMLServer on a Windows XP system in the past but do not test this type of installation with updates to the server. We have used both Fedora release 12 (Constantine) and CentOS 5.3 and 5.5 as systems for installation and testing but any recent Linux system should work. There are several pieces of software required for SBMLServer and SBMLServerREST. The following software list gives the version number tested and url for downloading. Other version numbers may work as well.

JBoss

version = 4.2.3

http://www.jboss.org/jbossas/downloads

Java SE

Version = 1.6.0_18

http://www.oracle.com/technetwork/java/javase/downloads/index.html

Ant

version = 1.7.1

http://archive.apache.org/dist/ant/binaries/

(needed for JBossWS installation and building from source)

JBossWS

version = 3.0.5

http://www.jboss.org/jbossws/downloads

MySQL Community Server

version = 14.14, distribution = 5.1.47

http://dev.mysql.com/downloads/mysql/5.1.html#downloads

HornetQ

version = 2.1.1, Final

http://www.jboss.org/hornetq/downloads

(also see http://community.jboss.org/thread/154727?tstart=0 to fix error in jca resource adaptor after installation)

RESTEasy

version = 2.0.1, GA

http://sourceforge.net/projects/resteasy/files/Resteasy%20JAX-RS/2.0.1.GA/

(does not need to be installed – required RESTEasy libraries are supplied with included builds)

HornetQ REST Interface

version = 1.0-beta-3

http://sourceforge.net/projects/resteasy/files/HornetQ%20REST%20Interface/

(war file that implements the interface is provided in the repository and must be copied to the default-with-hornetq deploy directory – interface only needed for subscriptions using the HornetQ REST Interface as illustrated in the SBMLSubscriberHornetQREST sample code)

2) Build Database (jc3iedm_sbml) used by both SBML servers

Obtain the database initialization scripts, jc3iedm_init.sql and MSDL.sql, from the SBML repository under SBMLServerLib/setup.

Assuming that the database does not already exist, build the database by entering:

```
mysql –u root < jc3iedm_init.sql
mysql –u root < MSDL.sql
```

Note: This assumes the database root password is blank. If access to database is needed from another system set up permissions for remote access.

3) Set up SBML_HOME

Copy the SBML_HOME Directory onto the machine that will run SBMLServer and/or SBMLServerREST. Make sure that jboss has access privileges to SBML_HOME since the CSL compiler output will be written to this directory. Note that you will need to have two copies of SBML_HOME if you plan to deploy both SBMLServer and SBMLServerREST.

4) Set Environment Variables

For example (using the bash shell):

```
export JAVA_HOME=.../jdk
export JBOSS_HOME=.../jboss-4.2.3.GA
```

5) Set up sbml.properties and/or sbmlrest.properties

Located in SBMLServer/setup is the sample properties file, sbml.properties. There is a corresponding properties file in SBMLServerREST/ear/setup named, sbmlrest.properties. These property files are read during initialization of SBMLServer and SBMLServerREST and should be copied to $JBOSS_HOME/server/default-with-hornetq/conf before deploying SBMLServer or SBMLServerREST.

The following shows a typical properties file:

```
##
## Configuration file for SBML web service
##
SBMLHome=/yourdir/SBML_HOME

# SQL Database
dbActive=y
dbURL=jdbc:mysql://dbhost/jc3iedm_sbml
dbUname=db_acct_name
dbPwd=db_pwd

pubsubActive=y
```

The following properties will need to be set for this environment:

SBMLHome – points to the location of the SBML_HOME directory copied in a previous step. Note that there must be separate SBML_HOME directories for SBMLServer and SBMLServerREST.

dbURL – Contains the name of the host where the database server is located.
dbUname – The name of a MySQL account configured with rights on the database
dbPwd – The database password for dbUname
Note – The MySQL root account cannot be used remotely.

There are many other properties that can be set through the properties file. The following table summarizes them:

Property	**Description**	**Valid values**	**Req'd**	**Default**
SBMLHome	Absolute path to the SBML_HOME directory that contains scripts (csl and xml), scripting schema, namespace mappings, and IBML schemas		Yes	none
nsMappingFileName	Filename of namespace prefix to URI mapping file within SBMLHome		No	NSMapping.xml
schemaDirectory	Directory that contains IBML schemas within SBMLHome		No	schema
CSLDirectory	Directory that contains CSL scripts within SBMLHome		No	CSL_Scripts
replayLog	Generation of replay log	[yn]	No	n
Validate	Validation of input BML	[yn]	No	n
testMode	Testing mode for scripting changes -- csl scripts are re-compiled with every request	[yn]	No	n
riActive[x]	Requests to the RI available	[yn]	No	n
jc3Host[x]	IP address of the RI		Yes[+]	
jc3Port[x]	Port for SBMLServer to talk to the RI		Yes[+]	
jc3CbPort[x]	Call back port from the RI to SBMLServer		Yes[+]	

riLocal[x]	Use a local interface to the RI rather than JBoss remoting	[yn]	No	n
dbActive	Requests to the local MSG database available	[yn]	No	y
dbURL	URL of the MSG database		Yes[*]	none
dbUname	MySQL user name		Yes[*]	none
dbPwd	MySQL password (for the dbUname specified)		Yes[*]	none
pubsubActive	Subscription to published reports is available	[yn]	No	n

(+) jc3Host, jc3Port, and jc3CbPort are only required properties when riActive=y

(*) dbURL, dbUname, and dbPwd are only required properties when dbActive=y

(x) not available with publically available version of SBMLServer

Note: Either riActive or dbActive (or both) must be set to y for SBMLServer to process requests.

6) <u>Server Deployment</u>

a) Copy hornetq-jms.xml from the repository SBMLServerLib/setup to $JBOSS_HOME/server/default-with-hornetq/deploy/hornetq.sar. If you have made local updates to hornetq-jms.xml, you may have to merge the two files. The SBML servers are dependent on the definition of the topic SBMLTopic and entry name of the NettyConnection factory defined within the repository version of hornetq-jms.xml.
b) Copy hornetq-configuration.xml from the repository SBMLServerLib/setup to $JBOSS_HOME/server/default-with-hornetq/deploy/hornetq.sar. Edit the configuration file so that the netty connector has its host set to machine that has JBoss installed. If you have made local updates to hornetq-jms.xml, you may have to merge the two files. The provided hornetq-configuration.xml file disables security roles and provides a possible password for cluster administration.
c) Copy mysql JDBC connector (mysql-connector-java-5.1.6-bin.jar –found in SBMLServerLib/setup) to $JBOSS_HOME/server/default-with-hornetq/lib
d) Copy jboss-log4j.xml from SBMLServerLib/setup to $JBOSS_HOME/server/default-with-hornetq/conf. If you have made updates to the log4j configuration file, you may have to merge the copy supplied with your local copy.

e) Copy jaxrs-api-2.0.1.GA.jar from SBMLServerLib/setup to $JBOSS_HOME/server/default-with-hornetq/lib.

7) SBMLServer deployment

a) For the SOAP based version of the server, copy SBMLServer.jar from SBMLJars (or build another copy of SBMLServer.jar from source) to $JBOSS_HOME/server/default-with-hornetq/deploy
b) Copy sbml.properties with any local configuration updates from SBMLServer/setup to $JBOSS_HOME/server/default-with-hornetq/conf

8) SBMLServerREST deployment

a) For the RESTful version of the server, copy SBMLServerREST.ear from SBMLJars (or build from source) to $JBOSS_HOME/server/default-with-hornetq/deploy
b) Copy sbmlrest.properties with any local configuration updates from SBMLServerREST/ear/setup to $JBOSS_HOME/server/default-with-hornetq/conf

9) SBMLHornetQREST war deployment

If you would like to subscribe to the SBMLTopic used by SBMLServer and SBMLServerREST to publish reports using the HornetQ REST interface, you must deploy SBMLHornetQREST.war which implements this interface. The repository subdirectory SBMLSubscriberHornetQREST is an example of how to use the HornetQ REST interface to subscribe using only an HTTP client library. Subscribing in this way is available to a client written in any language that has access to an HTTP client library.

10) Start jboss

Execute $JBOSS_HOME/bin/run.sh -b 0.0.0.0 –c default-with-hornetq

11) **Use sample code to access both versions of the SBML server. See the next Usage section for more information on accessing SBMLServer and SBMLServerREST services.**

Testing and Usage

After SBMLServer and SBMLServerREST are installed and started, this section describes how to test and use these web services

Usage – Included with the SBML repository in the `SBMLJars/SBMLClient/dist` directory is a sample test client `SBMLClient.jar` to access the SOAP based SBML server. In `SBMLJars/SBMLClientREST`

is a sample test client `target/SBMLClientREST-1.0-jar-with-dependencies.jar` to access the RESTful SBML server. In `SBMLSamples/input`, there is a set of IBML and C_BML test input files. In `SBMLSamples/output` is the corresponding set of expected output.

These sample test clients take four input parameters at execution: location of SBML server, input bml xml, submitterID, and domain. To run either of these clients, enter the following command:

```
java –jar <client jar> <ip of SBMLServer> <input xml> <submitterID> <domain>
```

The <input xml> is expected to adhere to the schemas provided in `$SBML_HOME/schema`.

The <submitterID> is not currently defined but will somehow identify the person/organization that is submitting the input bml for processing. Any string here will currently work.

For <domain> the options are IBML, CBML, or MSDL (case insensitive). For the publically available version of SBML server, the RI domain is disabled. The C_BML option is based on light mappings in the January 2011 Trial Use package and only currently implements a push and pull of a <Task>.

Also included in the `SBMLJars` directory is a build of the sample subscriber `SBMLSubscriberHornetQ` to the SOAP based SBML server. Under `SBMLSubscriberJmsREST` and `SBMLSubscriberHornetQREST` are builds of two sample subscribers to the RESTful SBML server. `SBMLSubscriberJmsREST` uses the HornetQ JMS api to create subscriptions while `SBMLSubscriberHornetQREST` uses a HTTP client library and the HornetQ REST Interface to create subscriptions. `SBMLSubscriberJmsREST` could only be written in Java while `SBMLSubscriberHornetQREST` could be written in any language that has access to a HTTP client library. This sample codes take one input parameter which specifies the location of SBMLServer or SBMLServerREST. You would run the subscriber code with the following command:

```
java –jar <subscriber jar> <ip-addr of SBMLServer>
```

Testing - As mentioned before, in the `SBMLSamples/input` directory is a set of sample input xml (using IBML and C_BML) that can be used for testing and illustrate the type of requests that can be made of SBMLServer and SBMLServerREST.

Test XML	Description
MSDL/initial_MSDL.xml	Initial MSDL info
MSDL/initial_short_MSDL.xml	Initial MSDL info
MSDL/list.xml	List out currently defined scenario names
MSDL/finalize.xml	Publish all MSDL
MSDL/addEquipment.xml	Add additional equipment

MSDL/addUnit.xml	Add an additional unit
MSDL/query.xml	Pull all MSDL
IBML/NewUnitTypeMSDL.xml	Set up unit types for MSDL
IBML/NewUnitType.xml	Add new unit types
IBML/NewUnit.xml	Add new units
IBML/ListWhoAll.xml	Retrieve information on all units
IBML/ListWho.xml	Retrieve information on a particular unit
IBML_09/SampleOrderPush.xml	Order in older IBML format
IBML_09/SampleOrderPull.xml	Pull of order
IBML_09/SBML_GSR_09Push.xml	General Status Report in older IBML format
IBML_09/SBML_TSR_09Push.xml	Task Status Report in older IBML format
C_BML/SampleIBMLOrderPush1.xml	Push an IBML order with C_BML components
C_BML/SampleIBMLOrderPull1.xml	Pull an IBML order with C_BML components
C_BML/Sample_NewWhere.xml	Create a location
C_BML/Sample_WherePull.xml	Retrieve the newly created location
C_BML/SampleOpOrdPush1.xml	Push a NATO OpOrd with C_BML components
C_BML/SampleOpOrdPull1.xml	Pull a NATO OpOrd with C_BML components
C_BML/SampleReportPush_CBML_IBML09.xml	Report in older IBML format w/C_BML parts

Note: There is not a sample IBML air order included. Differentiating between different types of orders for past versions of SBMLServer required adding additional columns to JC3IEDM tables. Script changes that do not rely on these extra columns are not yet complete.

After initializing the MSG database, you can test your installation by running each of these sample input xml files in the order presented in the above table and comparing to the expected output which is located in `SBMLSamples/output`. A python test script is provided in `SBMLSamples/scripts/run_all_tests.py` to make this testing more automatic. Note: this python script has run successfully on Linux with python 2.6.2 but is untested on other operating systems. To run this script, enter: `python run_all_tests.py <location of SBML client jar>` at the shell prompt. After the script has finished, please check the file `diffResults.txt` to make sure there were no differences with the expected output – an empty file means that the tests ran successfully.

When executing the `SBMLSubscriberHornetQ` subscriber code, expect output similar to the following:

```
(OUTPUT shown is old – TODO:  need to update)

Begin Subscriber
Connecting to: localhost
InitialContext created
```

```
Subscribing to allPSR with search = //TypeOfReport[.
PositionStatusReport']
Subscribing to allGSR with search = //TypeOfReport[. =
'GeneralStatusReport']
Subscribing to allSIMCI with search = /*[contains(name(),'REP')]
Waiting for messages
```

When the subscriber detects a new report, expect the output to be similar to:

(OUTPUT shown is old – TODO: need to update)

```
   1  2011/41/23 10:41:51  BMLType:IBML Sent:Wed Mar 23 10:41:51 EDT 2011
MsgSelector:allPSR
   RootNode:BMLREPORT Date: FirstReportID:
ReportType:PositionStatusReport  Reporter:1-675  Executer:1-675
   2  2011/41/23 10:41:51  BMLType:IBML Sent:Wed Mar 23 10:41:51 EDT 2011
MsgSelector:allSIMCI
   RootNode:BMLREPORT Date: FirstReportID:
ReportType:PositionStatusReport  Reporter:1-675  Executer:1-675
```

The output for `SBMLSubscriberJmsREST` and `SBMLSubscriberHornetQREST` is very similar. Note that there is just one topic named SBMLTopic that is currently shared between SBMLServer and SBMLServerREST. Any subscriber will get messages published from either SBML server. In the future, this could change.

On-line Documentation for RESTEasy and HornetQ

HornetQ 2.1 Quick Start Guide
http://hornetq.sourceforge.net/docs/hornetq-2.1.0.Final/quickstart-guide/en/html_single/index.html

HornetQ 2.1 User Manual
http://hornetq.sourceforge.net/docs/hornetq-2.1.0.Final/user-manual/en/html_single/index.html

RESTEasy JAX_RS: RESTful Web Services for Java 2.0.1.GA
http://docs.jboss.org/resteasy/docs/2.0.0.GA/userguide/html_single/index.html

HornetQ REST Interface 1.0-beta-3
http://docs.jboss.org/resteasy/hornetq-rest/1.0-beta-3/userguide/html_single/index.html