Testing SBML 2.3 and Usage

After SBMLServer is installed and started, this document describes how to test and use this web service.

Usage – Included with the SBML repository in the SBMLJars directory is a sample test client SBMLClient.jar. This sample test client takes three input parameters at execution: location of SBMLServer, input bml xml, and domain/database. Run this client with the following command:

    java -jar SBMLClient.jar <ip-addr of SBMLServer> <input xml> <domain>

The <input xml> must adhere to the schema provided in $SBML_HOME/schema. To have SBMLServer validate the <input xml> against the schema, add the property validate=y to the sbml.properties file installed in $JBOSS/server/default/conf.

For <domain/database> the options are:

<table>
<thead>
<tr>
<th>&lt;domain&gt;</th>
<th>Name of script in $SBML_HOME</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order</td>
<td>OrderServices.xml</td>
<td>MSG</td>
</tr>
<tr>
<td>Order_RI</td>
<td>OrderServices.xml</td>
<td>RI</td>
</tr>
<tr>
<td>Report</td>
<td>ReportServices.xml</td>
<td>MSG</td>
</tr>
<tr>
<td>Report_RI</td>
<td>ReportServices.xml</td>
<td>RI</td>
</tr>
<tr>
<td>SIMCIReport</td>
<td>SevenReports.xml</td>
<td>MSG</td>
</tr>
<tr>
<td>SIMCIReport_RI</td>
<td>SevenReports.xml</td>
<td>RI</td>
</tr>
</tbody>
</table>

To use the MSG database, the db_active property in sbml.properties must be turned on. To use the RI database, the ri_active property must be enabled. SBMLServer must have at least one of these properties enabled to run. Both properties may be turned on to allow access to either database with one copy of SBMLServer.

Also included in the SBMLJars directory is a sample subscriber SBMLSubscriber.jar. This sample code takes one input parameter which specifies the location of SBMLServer. You would run the subscriber code with the following command:

    java -jar SBMLSubscriber.jar <ip-addr of SBMLServer>

Testing - In the SBMLClient/Sample_Input directory is a set of sample input xml that can be used for testing and illustrate the type of requests that can be made of SBMLServer.

<table>
<thead>
<tr>
<th>Test XML</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NewWhoPush.xml</td>
<td>Add new units</td>
</tr>
</tbody>
</table>
After initializing the MSG database, the following successful executions of SBMLClient on the same machine as SBMLServer will test your installation:

Adding units to be used by later in orders or reports:

```java
java -jar SBMLClient.jar localhost NewWhoPush.xml Order
```

Retrieving information on the units just added:

```java
java -jar SBMLClient.jar localhost ListWhoAll.xml Order
```

Retrieving information on one particular unit (abc5):

```java
java -jar SBMLClient.jar localhost ListWho.xml Order
```

Pushing a sample ground order:

```java
java -jar SBMLClient.jar localhost SampleOrderPush1.xml Order
```

Retrieving that same ground order just pushed:

```java
java -jar SBMLClient.jar localhost SampleOrderPull.xml Order
```

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

Adding a variety of reports:

```java
java -jar SBMLClient.jar localhost SamplePositionStatusReportPush.xml SIMCIReport
```
Retrieving a list of all the latest reports:

```
java -jarSBMLClient.jar localhost SampleGeneralStatusReportPush.xml SIMCIReport
java -jarSBMLClient.jar localhost SampleSpotReportPush.xml SIMCIReport
java -jarSBMLClient.jar localhost SampleBridgeReportPush.xml SIMCIReport
java -jarSBMLClient.jar localhost SampleMinefieldReportPush.xml SIMCIReport
java -jarSBMLClient.jar localhost SampleTrackReportPush.xml SIMCIReport
```

Retrieving a Bridge report:

```
java -jarSBMLClient.jar localhost ReportPull.xml SIMCIReport
```

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

```
Begin Subscriber
Connecting to: localhost
InitialContext created
Waiting for messages

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

```
1 2010/04/07 14:01:03 Topic: sbmlOpStatus Date: AT 20070101000000.000
   FirstReportID:34780 numReports: 1 Reporter: Executer:
2 2010/04/07 14:01:03 Topic: allSIMCI Date: AT 20070101000000.000
   FirstReportID:34780 numReports: 1 Reporter: Executer:
```

In this case, the one position report was picked up under two possible topics: sbmlOpStatus and allSIMCI