QAT Global
Code Sample

SysMgmt
Web Service

Contents

1 Introduction

As part of QAT training, we are developing several code samples. These code samples illustrate various technical as well as business aspects. These samples will help the developer get a quick understanding of the technology and framework. This will considerably reduce the learning time. The samples can be used as a reference to develop the application.

1.1 Pre-requisites

In order to extract the maximum benefit out the samples, its expected that the developers have

- Completed Basic Java training
- Completed Basic Spring training
- Completed Development environment setup – JDK , Eclipse, Oracle/postgreSQL, Tomcat/JBoss SOAPUI etc.
- Acquire the sysmgmt-multimodule Maven project from SVN or local file server or tech-lead/instructor. If downloading from SVN see Appendix A – Project Checkout from SVN for instructions.
- **Ask your instructor whether you will be using postgreSQL or Oracle as your DB**

2 Setting up the Database

The instructions below describe the necessary steps for setting up your local database prior to testing the sysmgmt sample. The instructions show setup for PostgreSQL and Oracle.

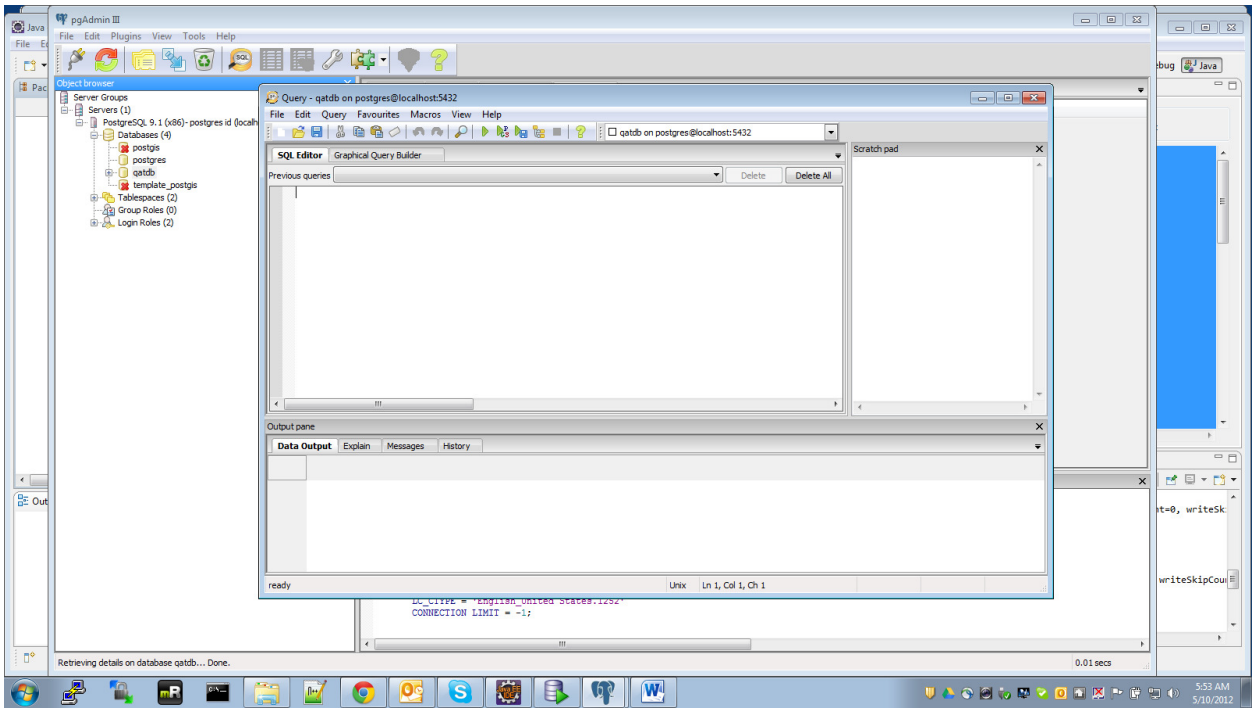
2.1 Setting up PostgreSQL DB

2.1.1 Locate and open the file in Notepad++ `sysmgmtsampledb-postgres.sql` file in the following directory.

`C:\QATEclipseWorkSpace\sysmgmt-multimodule\sysmgmt-implementation\src\test\resources\setup`

Do a select all and copy……

2.1.2 Open pgAdmin III and connect to localhost and select the “qatdb” under databases and click the SQL button from the menu



3. Copy the contents of the **`sysmgmtsampledb-postgres.sql`** file into the window and run the statements. Verify the tables are there. Database setup is complete.

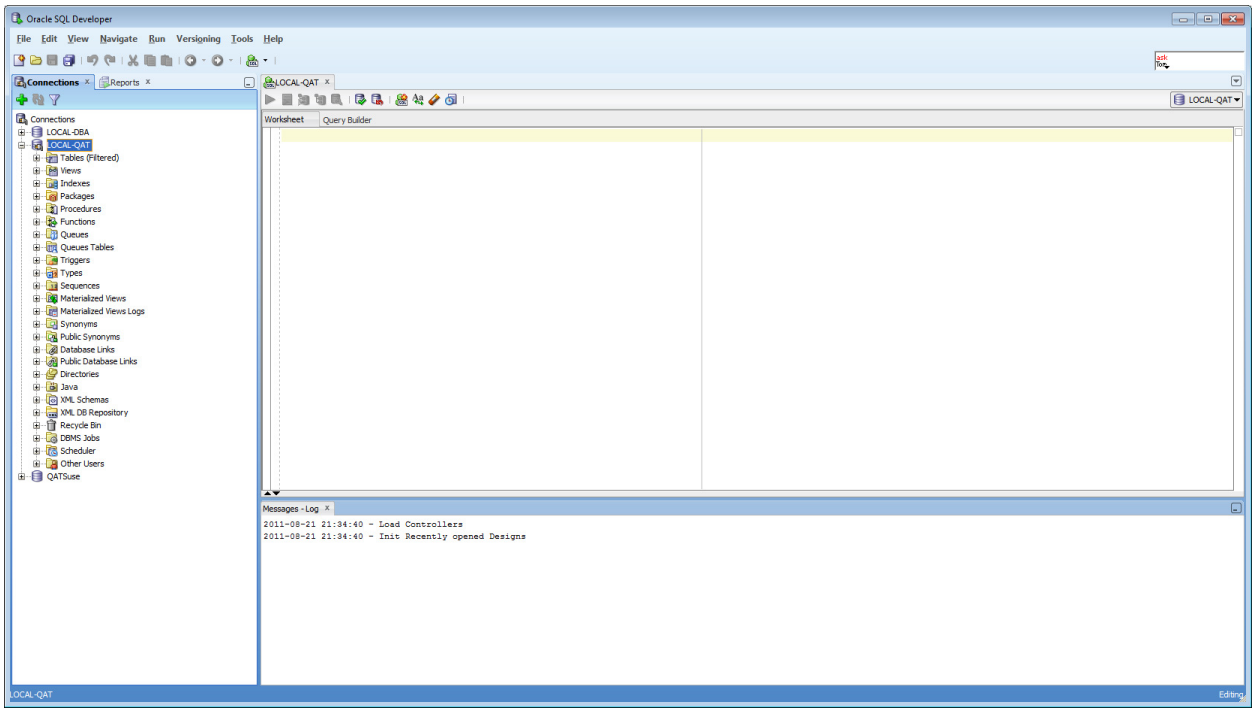
2.2 Setting up ORACLE DB

2.2.1 Locate and open the file in Notepad++ `sysmgmtsampledb-oracle.sql` file in the following directory.

`C:\QATEclipseWorkSpace\sysmgmt-multimodule\sysmgmt-implementation\src\test\resources\setup`

Do a select all and copy……

2.2.2 Open SQL Developer and connect to localhost



2.2.3 Copy the contents of the **`sysmgmtsampledb-oracle.sql`** file into the window and run the statements. Verify the tables are there. Database setup is complete.

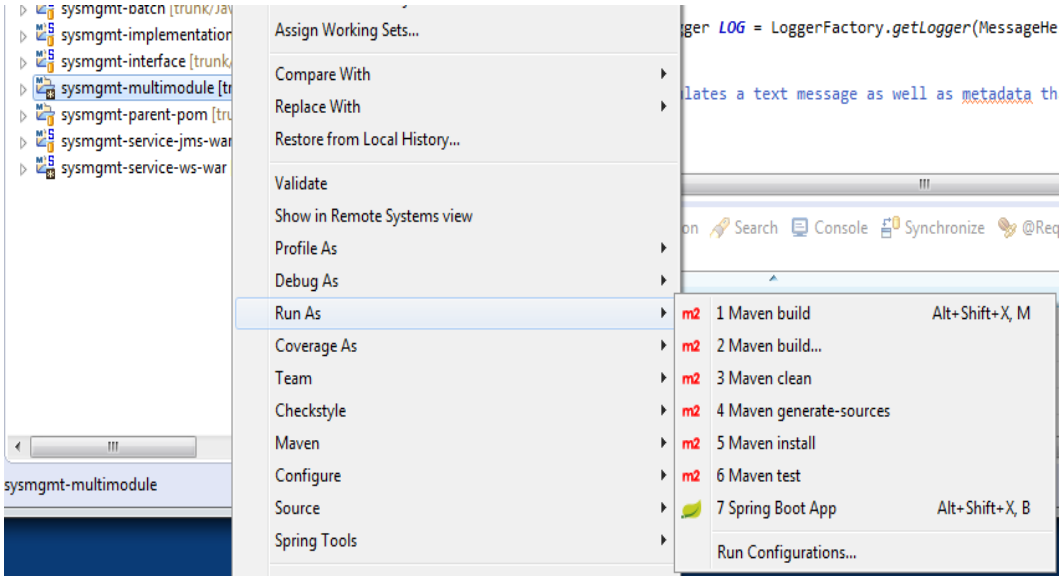
3 Building the Project

3.1 Network Access and Artifactory Access

Note when you first open the projects you need to have network access and access to the Artifactory repository containing the framework JAR files.

3.1.1 Building and publishing for the first time

To build the modules the first time simply right-click on the sysmgmt-multimodule in the Eclipse Package Explorer. Select “Run As > Maven Install”.



Note: This will also run Junit tests. Some of these are dependent on the database setup described in section 2. By default, the projects use PostgreSQL If you are using Oracle, look for the `@ActiveProfiles("postgres")` annotations in the BAR tests in the sysmgmt-implementation module and change them to `@ActiveProfiles("oracle")`.

Also note that the Oracle JDBC driver requires a license because there is no legal Maven site for a download. You will need to get an ojdbc6.jar and add it to your local Maven repository using something similar to the following Maven install command:

```
mvn install:install-file -Dfile=C:\SomeFolder\ojdbc6.jar -DgroupId=com.oracle -
DartifactId=ojdbc6 -Dversion=11.2.0 -Dpackaging=jar
```

Then locate the commented out dependency for the ojdbc6.jar in the pom.xml for the sysmgmt-implementation module and uncomment it specifying, of course, the version number which matches the one supplied in the mvn install command.

When the Maven Install completes, you should see messages similar to the following at the end of the Console log:

```
[INFO] Reactor Summary:
[INFO]
[INFO] sysmgmt-parent-pom ............................... SUCCESS [  0.562 s]
[INFO] sysmgmt-interface ................................ SUCCESS [  1.124 s]
[INFO] sysmgmt-implementation ........................... SUCCESS [  4.484 s]
[INFO] sysmgmt-batch .................................... SUCCESS [ 18.496 s]
[INFO] sysmgmt-service-ws-war ........................... SUCCESS [  4.028 s]
[INFO] sysmgmt-multimodule .............................. SUCCESS [  0.016 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 28.836 s
```

```
[INFO] Finished at: 2015-05-21T10:28:19-06:00
[INFO] Final Memory: 10M/83M
[INFO] ------------------------------------------------------------------------
```

3.1.2 Subsequent changes and building

When you make a source change make sure you right-click on the module where the change is made and select “Run As > Maven Install” to produce an up to date JAR file.

3.2 Deploying the default sysmgmt-web war

3.2.1 Building the projects using Eclipse

If you followed the instructions in Section 3.1 above, your WAR file should be ready to deploy.

3.2.2 Building the WAR file

1. Locate the sysmgmt-service-ws-war module.
2. Right-click on this module and “Run As” followed by the “Maven Install” menu item. This will execute the Maven build script and produce a WAR file under the “target” folder under the sysmgmt-multimodule/sysmgmt-service-ws-war folder.

3.2.3 Deploy to your web server

1. Copy the qat-sysmgmt-sample-ws-service.war from the “target” folder to either the tomcat or jboss hot deploy directories
2. Wait 30 seconds and then proceed to the testing section.

4 Testing

4.1 Testing via SOAP-UI

4.1.1 Double click on SOAP-UI icon. If the console shows the message JAVA_HOME is not set, please set the JAVA_HOME.

4.1.2 To create a new project, press Ctrl+N or File->New Soap UI project. Or import "QAT-SysMgmt-CXF-WS-soapui-project.xml" from the sysmgmt-service-ws-war module under the `src/test/resources/soapui` directory in your source tree.

4.1.3 Enter the project specific information.

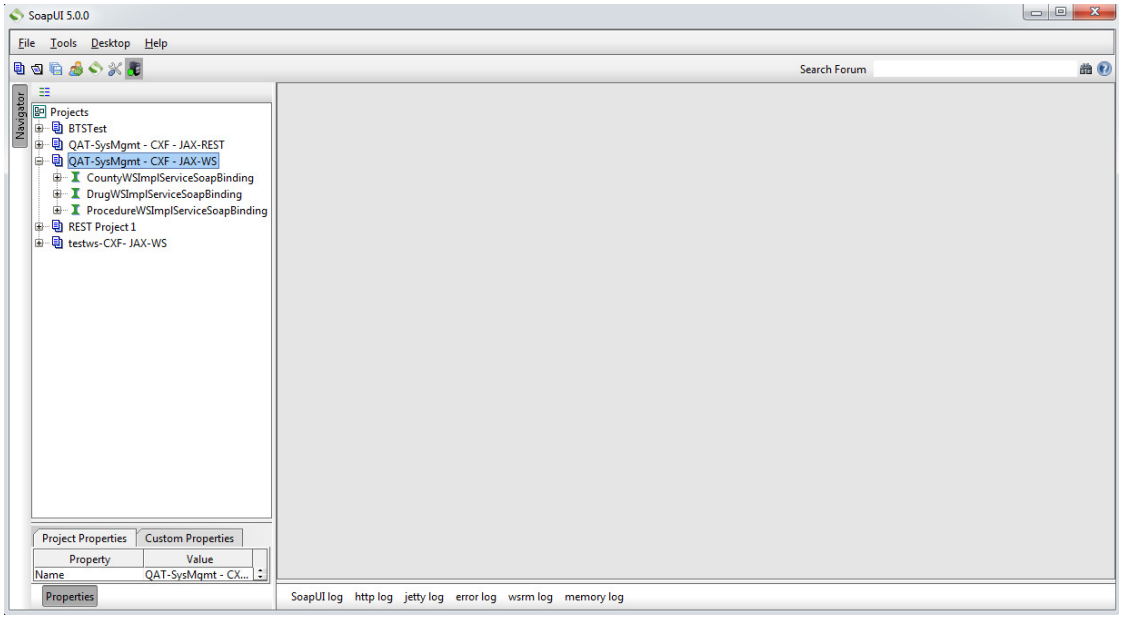
The WebServices WSDL URLs are the following::

http://localhost:8080/qat-sysmgmt-sample-ws-service/services/ws/CountyService?wsdl

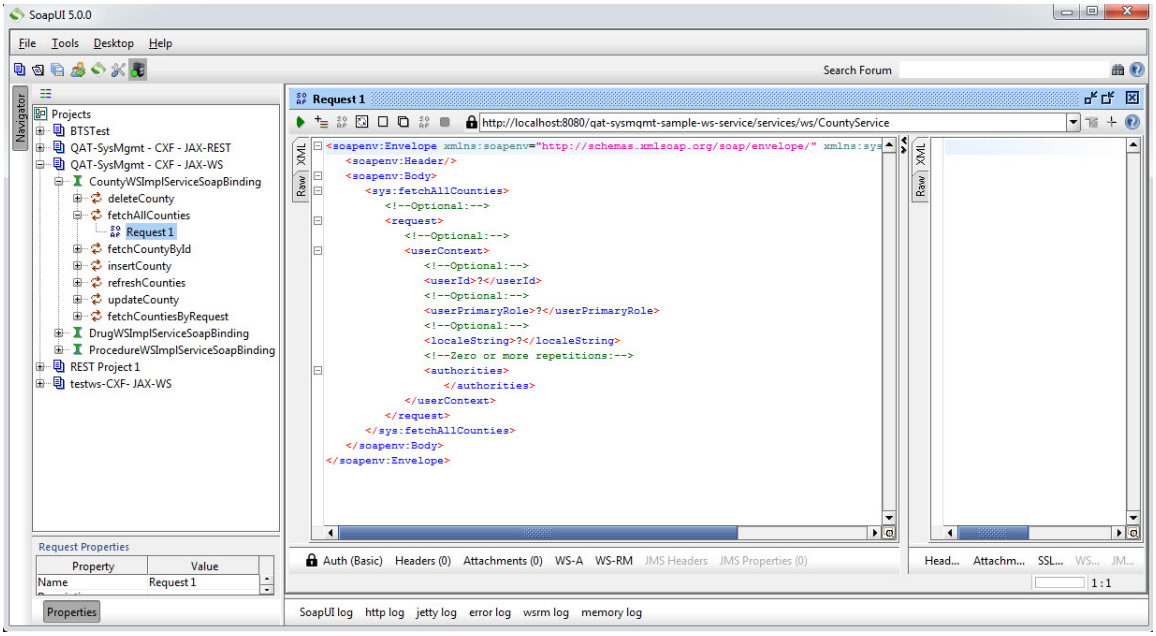
http://localhost:8080/ qat-sysmgmt-sample-ws-service /services/ws/ProcedureService?WSDL

http://localhost:8080/ qat-sysmgmt-sample-ws-service /services/ws/DrugService?WSDL

4.1.4 A new project is created and all the WS methods that are to be tested are displayed. Save the project. Make sure you add WS-Security Outgoing Settings otherwise not all your requests will work.

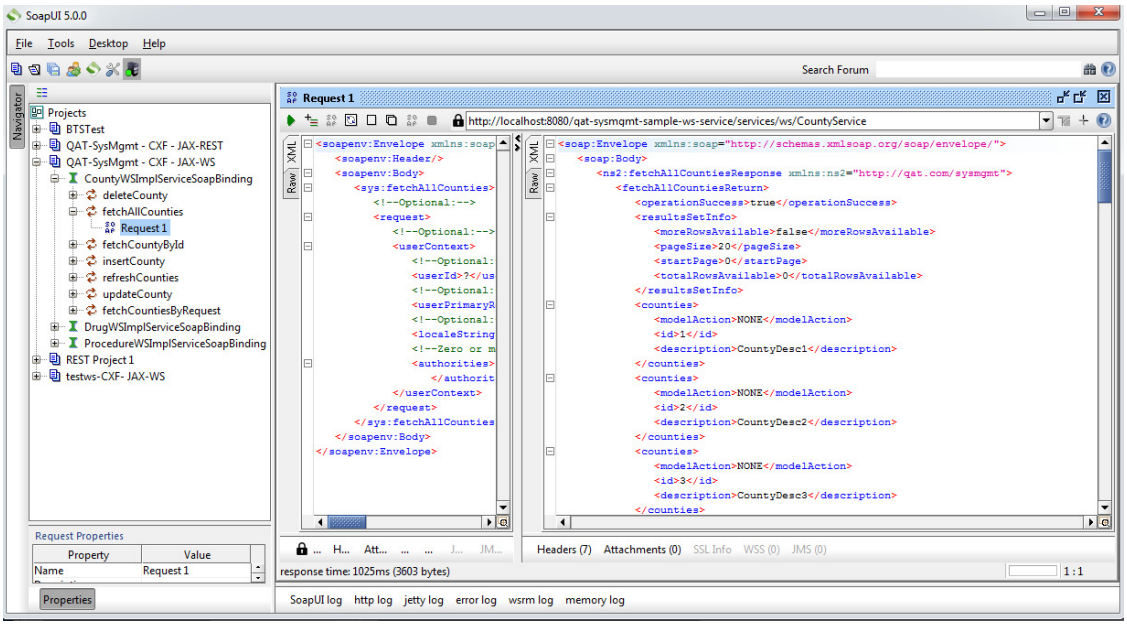


1. Click on the one of the requests.



2. You should see the request XML. Fill in appropriate parameters (where there are '?' marks) and click on the green button to execute the WS.

3. The XML response is shown in the next frame.

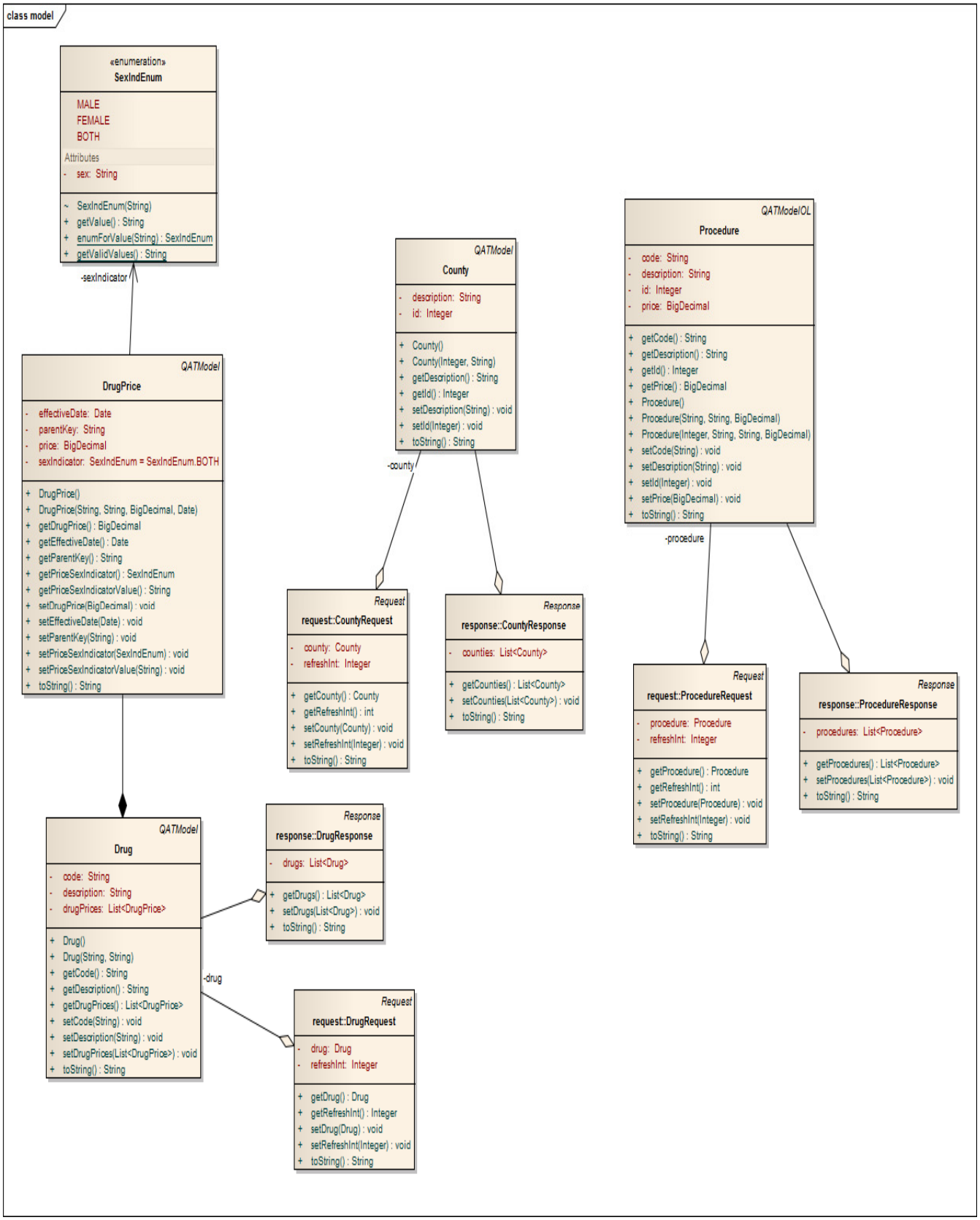


Modify the XML to feed in various parameters. You can export the request / results and make sure that the WS is functioning as required.

4. First test the following Requests,
    a. refreshCounties(XX) - this will load XX (it is an integer) rows of data
    b. refreshProcedures(XX) - this will load XX (it is an integer) rows of data
    c. refreshDrugs – No integer
    d. fetchAllCounties
    e. fetchAllProcedures
    f. fetchAllDrugs

    then use the ids returned in the above responses, to make the following requests

    g. fetchCountyById
    h. fetchProcedurebyId
    i. fetchDrugByCode
5. In the sysmgmt-service-ws-war module there is a client test file which can be run to test the county fetch web service.
    a. In Eclipse, under sysmgmt-service-ws-war/src/test/java, in the com.qat.samples.sysmgmt.client package, there is a CounyWSClient.java file. Right-click on it and select Run As > Java Application. The CXF web service client will talk to the CXF web service and return a list of counties.

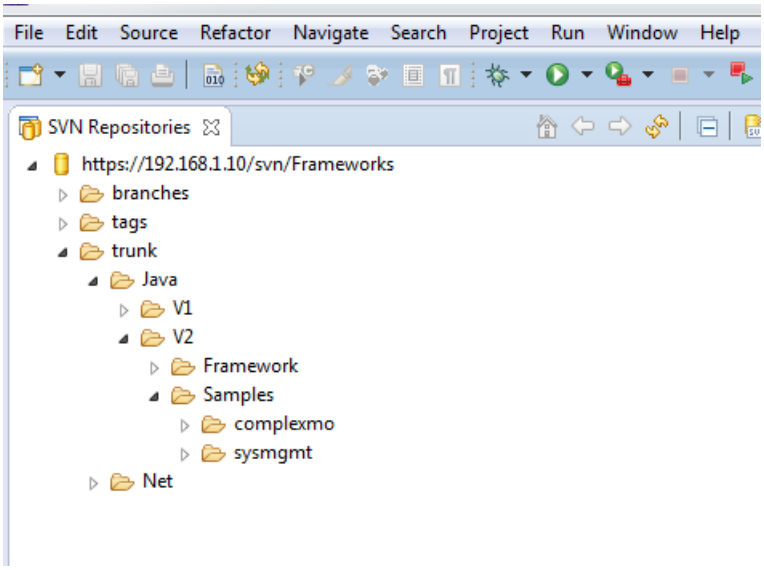


Appendix A – Project Checkout from SVN

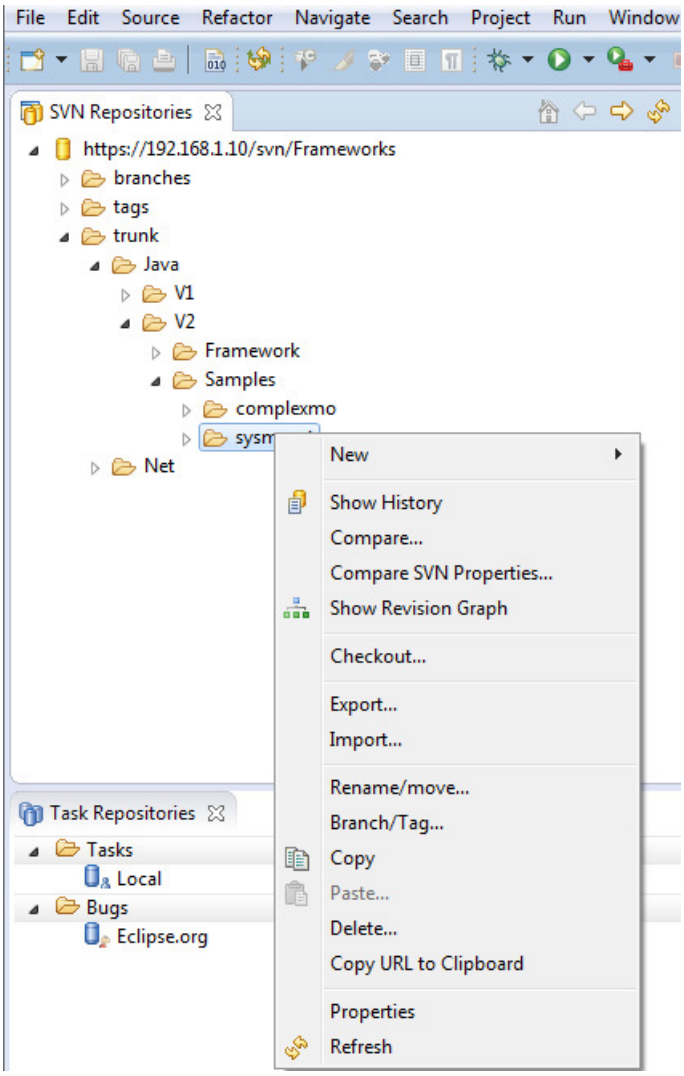
To check out the sysmgmt Maven project from SVN into your Eclipse workspace, first open the SVN Repositories perspective in Eclipse. The SVN location should be:

https://192.168.1.10/svn/Frameworks

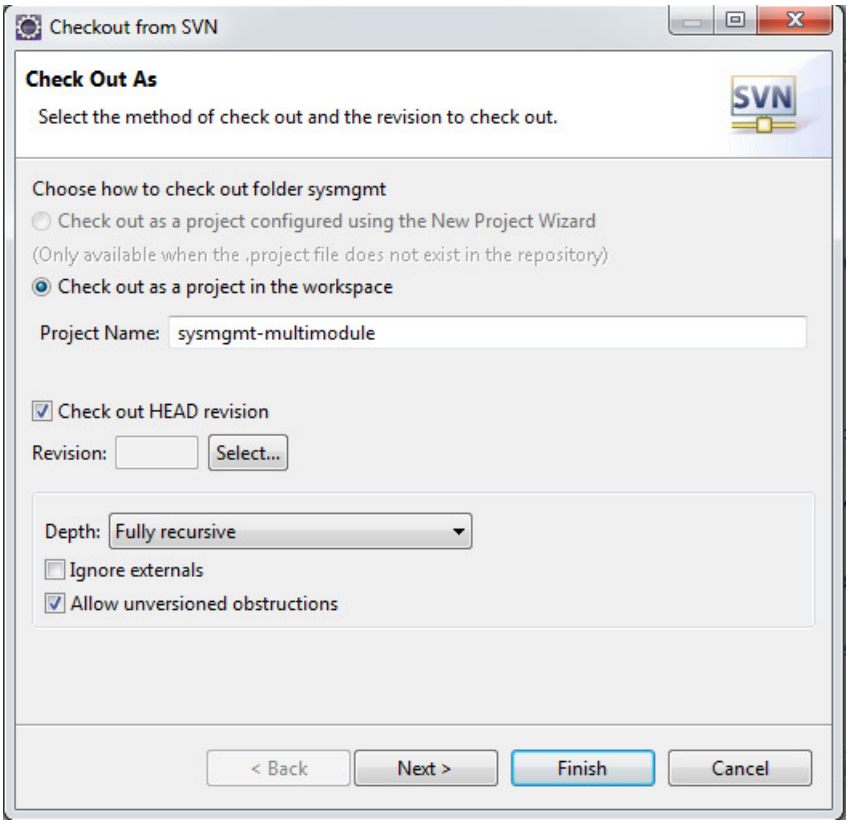
Once you have gained access to the location, expand selections from trunk > Java > V2 > Samples until you see the sysmgmt folder:



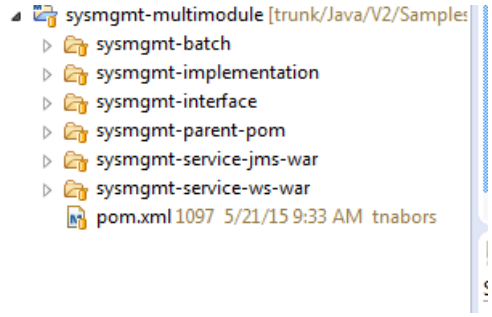
Right-click on the sysmgmt folder and select “Checkout…”



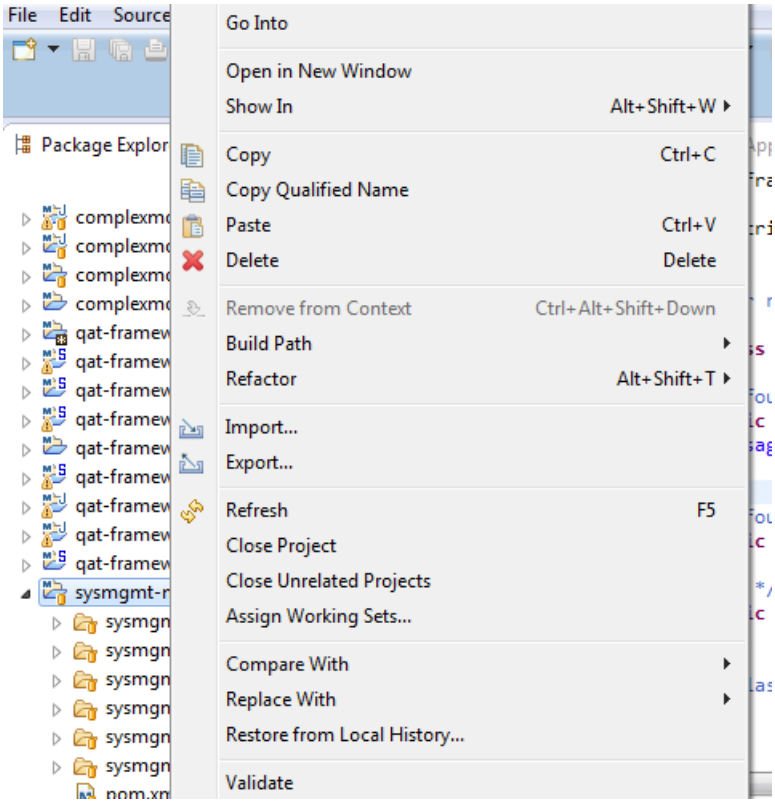
You should see a “Checkout from SVN” dialog:



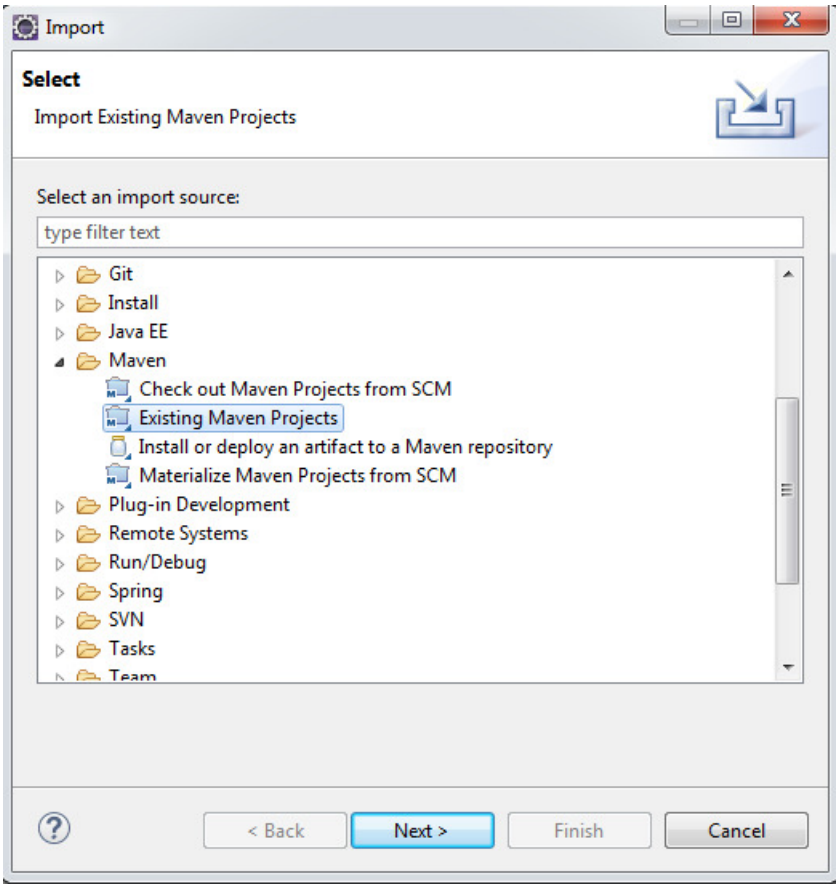
Select the “Finish” button to check out the project to your default work space. The project is a Maven “multimodule” project. After checking it out you should see the following in the Package Explorer after selecting the expansion pointer:



To be able to work with the modules in Eclipse you will want to import the modules so they appear as projects. To do this, right-click on “sysmgmt-multimodule” in the Package Explorer and select “Import…”

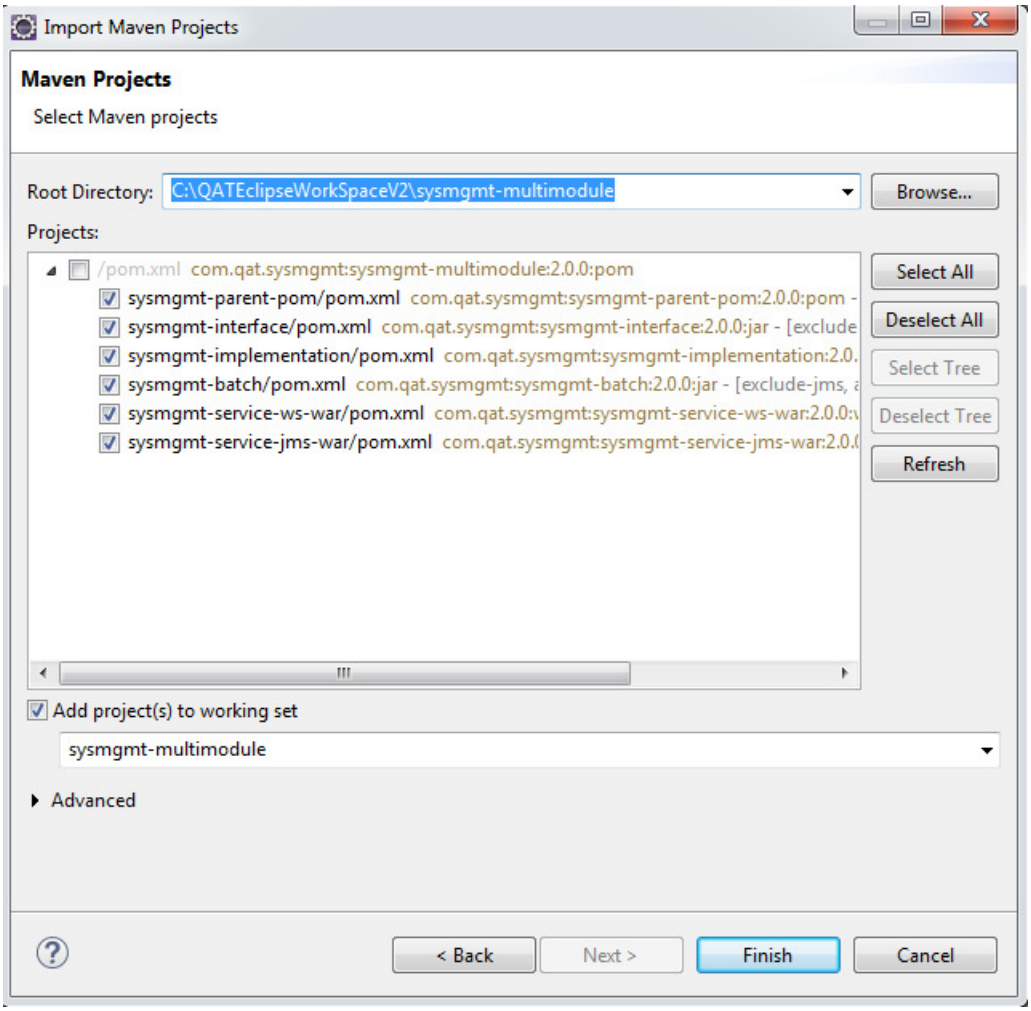


In the “Import” dialog, select Maven > Existing Maven Project



Select the “Next >” button.

In the “Import Maven Projects” dialog you should see the modules preselected:



Select the “Finish” button. After the import is completed you should see the modules appear as individual projects in the Eclipse Package Explorer: