Setup for CTS2 Framework Plugin:

Required:

Java 1.6 or newer

MySQL 5.16

Maven 2.2.1

Intellij Community Edition (preferred) or Eclipse

Gene Ontology Data Base dump

Download and install MySQL

<http://www.mysql.com/downloads/mysql/5.1.html>

I used all the defaults except I pushed the character encoding to UTF-8 and included the option for the command line client

Download and install Maven:

<http://maven.apache.org/download.html>

I stuck with 2.2.1 to try to insure compatibility with Intellij’s version. This note on Maven’s install instructions: in Windows 7 I needed to create the environmental variables in the system, rather than user, variables in order to get Maven working from the command line. I created all the recommended environmental variables including JAVA\_HOME.

Intellij or Eclipse:

<http://www.jetbrains.com/idea/download/>

<http://www.eclipse.org/downloads/>

Intellij’s Maven integration works better for me than Eclipse. You can install Eclipse’s M2e plugin for maven if you don’t mind ignoring the errors it generates.

Gene Ontology database import:

This is an ontology chosen for it’s small size and relatively rich ontology metadata table. It’s not the full Gene Ontology. You’ll need to create a new database named “godata” and import it into mysql;

From the msyql client you can just do:

Create database godata;

And then:

Mysql –u “your username” –p”your password “godata < database.dump.name

You will also need to pull in the example-service from dropbox.

Intellij:

Create a new project, add a java SDK to the project, copy source and pom files to the project. Right click on src directory and go to Mark Directory/Root Source.

Right click on the pom.xml file and select add the project as Maven Project

Comment out the assemblies plugin in the pom.xml

Using the Maven Project view, do a clean install. Both should return a value of 0.

You shoud see a small green popup in the upper right corner. Click on the upload link to upload all dependencies to maven so the application can be run.

From the CTS2Tutorial folder, get the JDBCConnection.java file and place it in the source folder.

If you run the main class here it will fail. Pull in a connector with adjustments to the pom file.

Run the main class in the jdbc connector again and it should give you metadata about the GO ontology.

Now we want to add this to the build. Using additions to the pom file we'll add the connector to a maven build.

Do a clean install

download the cts2 standalone framwork server

from whatever directory you choose to install it, run java -jar

And login as admin/admin

Choose install update from the bar at the top of the list of plugins and find the jar constructed in the maven build target folder.

Refresh the browser

Click the play/stop button next to the example service plugin.

Type localhost:8080/codesystem/GO

This should give you a hard coded response

replace the ExampleCodeSystemCatalogReadService method read with the one in the code fragments:

Use alt enter to resolve dependencies.

run clean install and upload changes to the framework.

Once again we'll type localhost:8080/codesystem/GO

This time the service goes to the database and returns GO metadata.

Eclipse instructions:

Install Eclipse and it's Maven plugin.

Extract the example-service zip to the Eclipse workspace folder

Create a new project named example-service and you should see the project structure of the new project displayed in the package explorer.

Right click on the project and select Configure/Convert to Maven Project

Your pom file will have errors but they can be ignored.