Glast Java Development Tutorial

GLAST Java Development Tutorial

This is the beginning of a tutorial on performing Java development for GLAST software. The idea of using confluence for this tutorial is to encourage others to comment on and extend the material here. The current content is certainly very incomplete so it should not be hard to improve on it.

Topics covered

Tools Overview

Tool	Used for...
CVS	Version control, a repository which keeps all of the source code, and tracks changes to source code over time
Tortoise CVS	A windows based version of CVS which integrates CVS functionality directly into windows explorer.
View CVS	A web based tool that allows the CVS repository to be browsed online
CruiseControl	An automated build system which attempts to rebuild projects automatically as soon as changes are checked into CVS. You can view the status of cruise-control on the web. You should receive an e-mail from CruiseControl everytime you check something into CVS, telling you whether you broke everything or not.
cvsspam	A system which can send you e-mail whenever anyone checks code into our CVS repository. We provide an email list you can subscribe to (go to http://www-glast.stanford.edu/cgi-prot/subscribe.pl and subscribe to the Java CVS archive updates list) to receive these commit notifications directly, or you can simply access the email archive.
Maven	Build tool and project management. Almost all GLAST software projects use Maven. Maven can be used to compile code, deploy applications, build web sites. It is based on a project description which declares how the code is to be build, and what other projects (Glast and 3rd party) the code depends on. Maven takes care downloading the appropriate versions of all dependencies for you. Note we use maven 1.0.2 for our projects, newer versions are incompatible and will not work
Tomcat	Web/Application server. Tomcat is a Java based web server, commonly called an application server. Tomcat allows web applications to be dynamically (re) deployed. We run a number of different tomcat servers, some for production use, some for development/testing. Tomcat can also be started on your own machine for testing applications as you develop them.
Netbeans	An integrated development environment for developing Java applications. We primarily use netbeans for developing web applications, and use the tomcat server which is built-in to netbeans for testing these applications.
mevenide	mevenide is a plugin which can be installed into netbeans, which extends netbeans so that it understands how to open, build and deploy maven based applications
glast-ground	Our main web site, from which all web applications are accessed.
CAS	A "single sign-on" system used for logging in and out of our web applications.

Installing the tools

Windows

Instructions for installing Java, Maven and Netbeans with mevenide support are already available as part of the ILC confluence web space.

- Java installation instructions
- Java Cryptography Extension (JCE)
- Maven installation instructions
- Netbeans and mevenide installation instructions

Though not strictly required, it is also recommended to install TortoiseCVS.

Unix

On SLAC unix most of the tools you require, including Java, Netbeans, Maven, are already installed centrally.

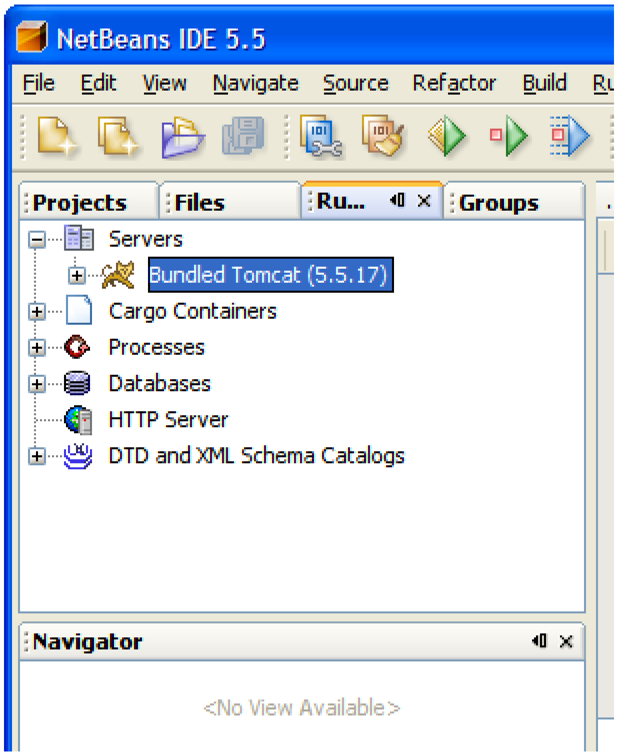
- Java is in /usr/local/bin/java.
- Netbeans is in /usr/local/bin/netbeans
- Maven in in /usr/local/bin/maven

The mevenide plugin is not installed, so needs to be installed manually. You can follow the instructions for installing mevenide given on the ILC website. Charlotte has created some additional notes.

Configuring netbeans

When building web applications for GLAST we do not build in the information about how to connect to a database. Instead the applications refer to database connections by symbolic names like "jdbc/glastgen" which are linked to real database connections in the tomcat configuration file. This avoids having to put database passwords into the applications themselves, and makes it possible to use tomcat's "connection pooling" mechanism (which makes database access more efficient).

When running tomcat inside netbeans we need to configure it to be able map these database connections. To do this, start netbeans, go to the "runtime" tab on the left, select Servers, Bundled Tomcat, right click and select "Edit server.xml".



The server.xml file should now open in the editor. You need to post all of the standard GLAST connections into this file between the <GlobalNamingResources> </GlobalNamingResources> tags.

The best place to get these would be from ~glast/tomcat/BASE55/common/conf/server.xml, which is the file used by the production tomcat servers, however it is protected so only the GLAST account can read it (because it contains the passwords). For now you will have to ask someone who has the glast password (Tony or Max for example) to send it to you.

You also need to set up the tomcat server installed as part of Netbeans to have access to the oracle JDBC driver. To do this download ojdbc.jar and save it in:

```
~/.netbeans/5.5/apache-tomcat-5.5.17_base/common/lib
```

where ~ is your home directory (on windows this normally means c:\documents and settings\<userid>). You will probably need to create the common and lib directories. Then edit the catalina catalina.properties in

```
~/.netbeans/5.5/apache-tomcat-5.5.17_base/conf/catalina.properties
```

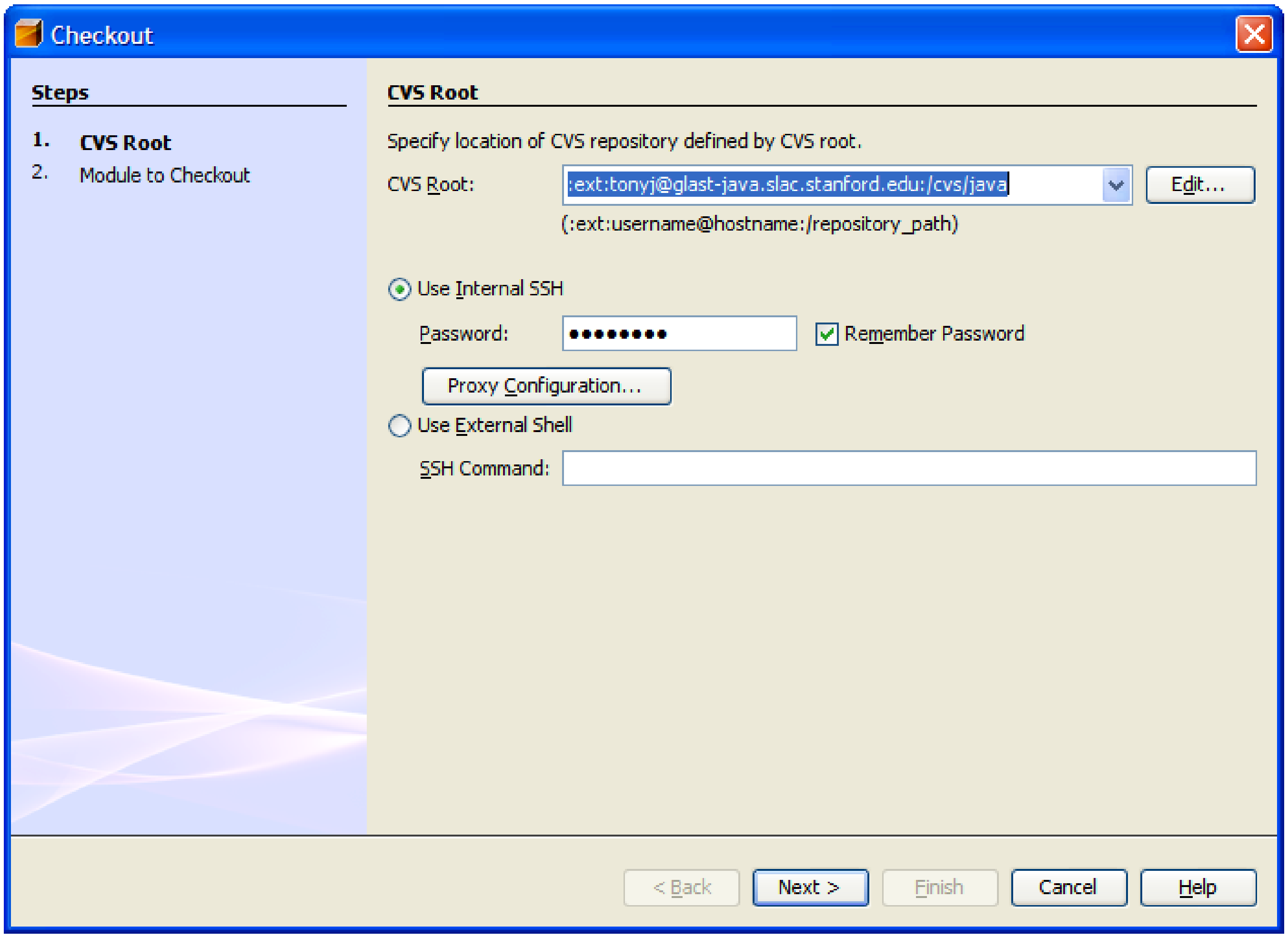
and change the "common.loader" line to:

```
common.loader=${catalina.home}/common/classes,${catalina.home}/common/i18n/*.jar,${catalina.home}/common/lib/*.
jar,${catalina.base}/common/lib/*.jar
```

Getting Started with Netbeans

Checking out a project

Start netbeans, from the menu choose "CVS", "Checkout", fill in the form below (substitute your userid for tonyj and fill in your password)



Specify location of CVS repository defined by CVS root.

CVS Root: :ext:tonyj@glast-java.slac.stanford.edu:/cvs/java

(:ext:username@hostname:/repository_path)

[✓ Use Internal SSH]

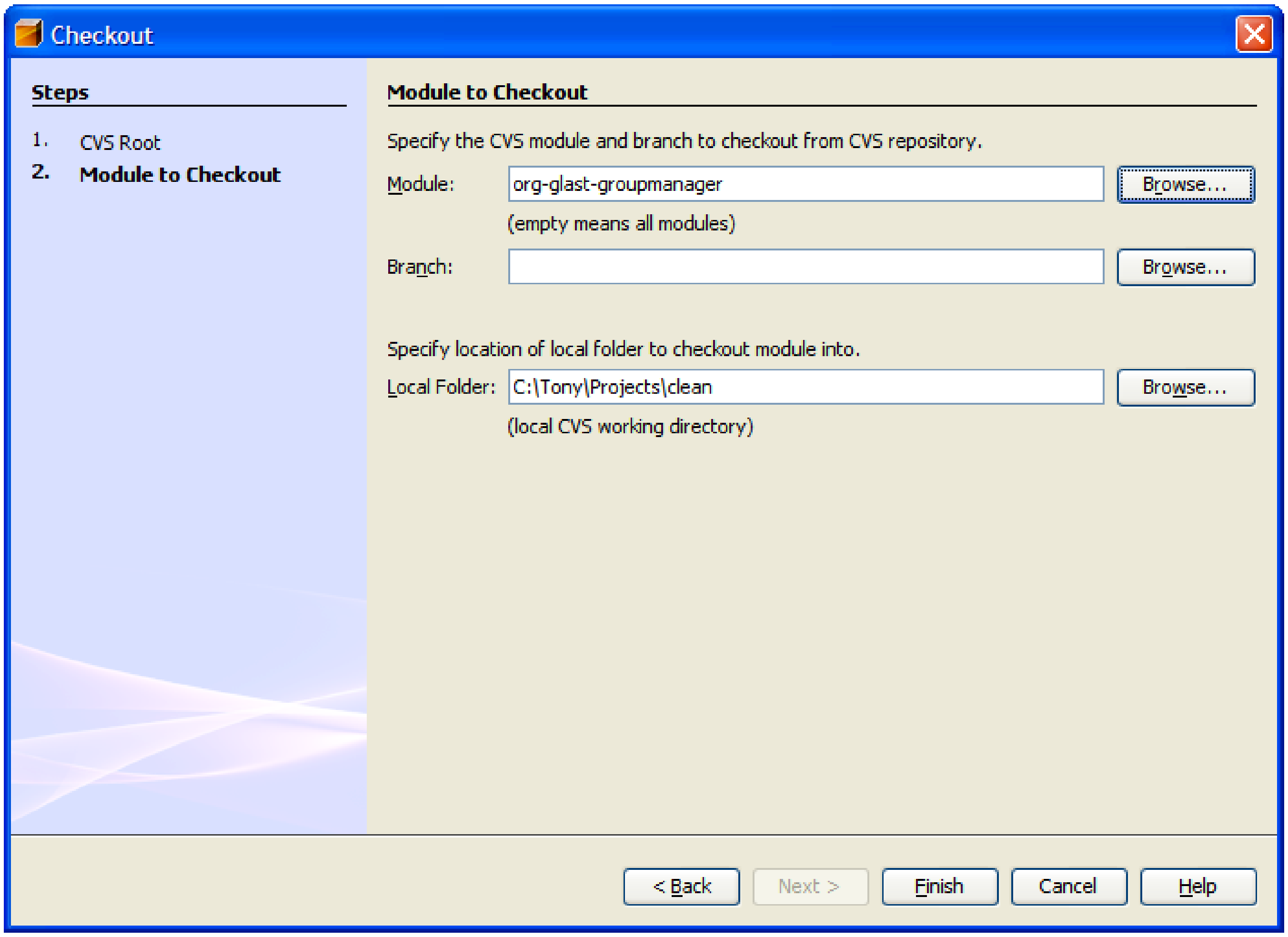
Password: ●●●●●●●●●

[✓ Remember Password]

Use External Shell

SSH Command:

next select the module you want to checkout, you can use the "Browse...." button, and also select where you want the project stored on your machine.



Specify the CVS module and branch to checkout from CVS repository.

Module: org-glast-groupmanager

(empty means all modules)

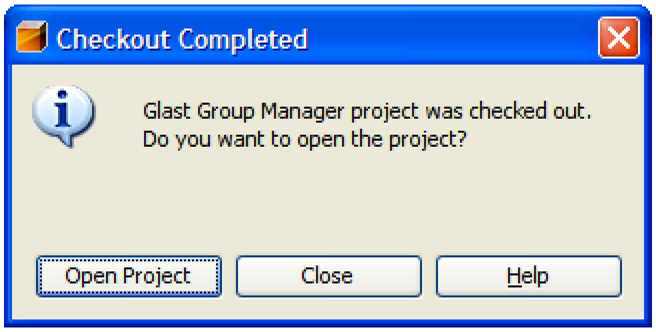
Branch:

Specify location of local folder to checkout module into.

Local Folder: C:\Tony\Projects\clean

(local CVS working directory)

Finally select "Open Project" when prompted.



Glast Group Manager project was checked out.
Do you want to open the project?

Opening a project with Netbeans

If you followed the instructions above to checkout and open a project you are ready to go. If you check the project out from CVS yourself then in Netbeans select "File", "Open Project" then navigate to the folder where you checked out the project and select "Open Project Folder".

If Netbeans doesn't recognize your folder as a project that probably indicates that mevendie was not successfully installed.

Anatomy of a web application

We will use the org-glast-groupmanager project as an example. Its contents can be viewed in CVS.

All of our web applications consist of the following:

- A top level directory which contains the maven files.

    - project.xml – The main description of the maven project, including the list of dependencies
    - project.properties – Additional information used by project.xml
    - maven.xml – Specifies additional "goals" which can be run on the project. We use this file to specify how the project should be deployed to central tomcat servers.
- A src/webapp directory which contains the Java Server Pages (JSP) for the application
- A src/main/java directory which contains any Java source code, and related resource files.

Building Web Applications

Using the Java Standard Template Library (JSTL)

The Java standard template library (JSTL) provides some common libraries for use in JSP pages. The JSTL includes the following modules

- Core – core functionality such as if, loop, set, include
- SQL – database access
- XML – XML Processing
- Internationalization – I18N Formatting (not normally used by GLAST)

Some tutorials on using JSTL are available here.

Starting a new Web Application

Documentation on starting a new web application is available here.

Using the GLAST common libraries

Documentation on using the GLAST common libraries are available here.