Requirements

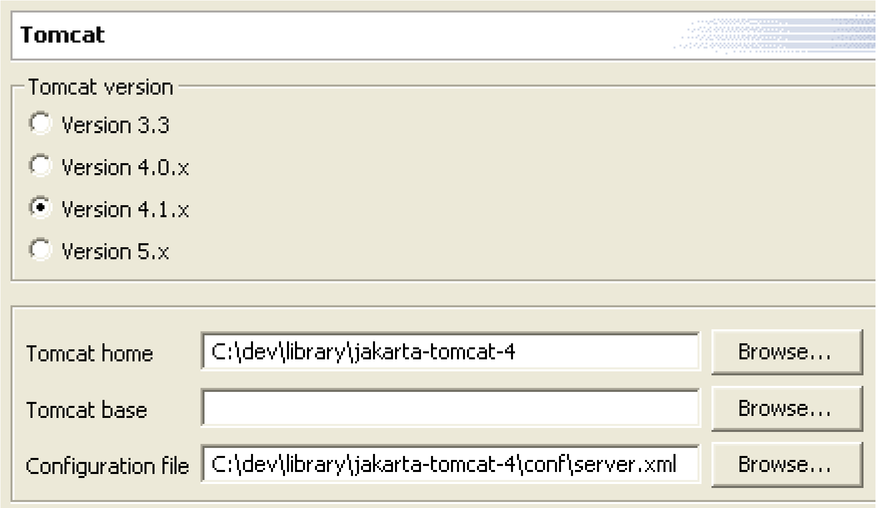
1. Eclipse Eclipse 3.0M8
2. Tomcat launcher Eclipse plugin (http://www.sysdeo.com/eclipse/tomcatPlugin.html)
3. Tomcat4 patch for JSP debugging (http://www.sysdeo.com/eclipse/tomcatPlugin.html)
4. Recent Tomcat4.1 distribution (e.g., 4.1.30)

If you need write access to the Cvs Repository:

5. Cygwin

Installation

1. Install the Eclipse plugin and the Tomcat4 patch (The patch is binary, so no recompilation of Tomcat needed)
2. Use Window -> Preferences... to activate the Tomcat plugin (see image below)



Tomcat

Tomcat version

○ Version 3.3

○ Version 4.0.x

● Version 4.1.x

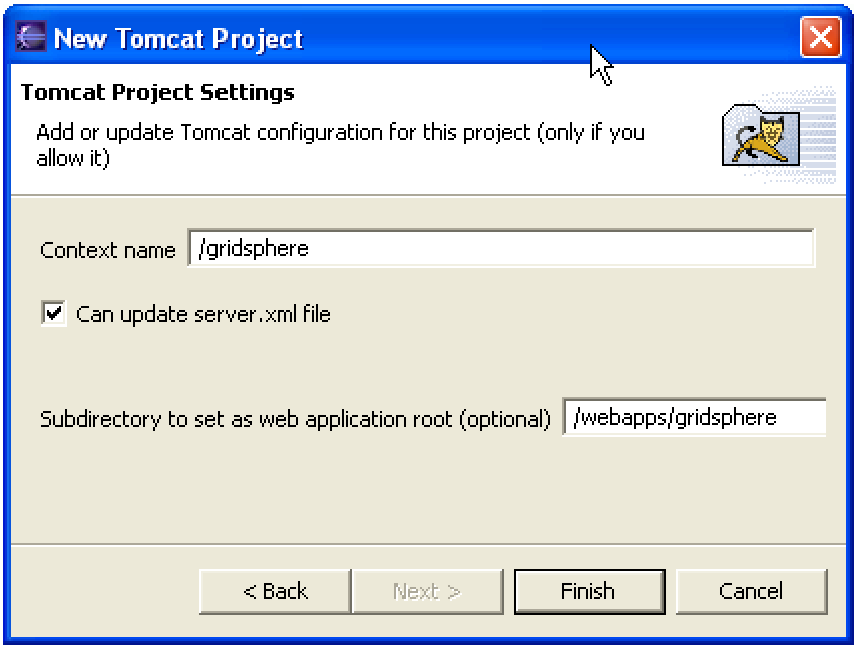
○ Version 5.x

Tomcat home: C:\dev\library\jakarta-tomcat-4

Tomcat base:

Configuration file: C:\dev\library\jakarta-tomcat-4\conf\server.xml

3. Create a new Tomcat Project (e. g. GSProject, context name="/gridsphere", web application root="/webapps/gridsphere", see image below).



New Tomcat Project

Tomcat Project Settings

Add or update Tomcat configuration for this project (only if you allow it)

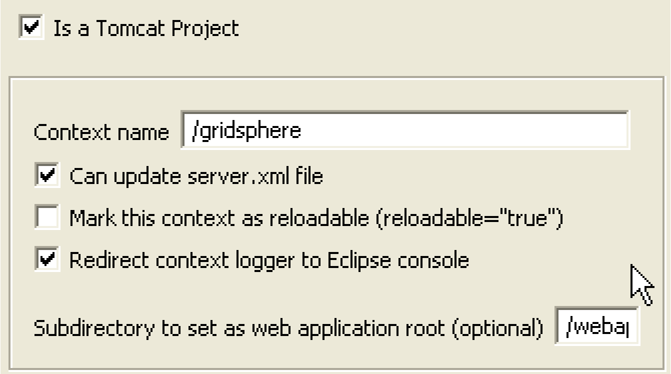
Context name: /gridsphere

☑ Can update server.xml file

Subdirectory to set as web application root (optional): /webapps/gridsphere

4. In the project's settings (see image below)

- Can update server.xml
- Don't be a reloadable context (It messes up anyway)
- Redirect context logger to Eclipse



☑ Is a Tomcat Project

Context name: /gridsphere

☑ Can update server.xml file

□ Mark this context as reloadable (reloadable="true")

☑ Redirect context logger to Eclipse console

Subdirectory to set as web application root (optional): /weba|

5. Check out as an existing project

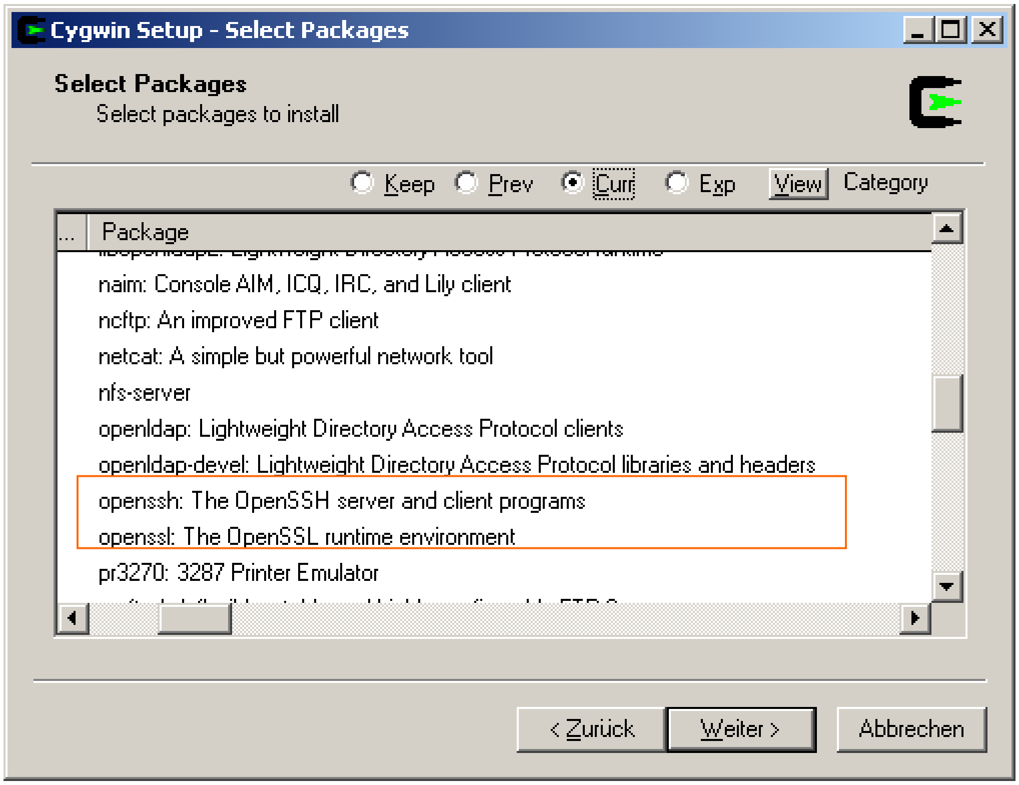
a) anonymous with no write access:

- Host: portal.aei.mpg.de
- Repository path: /home/repository
- User: anonymous
- Password: < leave blank >
- Connection type: pserver
- CVS Module: gridsphere

b) non-anonymous with write access (you will need a ssh2 key):

- Install Cygwin

- In the “Select Packages” Dialog select the following Packages (you will find them under the “Net” Folder):



Cygwin Setup - Select Packages

Select Packages

Select packages to install

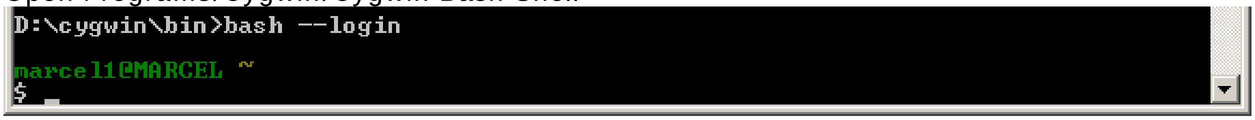
○ Keep ○ Prev ● Curr ○ Exp View Category

Package

naim: Console AIM, ICQ, IRC, and Lily client
ncftp: An improved FTP client
netcat: A simple but powerful network tool
nfs-server
openldap: Lightweight Directory Access Protocol clients
openldap-devel: Lightweight Directory Access Protocol libraries and headers
openssh: The OpenSSH server and client programs
openssl: The OpenSSL runtime environment
pr3270: 3287 Printer Emulator

< Zurück Weiter > Abbrechen

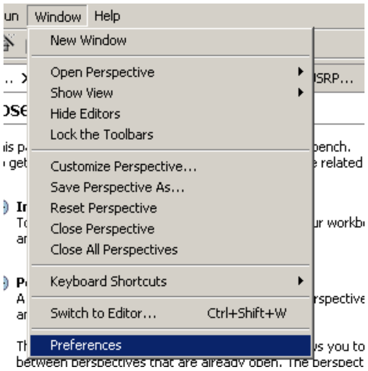
- Open Programs/Cygwin/Cygwin Bash Shell



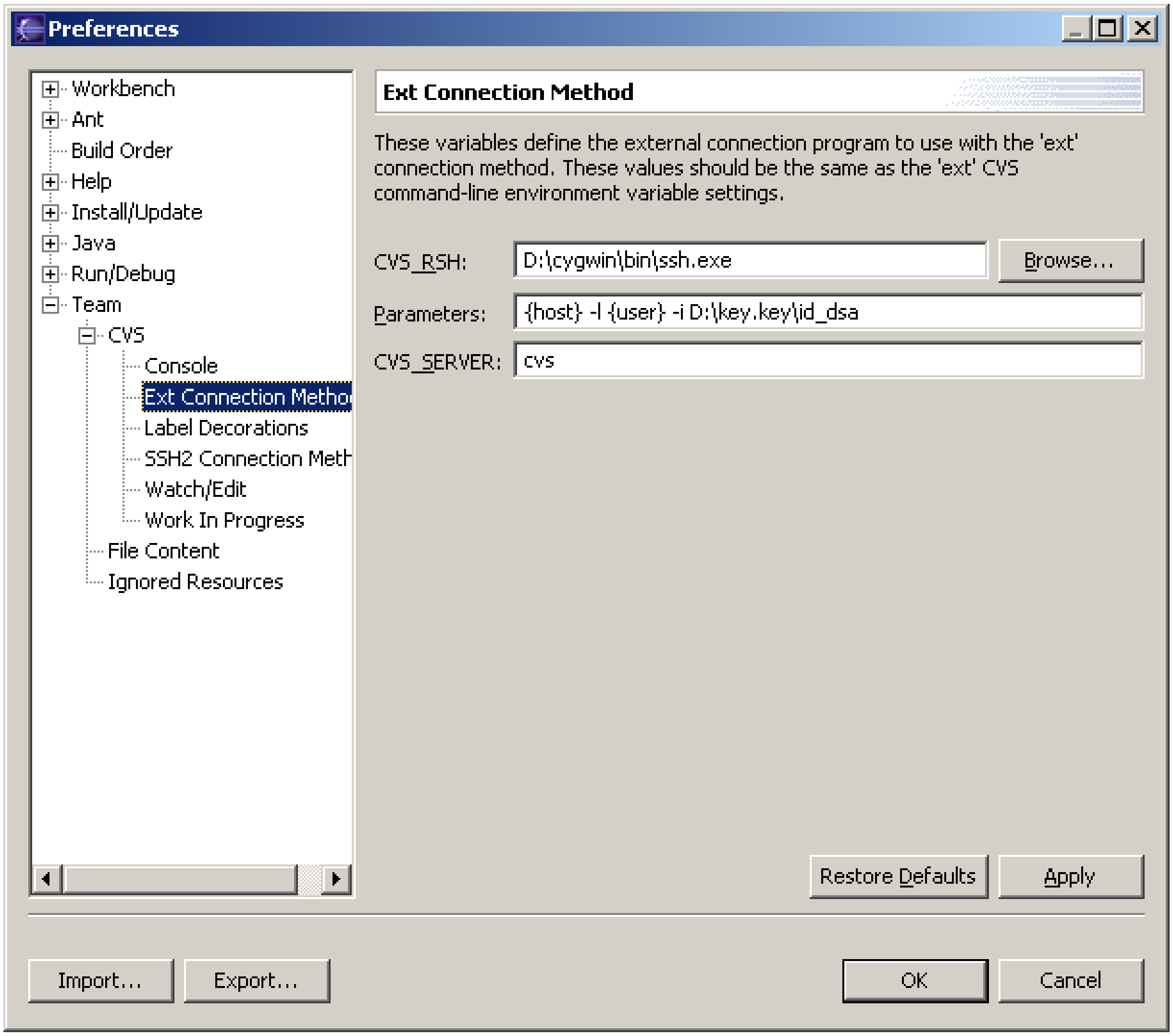
```
D:\cygwin\bin>bash --login

marcel1@MARCEL ~
$ _
```

- Enter the following command: “ssh –l[username] portal.aei.mpg.de”
- He will ask you to accept the server fingerprint. Type yes.
- He will ask for a password. Press Ctrl+Break
- Type “exit” (for experts: this was to add the server to knowhosts)
- Start Eclipse
- Open the Preferences Dialog



- Open the “Ext Connection Method” Dialog



Ext Connection Method

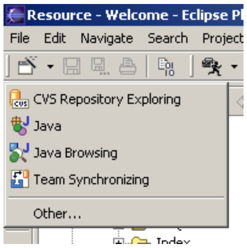
These variables define the external connection program to use with the 'ext' connection method. These values should be the same as the 'ext' CVS command-line environment variable settings.

CVS_RSH: D:\cygwin\bin\ssh.exe

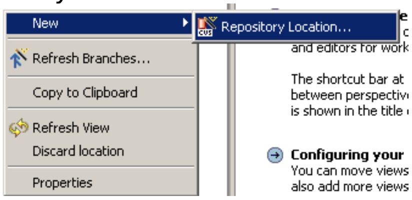
Parameters: {host} -l {user} -i D:\key.key\id_dsa

CVS_SERVER: cvs

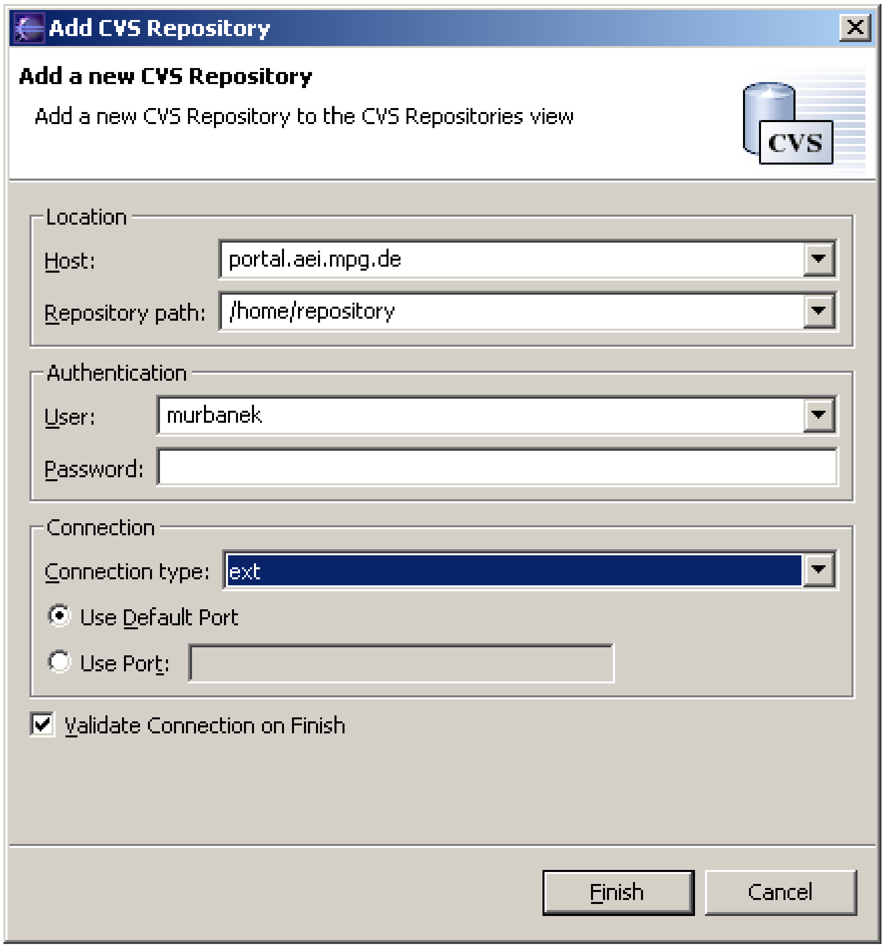
- Under CVS_RSH fill in the following: “[your cygwin directory]\bin\ssh.exe”
- Under Parameters: “{host} –l {user} –i [your private ssh2 keyfile]”
- Open the Repository View:



- Right-click “new”->”Repository location”



- Following dialog appears:



Add CVS Repository

Add a new CVS Repository

Add a new CVS Repository to the CVS Repositories view

Location

Host: portal.aei.mpg.de

Repository path: /home/repository

Authentication

User: murbanek

Password:

Connection

Connection type: ext

● Use Default Port

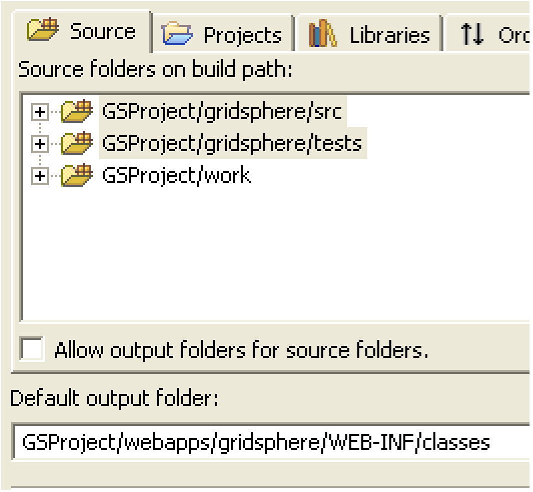
○ Use Port:

☑ Validate Connection on Finish

- Under host fill in: portal.aei.mpg.de
- Repository path:/home/repository
- User: your username
- No Password
- Connection type is “ext”
- Press “Finish”

6. Setup Java Build Path in Project Properties

- Add source folder gridsphere/src and gridsphere/tests (see image below)



Source folders on build path:

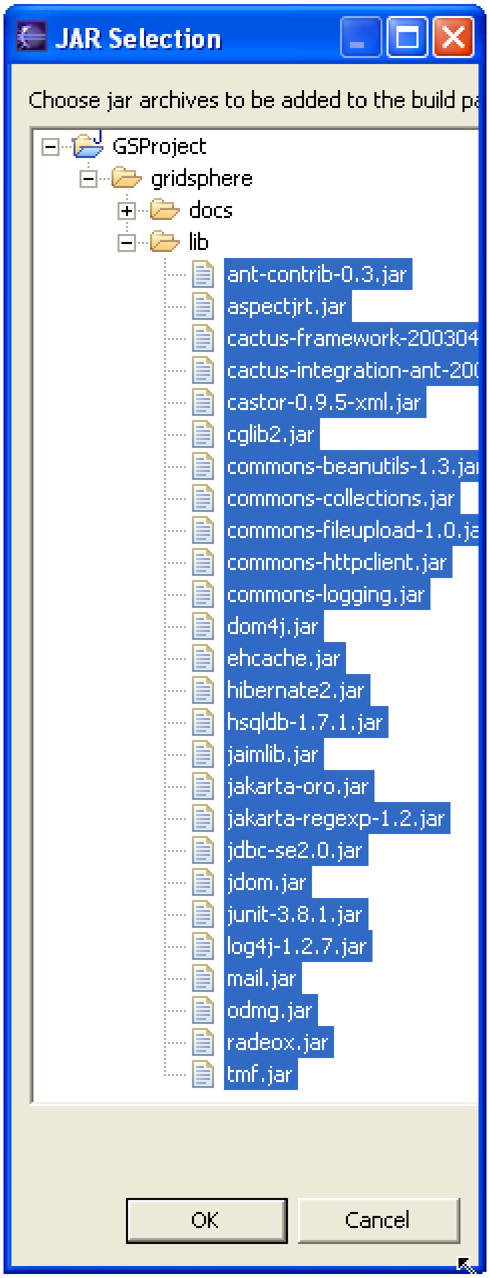
- GSProject/gridsphere/src
- GSProject/gridsphere/tests
- GSProject/work

☐ Allow output folders for source folders.

Default output folder:

GSProject/webapps/gridsphere/WEB-INF/classes

- Add the libraries in gridsphere/lib (see image below)



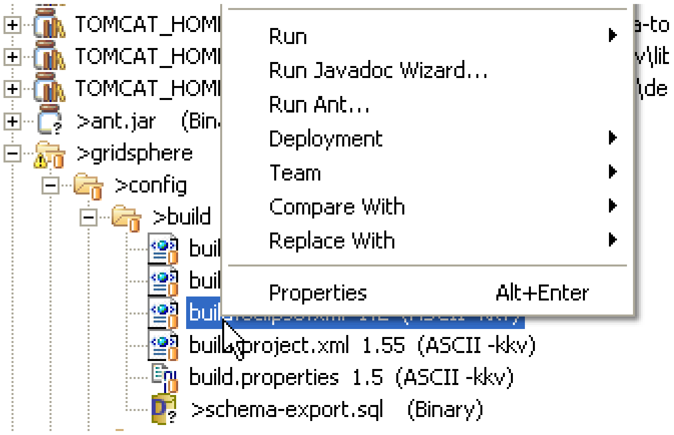
Choose jar archives to be added to the build pa

- GSProject
  - gridsphere
    - docs
    - lib
      - ant-contrib-0.3.jar
      - aspectjrt.jar
      - cactus-framework-200304
      - cactus-integration-ant-20(
      - castor-0.9.5-xml.jar
      - cglib2.jar
      - commons-beanutils-1.3.jai
      - commons-collections.jar
      - commons-fileupload-1.0.ja
      - commons-httpclient.jar
      - commons-logging.jar
      - dom4j.jar
      - ehcache.jar
      - hibernate2.jar
      - hsqldb-1.7.1.jar
      - jaimlib.jar
      - jakarta-oro.jar
      - jakarta-regexp-1.2.jar
      - jdbc-se2.0.jar
      - jdom.jar
      - junit-3.8.1.jar
      - log4j-1.2.7.jar
      - mail.jar
      - odmg.jar
      - radeox.jar
      - tmf.jar

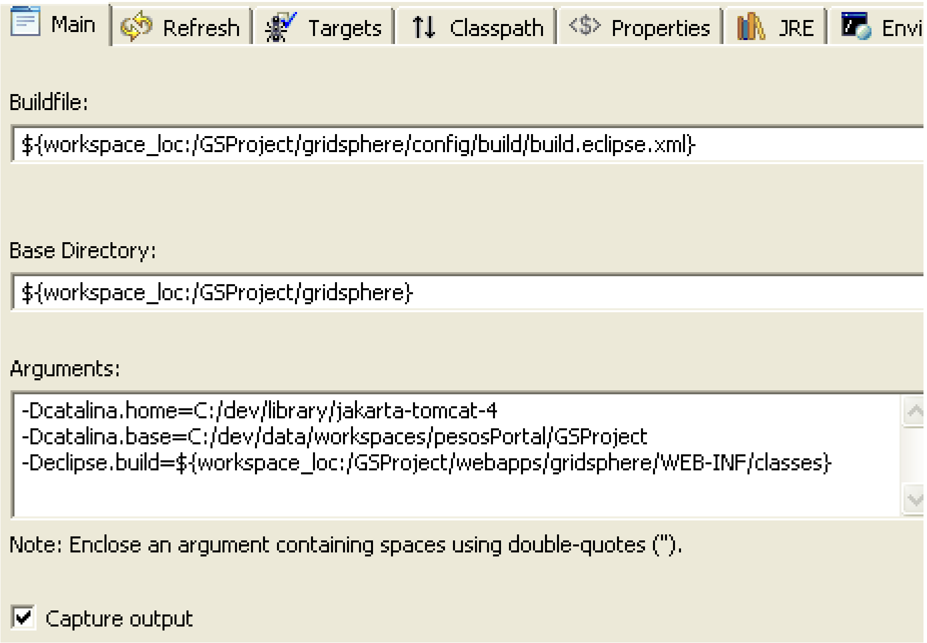
- Extend the variable TOMCAT_HOME
  - Add the activation.jar from Tomcat common/lib
  - Add the ant.jar from Tomcat common/lib
  - Add the jta.jar from Tomcat common/lib
  - Add the xercesImpl.jar from Tomcat common/endorsed
  - Add the XmlParserAPIs.jar from Tomcat common/endorsed

- Export the libraries and the source folders (Order and Export tab, select all, Ok)

7. Right Click on GSProject/gridsphere/config/build/eclipse.build.xml and selection “Run Ant” (see images below)



- Ant resource: GSProject/gridsphere/config/build/eclipse.build.xml
- Working directory: GSProject/gridsphere
- Parameters:
  - -Dcatalina.home=/path/to/tomcat/installation
  - -catalina.base=/path/to/gridsphere project (note the use of slash "/" instead of backslash "\".)
  - -Declipse.build=${workspace_loc:/GSProject/path/to/build/directory}



Buildfile:

${workspace_loc:/GSProject/gridsphere/config/build/build.eclipse.xml}

Base Directory:

${workspace_loc:/GSProject/gridsphere}

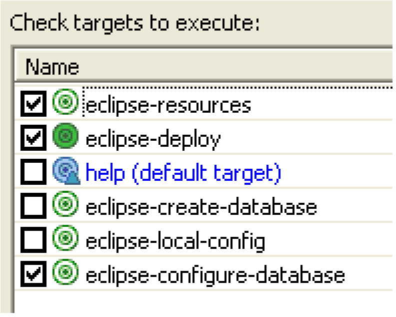
Arguments:

```
-Dcatalina.home=C:/dev/library/jakarta-tomcat-4
-Dcatalina.base=C:/dev/data/workspaces/pesosPortal/GSProject
-Declipse.build=${workspace_loc:/GSProject/webapps/gridsphere/WEB-INF/classes}
```

Note: Enclose an argument containing spaces using double-quotes (").

☑ Capture output

- Targets (in this order, **order matters**): eclipse-deploy, eclipse-create-database, eclipse-resources



Check targets to execute:

Name

☑ eclipse-resources
☑ eclipse-deploy
☐ help (default target)
☐ eclipse-create-database
☐ eclipse-local-config
☑ eclipse-configure-database

- Click Run, you should see something in the console like below.

```
Buildfile: C:\dev\data\workspaces\pesosPortal\GSProject\gridsphere\config\build\build.eclipse.xml
eclipse-resources:
    [mkdir] Created dir: C:\dev\data\workspaces\pesosPortal\GSProject\webapps\gridsphere\WEB-
INF\classes\gridsphere
     [copy] Copying 1 file to
C:\dev\data\workspaces\pesosPortal\GSProject\webapps\gridsphere\WEB-INF\classes\gridsphere
     [copy] Copying 10 files to
C:\dev\data\workspaces\pesosPortal\GSProject\webapps\gridsphere\WEB-
INF\classes\gridsphere\resources
eclipse-deploy:
eclipse-configure-database:
    [mkdir] Created dir: C:\dev\data\workspaces\pesosPortal\GSProject\webapps\gridsphere\WEB-
INF\persistence
     [copy] Copying 1 file to
C:\dev\data\workspaces\pesosPortal\GSProject\webapps\gridsphere\WEB-INF\persistence
     [copy] Copying 192 files to C:\dev\data\workspaces\pesosPortal\GSProject\webapps\gridsphere
eclipse-configure-database:
     [copy] Copying 1 file to
C:\dev\data\workspaces\pesosPortal\GSProject\webapps\gridsphere\WEB-INF\persistence
eclipse-create-database:

  [echo] Successfully created new database
eclipse-resources:
BUILD SUCCESSFUL
Total time: 6 seconds
```

- If you add the following 4 lines to the GSProject\gridsphere\config\log4j.properties file, you see hibernate logging which tells you the specifics of the database set up. I recommend doing this.

```
1. log4j.logger.net.sf.hibernate=DEBUG
2. log4j.logger.net.sf.hibernate.cfg.Binder=debug
3. log4j.logger.net.sf.hibernate.cfg.Environment=debug
4. log4j.logger.net.sf.hibernate.util.DTDEntityResolver=DEBUG
```

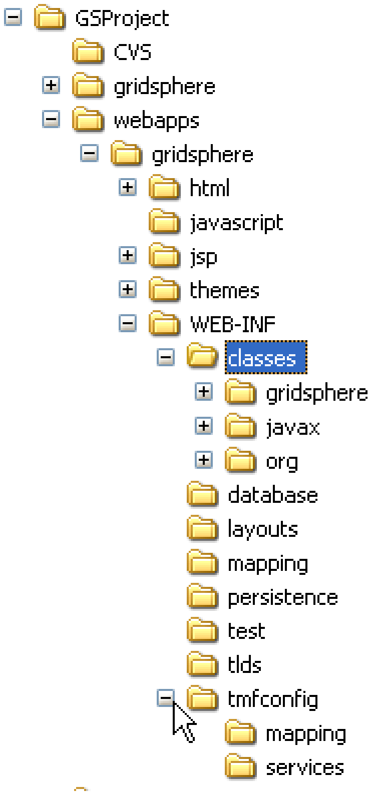
Sample Hibernate Logging:

```
SchemaExport  - Running hbm2ddl schema export
SchemaExport  - writing generated schema to file: schema-export.sql
SchemaExport  - exporting generated schema to database
DriverManagerConnectionProvider  - Using Hibernate built-in connection pool (not for production
use!)
DriverManagerConnectionProvider  - Hibernate connection pool size: 20
DriverManagerConnectionProvider  - using driver: org.hsqldb.jdbcDriver at URL:
jdbc:hsqldb:C:/dev/data/workspaces/pesosPortal/GSProject/webapps/gridsphere/WEB-
INF/database/gridsphere
DriverManagerConnectionProvider  - connection properties: {poolsize=40, user=sa, password=}
DriverManagerConnectionProvider  - total checked-out connections: 0
DriverManagerConnectionProvider  - opening new JDBC connection
DriverManagerConnectionProvider  - created connection to:
jdbc:hsqldb:C:/dev/data/workspaces/pesosPortal/GSProject/webapps/gridsphere/WEB-
INF/database/gridsphere, Isolation Level: 1
```

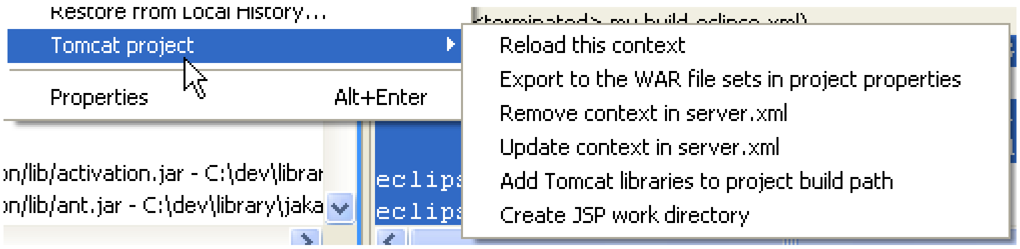
- Check that the database is installed in the right place. You should have 2 files in the \GSProject\webapps\gridsphere\WEB-INF\database directory.

  1. gridsphere.properties
  2. gridsphere.script

- You should see a dir structure like the image below.



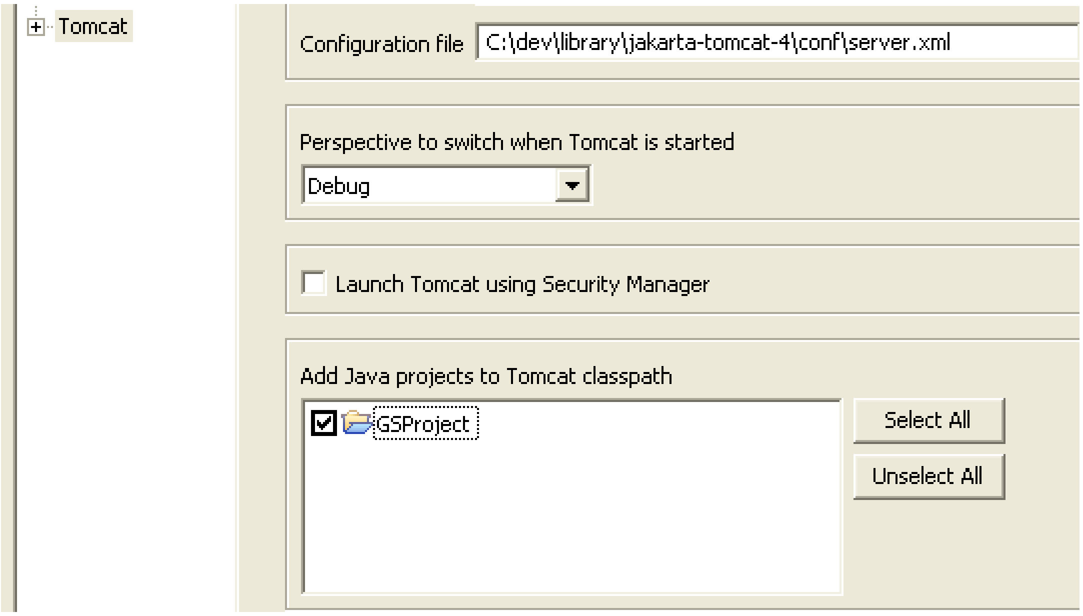
8. Refresh the Eclipse project
9. From the project context menu -> Tomcat -> update server.xml (see image below)


Restore from Local History...
Tomcat project
Properties
Alt+Enter
Reload this context
Export to the WAR file sets in project properties
Remove context in server.xml
Update context in server.xml
Add Tomcat libraries to project build path
Create JSP work directory
on/lib/activation.jar - C:\dev\librar
on/lib/ant.jar - C:\dev\library\jaka
eclips
eclips

10. Go to server.xml (e.g., jakarta-tomcat-4\conf\server.xml) and adjust the generated Context so that it includes the crossContext="true" attribute. See sample below.

```
<Context crossContext="true"  path="/gridsphere" reloadable="false"
docBase="C:\dev\data\workspaces\pesosPortal\GSProject\webapps\gridsphere"
workDir="C:\dev\data\workspaces\pesosPortal\GSProject\work\org\apache\jsp" >
```

11. In Window -> Preferences -> Tomcat add the project (here GSProject) to the Tomcat Classpath. Also change the Tomcat 4.1.x launch configuration to start in the Debug perspective (see image below)



Configuration file C:\dev\library\jakarta-tomcat-4\conf\server.xml

Perspective to switch when Tomcat is started

Debug

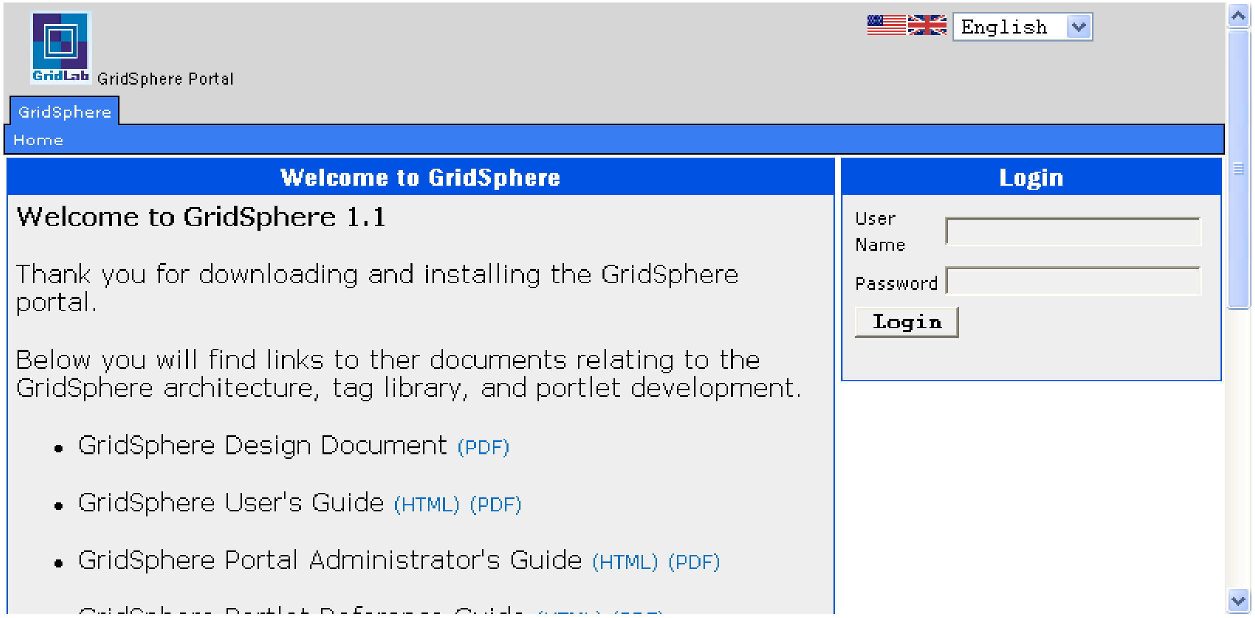
☐ Launch Tomcat using Security Manager

Add Java projects to Tomcat classpath

☑ GSProject

12. Start Tomcat and enter http://127.0.0.1:8080/gridsphere/gridsphere in a brower

- You should see the following in your browser window.



Welcome to GridSphere

Welcome to GridSphere 1.1

Thank you for downloading and installing the GridSphere portal.

Below you will find links to ther documents relating to the GridSphere architecture, tag library, and portlet development.

- GridSphere Design Document (PDF)
- GridSphere User's Guide (HTML) (PDF)
- GridSphere Portal Administrator's Guide (HTML) (PDF)

Login

- If you followed my advice about above the Hibernate logging, then your console will contain the following information – very useful.

```
instantiating session factory with properties: {

        hibernate.connection.username=sa,
        hibernate.connection.poolsize=40,
        hibernate.connection.password=,
        hibernate.dialect=net.sf.hibernate.dialect.HSQLDialect,
        hibernate.connection.url=jdbc:hsqldb:C:/dev/data/workspaces/pesosPortal/GSProject/webapps/gridsphere/WEB-INF/database/gridsphere,
        hibernate.show_sql=false,
        hibernate.connection.driver_class=org.hsqldb.jdbcDriver
}
```