Windows XP Setup Guide for Downloading, Developing, Deploying and Debugging GeoNetwork within Eclipse

Dominic Owen
Arizona Geological Survey
2/26/2010

Purpose:

With the help of this document, developing for GeoNetwork within Eclipse becomes a much easier task.

The setup provided in this tutorial results in the following capabilities:

Interface with GeoNetwork SVN repository through Eclipse (using Subclipse).
Develop for GeoNetwork in Eclipse.
Deploy GeoNetwork from Tomcat inside Eclipse.
Debug GeoNetwork inside Eclipse (Tomcat debug mode allows you to step through java code as GeoNetwork executes).

Also of note, this setup relies upon a Mysql dbms and Tomcat instead of Jetty.

** This tutorial aims to cover every setup step involved, so you may skip any of the early setup steps that your computer is already compliant for.*

1) Install JDK 1.5 (Java SDK)

http://java.com/en/download/faq/other_jreversions.xml

2) Install Tomcat 6

http://tomcat.apache.org/download-60.cgi

3) Install Eclipse Java EE version And Subclipse

Eclipse Download URL:
http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/galileo/SR1/eclipse-jee-galileo-SR1-win32.zip

** NOTE: When setting up your GeoNetwork workspace, ensure you run from a path with no spaces. I.E.* "Documents and Settings" *will break things. For example:*
C:\geonetwork_workspace or C:\workspace are good workspace names.

Subclipse Setup Instructions (you have to scroll down the linked page to find the instructions):
http://subclipse.tigris.org/servlets/ProjectProcess?pageID=p4wYuA

4) GeoNetwork Eclipse/Subclipse Setup:

a) In Eclipse, browse to *File -> New -> Other -> SVN -> Checkout Projects from SVN* and hit Next.

b) Select “Create a new repository location” and click Next.

c) Enter the URL: https://geonetwork.svn.sourceforge.net/svnroot/geonetwork/trunk
and click Next.

d) Select the root of the provided directory structure (It should say:
https://geonetwork.svn.sourceforge.net/svnroot/geonetwork/trunk) and select Next.

e) Select “Check out as a project configured using the New Project Wizard” and click Finish.

f) The New Project Wizard will now appear. Browse to *Java->Java Project* and select Next. Name your project something similar to “Geonetwork.” Select Finish.

You will receive a Confirm Overwrite warning. Select OK. Eclipse will now download the trunk from the SVN repository; this may take up to 30 minutes.

g) You will initially only have one source folder. You need to add jeeves/src and cachingxslt/src folders as well.

To do so, right click the root of the project, and navigate to *New -> Source Folder*. For Project Name: select your project. For folder name, browse to jeeves/src and select OK. Next, select Finish. Do the same for cachingxslt/src.

h) The project will have many errors. You now need to some add jars to your classpath in Eclipse.

To add jars, right click on your project, and navigate to *Build Path -> Configure Build Path….* From the libraries tab, select add external JARs (Libraries tab > Add External Jars…). Find the following jar files, and add them all to your build path (there will be several duplicate jars, but it is easier to include duplicates than specify each jar from each folder to add):

jetty/lib/**/*.jar
web/geonetwork/WEB-INF/lib/*.jar
web/geoserver/WEB-INF/lib/*.jar
web/intermap/WEB-INF/lib/acme.jar

i) In some cases, you will have remaining “Access restriction” errors that can be resolved by removing and re-adding the "JRE System Library" under *Build Path -> Configure Build Path…. -> Libraries.*

j) Right click on your project's base directory, and browse to Run As -> Run Configurations... Browse to the Arguments tab and at the bottom of VM arguments add the following lines:
-XX:MaxPermSize=256m
-XX:PermSize=128m
-Xms48M
-Xmx512M

5) Install MySQL:

a) Download MySQL from: http://dev.mysql.com/downloads/mirror.php?id=379954

b) Do a default installation accepting the default settings. Remember to make a note of the password for future reference.

6) Configure MySQL for GeoNetwork

a) After installation, open up the MySQL Command Line Client. Enter your password.

b) Next enter the following two commands (second one is optional):
grant all privileges on geonetwork. to 'geonetwork'@'localhost' identified by 'password';*
grant all privileges on geonetwork. to 'geonetwork'@'159.87.39.14' identified by 'password';*
where 159.87.39.14 is your machine's IP address

7) Configure GeoNetowork for MySQL

- In your GeoNetwork install directoy, browse to:
<path to Geonetwork>/*geonetwork/web/geonetwork/WEB-INF/config.xml*

Find the <resources> node and its children. Make the changes outlined below:

```
<resources>
	<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
	<!-- mckoi standalone -->
	<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->

	<resource enabled="false">
		<name>main-db</name>
		<provider>jeeves.resources.dbms.DbmsPool</provider>
		<config>
			<user>xRgAPQLl</user>
			<password>X7ByXqvJ</password>
			<driver>com.mckoi.JDBCDriver</driver>
			<url>jdbc:mckoi://localhost:9157/</url>
			<poolSize>10</poolSize>
		</config>

	<activator
class="org.fao.geonet.activators.McKoiActivator"><configFile>WEB-
INF/db/db.conf</configFile></activator></resource>

	<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
	<!-- mysql -->
	<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
```

```
      <resource enabled="true">
            <name>main-db</name>
            <provider>jeeves.resources.dbms.DbmsPool</provider>
            <config>
                  <user>geonetwork</user>
                  <password>password</password>
                  <driver>com.mysql.jdbc.Driver</driver>
                  <url>jdbc:mysql://localhost:3306/geonetwork</url>
                  <poolSize>10</poolSize>
                  <reconnectTime>3600</reconnectTime>
            </config>
</resource>
```

8) Point Tomcat to GeoNetwork in Eclipse

a) The first step is to import Tomcat into your Eclipse workspace. To do so, browse to *New->Other->Server->Server* and select Next. Choose Apache then Tomcat v6.0 and select Next. Now browse to your Tomcat 6.0 installation (Mine is C:\Program Files\Apache Software Foundation\Tomcat 6.0), then select OK. Now click Finish.

b) Now click the server tab in the Eclipse workbench. Double click your server and then select the Modules tab. Select Add External Web Module.

For Document base, browse to <path to GeoNetwork in project workspace>\web\geonetwork> and select OK. For path, enter "/geonetwork"

Do the same for intermap and geoserver. For example, my three Web modules are defined as follows:

Path: C:\workspace\Geonetwork\web\geonetwork
Document base: /geonetwork

Path: C:\workspace\Geonetwork\web\geoserver
Document base: /geoserver

Path: C:\workspace\Geonetwork\web\intermap
Document base: /intermap

9) Update Some Configuration Files.

In <path to geoserver directory>/web/geoserver/WEB-INF/web.xml:
Set the value of the parameter 'GEOSERVER_DATA_DIR' to: '<your path to geoserver directory>/data/geoserver_data'.

Next, open web\intermap\WEB-INF\mapServers.xml. Locate the <default name="Layers for default map -- DUMMY NAME:not used"> element. Update it to look like this:

```
<default name="Layers for default map -- DUMMY NAME:not used">
        <server url="http://localhost:8080/geoserver/wms" type="2">
           <layer name="gn:world" />
        </server>

        <server url="http://localhost:8080/geoserver/wms" type="2">
           <layer name="gn:gboundaries" />
        </server>
</default>
```

10) Populate the Database

a) To link GeoNetwork to your database and populate the database, you must first make a quick configuration file (currently a needed configuration file is not attached to the trunk). Thus, navigate to <directory to Geonetwork>/bin and create a file called jetty.xml (make sure windows thinks it is an xml file, not a text document). In the file, paste the contents specified at the end of this document in Appendix A.

b) Next, in eclipse right click the build.xml file at the bottom of your GeoNetwork project. Select *Run As->Ant Build*. This creates the gast.jar file used in the next step.

c) Open GAST (GeoNetwork Administrator Survival Tool):
To do so, open a command line and browse to <GeoNetwork install directory>/gast and type java –jar gast.jar. GAST will now open.

d) In GAST, navigate to *Options->Config* and select Use this account. Enter GeoNetwork's admin username and password then press OK (GAST will not save these settings, so if you have any errors in GAST, go back to config and make sure your settings are still set). Set user and pass to ‘admin’ (or whatever the password is, if you changed the default setting).

e) From DBMS Configuration select MySQL and enter the following:

Server: localhost (if MySQL is on same server as Geonetwork)
Database: geonetwork
Username: admin
Password: password (or the password used in the MySQL command entered above identified by ‘password’;).

You can leave port empty. Click save.

f) Go to Database Setup, and press Setup (ignore the Cyclic error if it appears press OK).

g) Start Tomcat (right click your Tomcat server in eclipse and choose Start).

h) Go to Database Sample data and click Import.

Close GAST.

i) Right click your Tomcat server in Eclipse and choose Restart Tomcat. Once this is complete, browse to http://localhost:8080/geonetwork/ - GeoNetwork should be up and running! You can now debug GeoNetwork just like you would any other Java application in Eclipse (you need to start Tomcat in debug mode to do so).

Note: Lastly, if you make changes to GeoNetwork's Java files, you need to run build.xml in order for your changes to be detected by the Eclipse Tomcat debugger.

Appendix A: jetty.xml file (Copy text below)

```
<?xml version="1.0"?>
<!DOCTYPE Configure PUBLIC "-//Mort Bay Consulting//DTD Configure//EN"
"http://jetty.mortbay.org/configure.dtd">

<!-- =============================================================== -->
<!-- Configure the Jetty Server                                      -->
<!--                                                                 -->
<!-- Documentation of this file format can be found at:              -->
<!-- http://docs.codehaus.org/display/JETTY/jetty.xml                -->
<!--                                                                 -->
<!-- =============================================================== -->


<Configure id="Server" class="org.mortbay.jetty.Server">

    <!-- =========================================================== -->
    <!-- Server Thread Pool                                          -->
    <!-- =========================================================== -->
    <Set name="ThreadPool">

      <New class="org.mortbay.thread.QueuedThreadPool">
        <Set name="minThreads">10</Set>
        <Set name="maxThreads">200</Set>
        <Set name="lowThreads">20</Set>
        <Set name="SpawnOrShrinkAt">2</Set>
      </New>

      <!-- Optional Java 5 bounded threadpool with job queue
      <New class="org.mortbay.thread.concurrent.ThreadPool">
        <Set name="corePoolSize">50</Set>
        <Set name="maximumPoolSize">50</Set>
      </New>
      -->
    </Set>



    <!-- =========================================================== -->
    <!-- Set connectors                                              -->
    <!-- =========================================================== -->
    <!-- One of each type!                                           -->
    <!-- =========================================================== -->

    <!-- Use this connector for many frequently idle connections
         and for threadless continuations.
    -->
    <Call name="addConnector">
      <Arg>
          <New class="org.mortbay.jetty.nio.SelectChannelConnector">
            <Set name="host"><SystemProperty name="jetty.host"
default="localhost"/></Set>
            <Set name="port"><SystemProperty name="jetty.port"
default="8080"/></Set>
```

```
            <Set name="maxIdleTime">30000</Set>
            <Set name="Acceptors">2</Set>
            <Set name="statsOn">false</Set>
            <Set name="confidentialPort">8443</Set>
                <Set name="lowResourcesConnections">5000</Set>
                <Set name="lowResourcesMaxIdleTime">5000</Set>
          </New>
      </Arg>
    </Call>

    <!-- Use this connector if NIO is not available.
    <Call name="addConnector">
      <Arg>
          <New class="org.mortbay.jetty.bio.SocketConnector">
            <Set name="port">8081</Set>
            <Set name="maxIdleTime">50000</Set>
            <Set name="lowResourceMaxIdleTime">1500</Set>
          </New>
      </Arg>
    </Call>
    -->

    <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
    <!-- To add a HTTPS SSL listener                                        -->
    <!-- see jetty-ssl.xml to add an ssl connector. use                     -->
    <!-- java -jar start.jar etc/jetty.xml etc/jetty-ssl.xml                -->
    <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->

    <!-- =========================================================== -->
    <!-- Set up global session ID manager                            -->
    <!-- =========================================================== -->
    <!--
    <Set name="sessionIdManager">
      <New class="org.mortbay.jetty.servlet.HashSessionIdManager">
        <Set name="workerName">node1</Set>
      </New>
    </Set>
    -->

    <!-- =========================================================== -->
    <!-- Set handler Collection Structure                            -->
    <!-- =========================================================== -->
    <Set name="handler">
      <New id="Handlers" class="org.mortbay.jetty.handler.HandlerCollection">
        <Set name="handlers">
         <Array type="org.mortbay.jetty.Handler">
           <Item>
             <New id="Contexts"
class="org.mortbay.jetty.handler.ContextHandlerCollection"/>
           </Item>
           <Item>
             <New id="DefaultHandler"
class="org.mortbay.jetty.handler.DefaultHandler"/>
           </Item>
           <Item>
             <New id="RequestLog"
class="org.mortbay.jetty.handler.RequestLogHandler"/>
```

```
          </Item>
         </Array>
        </Set>
      </New>
    </Set>

    <!-- =============================================================== -->
    <!-- Configure the context deployer                                  -->
    <!-- A context deployer will deploy contexts described in            -->
    <!-- configuration files discovered in a directory.                  -->
    <!-- The configuration directory can be scanned for hot              -->
    <!-- deployments at the configured scanInterval.                     -->
    <!--                                                                 -->
    <!-- This deployer is configured to deploy contexts configured       -->
    <!-- in the $JETTY_HOME/contexts directory                           -->
    <!--                                                                 -->
    <!-- =============================================================== -->
<!--    <Call name="addLifeCycle">
      <Arg>
        <New class="org.mortbay.jetty.deployer.ContextDeployer">
          <Set name="contexts"><Ref id="Contexts"/></Set>
          <Set name="configurationDir"><SystemProperty name="jetty.home"
default="."/>/../web</Set>
          <Set name="scanInterval">5</Set>
        </New>
      </Arg>
    </Call>
-->
    <!-- =============================================================== -->
    <!-- Configure the webapp deployer.                                  -->
    <!-- A webapp  deployer will deploy standard webapps discovered      -->
    <!-- in a directory at startup, without the need for additional      -->
    <!-- configuration files.    It does not support hot deploy or       -->
    <!-- non standard contexts (see ContextDeployer above).              -->
    <!--                                                                 -->
    <!-- This deployer is configured to deploy webapps from the          -->
    <!-- $JETTY_HOME/webapps directory                                   -->
    <!--                                                                 -->
    <!-- Normally only one type of deployer need be used.                -->
    <!--                                                                 -->
    <!-- =============================================================== -->
    <Call name="addLifeCycle">
      <Arg>
        <New class="org.mortbay.jetty.deployer.WebAppDeployer">
          <Set name="contexts"><Ref id="Contexts"/></Set>
          <Set name="webAppDir"><SystemProperty name="jetty.home"
default="."/>/../web</Set>
          <Set name="parentLoaderPriority">true</Set>
              <!--  Refer to
http://geoserver.org/display/GEOSDOC/Dealing+with+native+JAI
                        for more background on the parentLoaderPriority=True
-->
              <Set name="extract">true</Set>
              <Set name="allowDuplicates">false</Set>
          <Set name="defaultsDescriptor"><SystemProperty name="jetty.home"
default="."/>/../bin/webdefault.xml</Set>
        </New>
```

```
      </Arg>
    </Call>

    <!-- =============================================================== -->
    <!-- Configure Authentication Realms                                 -->
    <!-- Realms may be configured for the entire server here, or         -->
    <!-- they can be configured for a specific web app in a context      -->
    <!-- configuration (see $(jetty.home)/contexts/test.xml for an       -->
    <!-- example).                                                       -->
    <!-- =============================================================== -->
  <!--  <Set name="UserRealms">
      <Array type="org.mortbay.jetty.security.UserRealm">
        <Item>
          <New class="org.mortbay.jetty.security.HashUserRealm">
            <Set name="name">Test Realm</Set>
            <Set name="config"><SystemProperty name="jetty.home"
default="."/>/etc/realm.properties</Set>
            <Set name="refreshInterval">0</Set>
          </New>
        </Item>
      </Array>
    </Set>-->

    <!-- =============================================================== -->
    <!-- Configure Request Log                                           -->
    <!-- Request logs  may be configured for the entire server here,     -->
    <!-- or they can be configured for a specific web app in a           -->
    <!-- contexts configuration (see $(jetty.home)/contexts/test.xml     -->
    <!-- for an example).                                                -->
    <!-- =============================================================== -->
    <Ref id="RequestLog">
      <Set name="requestLog">
        <New id="RequestLogImpl" class="org.mortbay.jetty.NCSARequestLog">
          <Set name="filename"><SystemProperty name="jetty.logs"
default="./logs"/>/yyyy_mm_dd.request.log</Set>
          <Set name="filenameDateFormat">yyyy_MM_dd</Set>
          <Set name="retainDays">90</Set>
          <Set name="append">true</Set>
          <Set name="extended">false</Set>
          <Set name="logCookies">false</Set>
          <Set name="LogTimeZone">GMT</Set>
        </New>
      </Set>
    </Ref>

    <!-- =============================================================== -->
    <!-- extra options                                                   -->
    <!-- =============================================================== -->
    <Set name="stopAtShutdown">true</Set>
    <Set name="sendServerVersion">true</Set>
    <Set name="sendDateHeader">true</Set>
    <Set name="gracefulShutdown">1000</Set>

</Configure>
```