

Developer Guide

The developer environment can be set in any Operating System (Linux, Windows, etc.) since it is a multi-platform system, but Linux is recommended, because is what it is being used.

You need to install Maven, eclipse IDE and some plugins. For OpenKM 4.0 and 5.0 you have these requirements:

- Maven at <http://maven.apache.org/>
- Eclipse Galileo (Eclipse IDE for Java EE Developers) at <http://www.eclipse.org/galileo/>
- Subversion eclipse plugin at <http://subclipse.tigris.org>
- Maven to eclipse plugin at <http://m2eclipse.sonatype.org>
- Java JDK 1.6 at <http://java.sun.com/javase/downloads/index.jsp>



Starting from Eclipse Helios you have a Marketplace accessible from Help > Eclipse Marketplace... Here you can search for "maven" to install the "Maven Integration for Eclipse" and "Maven Integration for Eclipse (Extras)" solutions. Also you can search for "subversion" to find the "Subclipse" plugin.

Once all them are installed, you can download OpenKM and build it.

- Configure Java in Eclipse
- Configure JBoss server in Eclipse
- Maven installation
- Checkout source from Subversion
- Development tips
- Browsing embeded database
- Debugging OpenKM
- Debugging remote server
- Debugging with GWT
- Doxygen OpenKM 5.0.x ^[1] (OpenKM packages & classes & files documentation)
- Doxygen OpenKM 5.1.x ^[2] (OpenKM packages & classes & files documentation)

Note: You can configure Eclipse to integrate with MantisBT. Read Mylyn-Mantis Repository Connecto ^[3] for more info.

Eclipse Development Quick Install Guide

This quick install is valid for Eclipse Indigo.

1. Download **Eclipse IDE for Java EE Developers** from <http://www.eclipse.org/downloads/>.
2. Go to Help > Eclipse Marketplace and install these plugins:
 1. Maven Integration for Eclipse ^[4]
 2. Subclipse ^[5]

Alternative: Add repository

- Subclipse 1.6.x Update Site -> http://subclipse.tigris.org/update_1.6.x
- Maven Integration for Eclipse - <http://m2eclipse.sonatype.org/sites/m2e/>
- Maven Integration for Eclipse Extras - <http://m2eclipse.sonatype.org/sites/m2e-extras/>

Changelog

Changes between developer guide 4.0 to 5.0

- Development tips changes (how to enable OpenKM extensions in compilation and disable automatic GWT compilation)
- GWT 2.0.4
- Java package has been refactoring to "com.openkm" all references to older package in configuration files has been changed

Changes between developer guide 2.0 to 4.0

- Changed to Eclipse Galileo
- Now we use maven in replacement of JBoss Tools

Changes between developer guide 1.2 to 2.0

- Changed IDE to Eclipse Europa / JBoss tools, before was JBoss IDE.
- GWT 1.5.3 is needed to compile, before was version 1.4.6
- Changed packaging to new jboss tools packaging.
- Changed GWT generating to temporary JBoss folder to project folders to developing.
- Added new generateback.sh and generateback.bat to GWT admin.
- Upgraded JBoss server to version 4.2.2.GA

Changes between developer guide 1.0 and 1.1 to 1.2

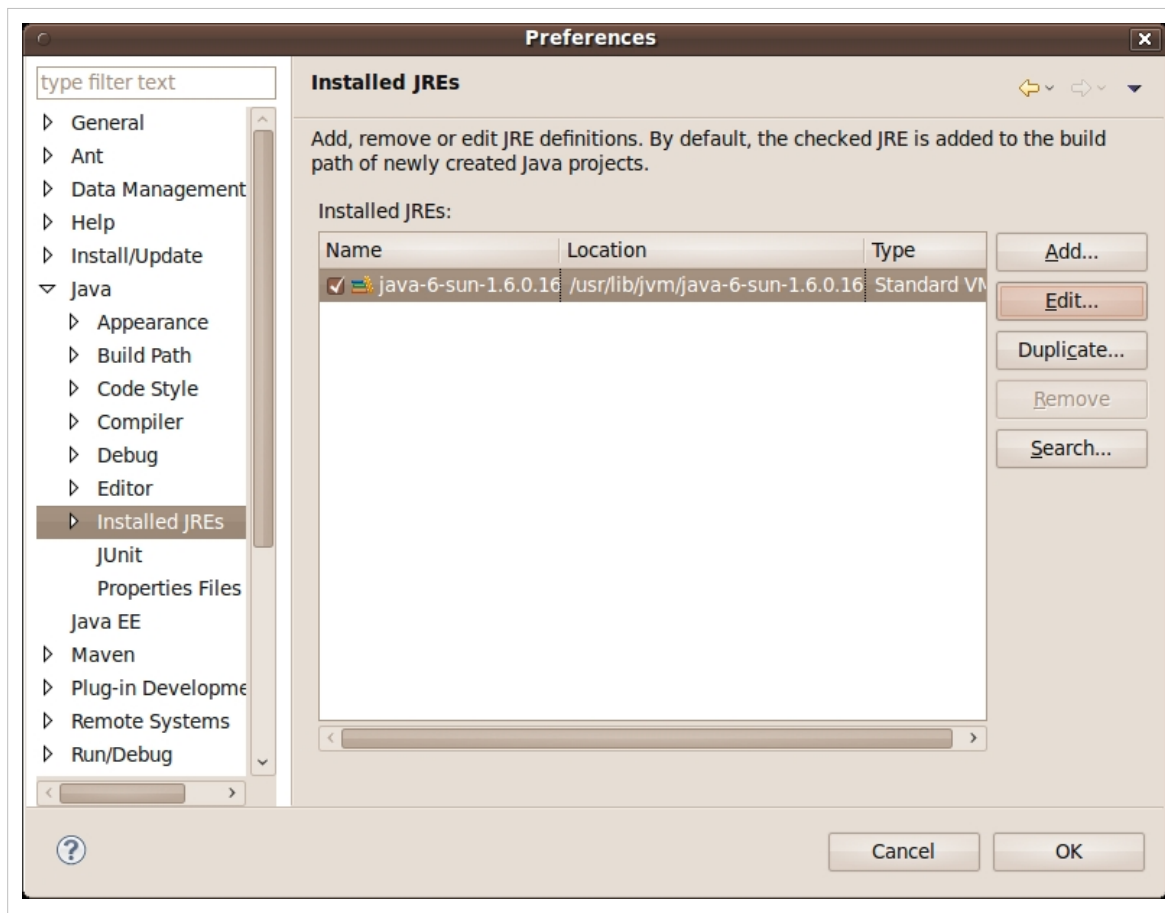
- GWT 1.4.6 is needed to compile, before was version 1.3.3.
- Changes on GWT shell configuration, now not uses GWTDS variable (code deleted).
- Configured generate.sh to deploy to tmp JBoss with OpenKM.ear deployed to fast developing.
- Added gecko to Main.gwt.xml to fast compiling only gecko when is uncommented.

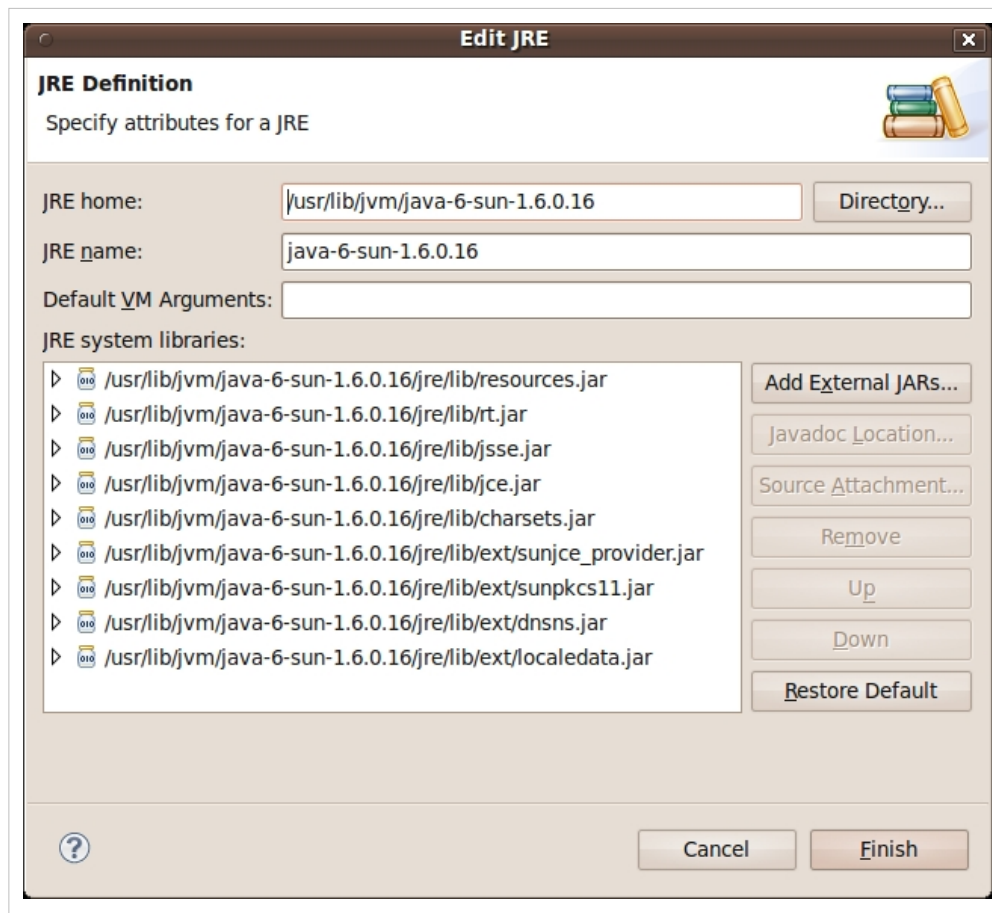
References

- [1] <http://doxygen.openkm.com/5.0.x>
 - [2] <http://doxygen.openkm.com/5.1.x>
 - [3] http://sourceforge.net/apps/mediawiki/mylyn-mantis/index.php?title=Main_Page
 - [4] <http://marketplace.eclipse.org/content/maven-integration-eclipse>
 - [5] <http://marketplace.eclipse.org/content/subclipse>
-

Configure Java in Eclipse

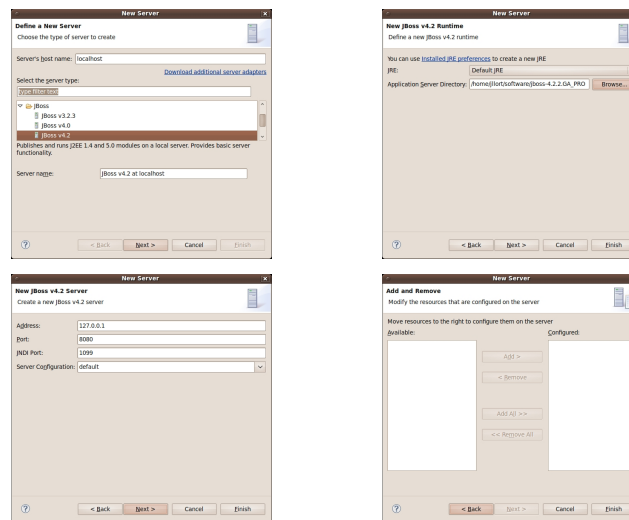
Go to **Window / Preferences / Java / Installed JRE**. You must have JDK 1.6 defined here:





Configure JBoss server in Eclipse

Go to servers tab Using contextual menus select **Add / server**. Select JBoss server:



Maven installation

Maven is a software tool for Java project management and build automation. It is similar in functionality to the Apache Ant tool, but is based on different concepts. Maven is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.

Maven uses a construct known as a Project Object Model (POM) to describe the software project being built, its dependencies on other external modules and components, and the build order. It comes with pre-defined targets for performing certain well defined tasks such as compilation of code and its packaging.

In GNU/Linux you can add the Maven executable to the binary path editing the `$HOME/.bashrc` file and adding this line at the end:

```
export PATH=$PATH:/path/to/maven/installation/bin
export MAVEN_OPTS="-Xmx512m"
```

Learn more about Maven at <http://maven.apache.org>.

Register missing Maven dependencies

Although project dependencies usually at located at Maven repositories, sometimes there are missing artifacts. So, we have to provide these libraries registering them in the local Maven repository. Follow these steps to register some libraries needed by OpenKM:

- Open an operating system console.
- Go to OpenKM project path in your local file system.
- Drive into *ext-lib*s and execute `install.sh` (Windows users simply open this file with text editor and execute the command)



Starting with OpenKM 5.0, you don't need to register manually these dependencies.

Some Maven tips

Download sources and javadoc from libraries dependency to improve the developer experience:

```
$ mvn install -DdownloadSources=true -DdownloadJavadocs=true
```

You can make Maven ignore test failures in the build process this way:

```
$ mvn install -Dmaven.test.failure.ignore=true
```

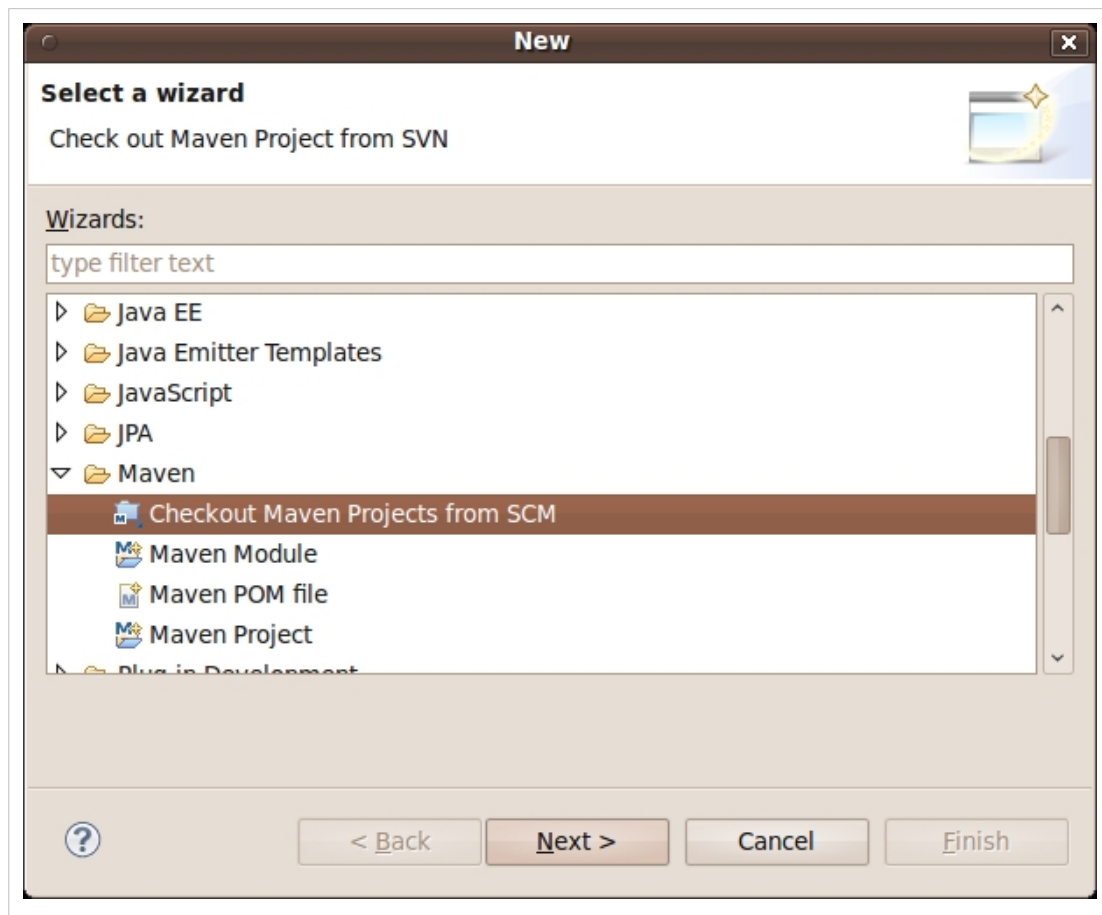
If you would like to skip tests, you can do so by executing the following command:

```
$ mvn install -Dmaven.test.skip=true
```

Checkout source from Subversion

Go to **File / New / Other** and select **Maven / Checkout Maven Projects from SCM**:

Hint: If your SCM dropdown list is empty, install the Maven Subclipse extension from (<http://m2eclipse.sonatype.org/sites/m2e-extras/>).



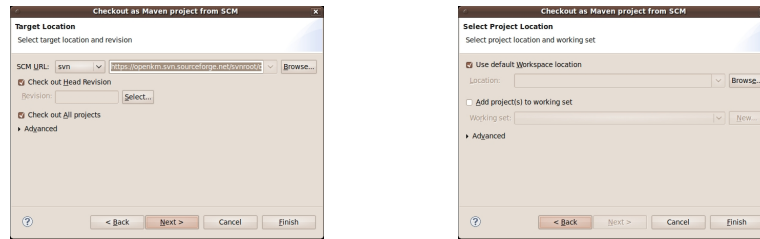
Select the **svn** type and type the url <https://openkm.svn.sourceforge.net/svnroot/openkm/trunk/openkm> to refer actual development:

Select the **svn** type and type the url <https://openkm.svn.sourceforge.net/svnroot/openkm/branches/5.1/openkm> to refer version 5.1:

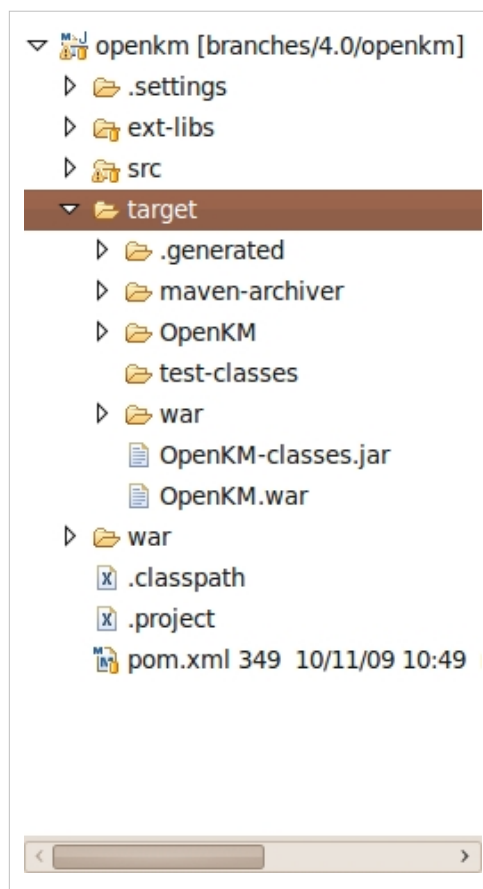
Select the **svn** type and type the url <https://openkm.svn.sourceforge.net/svnroot/openkm/branches/5.0/openkm> to refer version 5.0:

Select the **svn** type and type the url `https://openkm.svn.sourceforge.net/svnroot/openkm/branches/4.1/openkm` to refer version 4.1:

Select the **svn** type and type the url `https://openkm.svn.sourceforge.net/svnroot/openkm/branches/4.0/openkm` to refer version 4.0:



Be patient, first time you downloading OpenKM from svn, you're downloading all libraries to your maven repository. It could take several minutes depending your Internet connection.



After OpenKM will be downloaded, and automatically compiled.



If you want native support for Subversion, you have to install the *libsvn-java* package in Linux. The edit the *eclipse.ini* configuration file and add the following line:

```
-Djava.library.path=/usr/lib/jni
```

Development tips OpenKM 4.0

We recommend downloading JBoss+OpenKM from SourceForge, remove *OpenKM.war* and develop with this JBoss server configuration. For better development we recommend not deploying *OpenKM.war* file it's better setting in *\$JBoss_HOME/server/default/deploy* some alias to *target/OpenKM* folder:

▷ jboss-web.deployer	11 elementos	carpeta	mié 18 mar 2009 11:27
▷ jbossws.sar	15 elementos	carpeta	jue 29 ene 2009 13:11
▷ jms	9 elementos	carpeta	mié 29 oct 2008 12:56
▷ jmx-console.war	11 elementos	carpeta	mié 29 oct 2008 12:56
▷ juddi-service.sar	6 elementos	carpeta	jue 29 ene 2009 13:11
▷ management	1 elemento	carpeta	lun 22 oct 2007 11:43
▷ OpenKM.war	16 elementos	Enlace hacia carpeta	jue 21 ene 2010 12:32
▷ ota_jaxb.war	3 elementos	carpeta	mar 03 feb 2009 13:43
▷ uuid-key-generator.sar	2 elementos	carpeta	lun 22 oct 2007 11:43
bsh-deployer.xml	405 bytes	documento XML	lun 22 oct 2007 11:43
cache-invalidation-s...	2,0 KiB	documento XML	lun 22 oct 2007 11:43
client-deployer-servi...	1,9 KiB	documento XML	lun 22 oct 2007 11:43
database-ds.xml	5,3 KiB	documento XML	mar 25 ago 2009 13:34

In case you're making strong changes in OpenKM UI (GWT) we recommend disabling pom compile directive

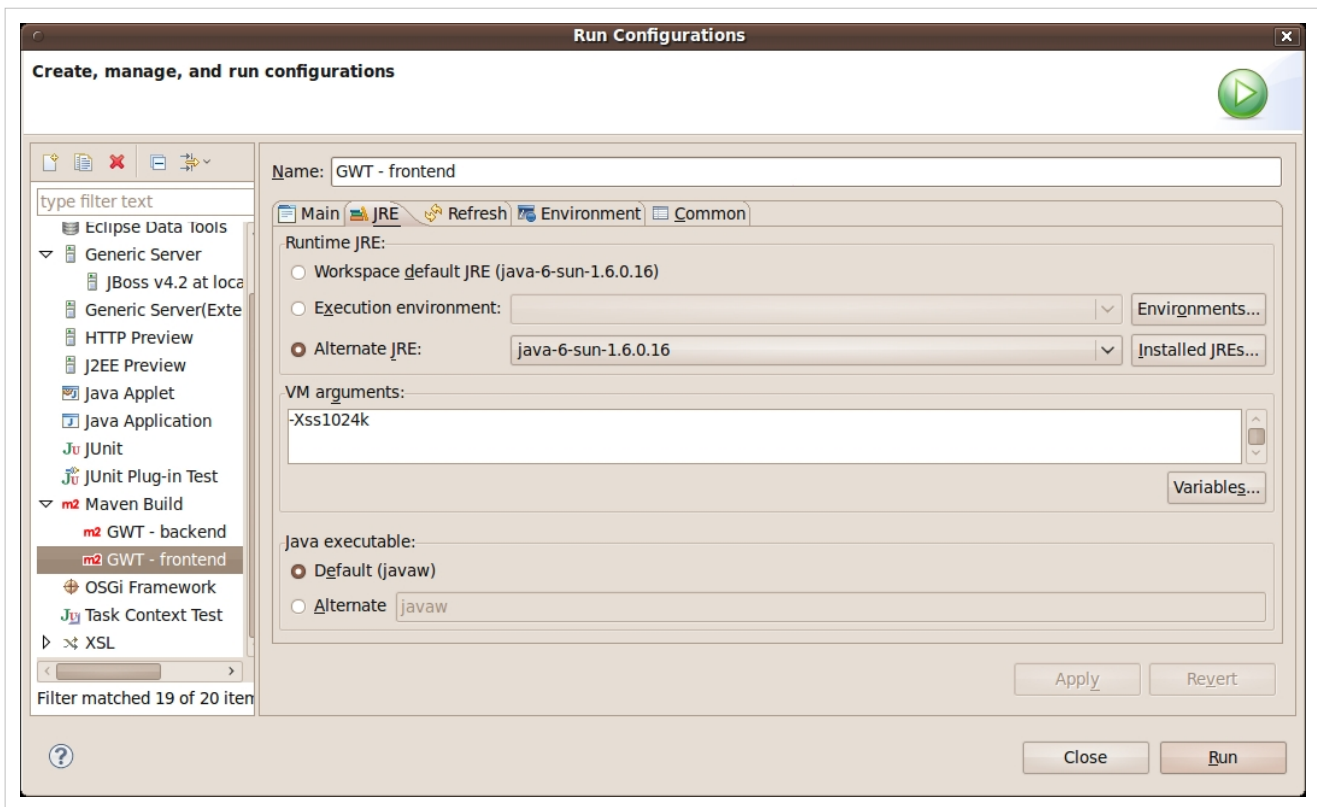
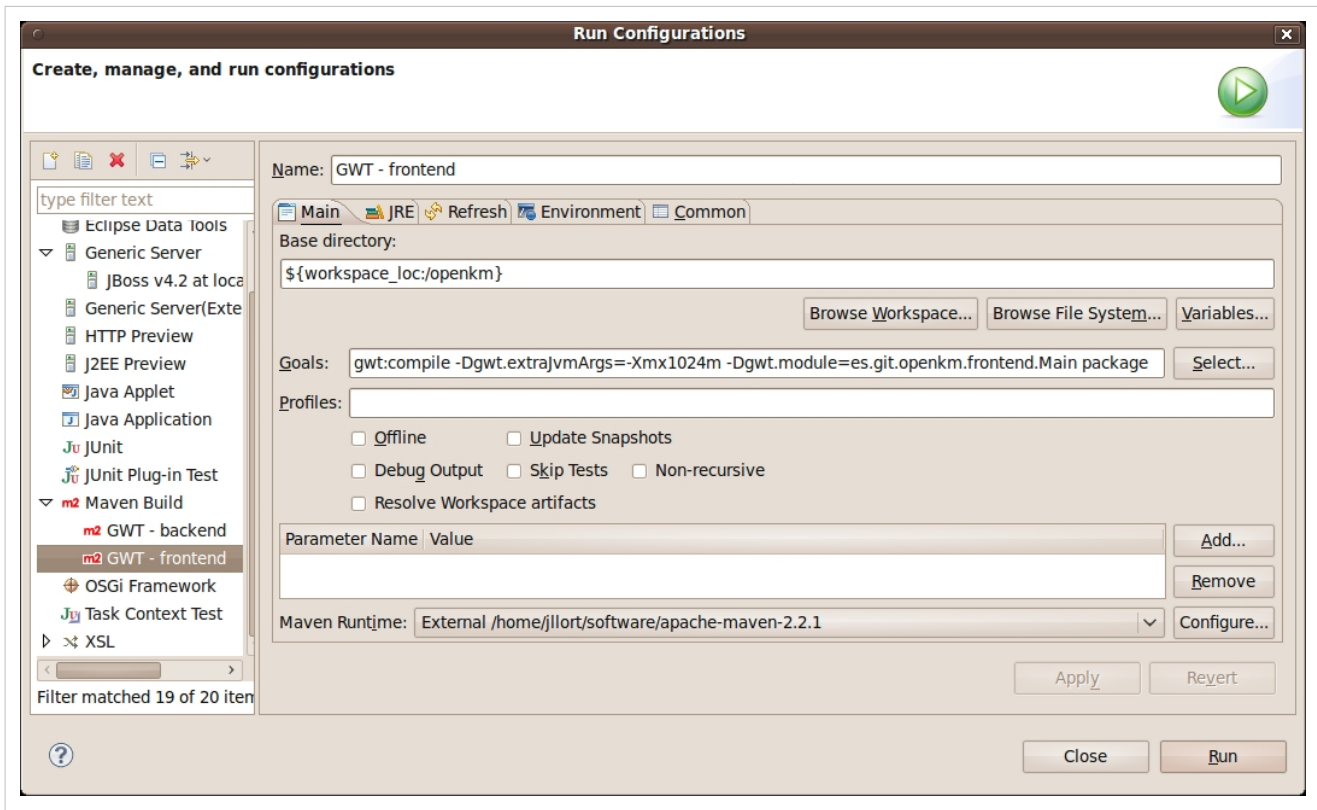
```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>gwt-maven-plugin</artifactId>
  <version>1.1</version>
  <configuration>
    <runTarget>es.git.openkm.frontend.Main/index.html</runTarget>
  </configuration>
  <executions>
    <execution>
      <goals>
        <!-- <goal>compile</goal> -->
        <!-- <goal>generateAsync</goal> -->
        <goal>test</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```



Is possible to compile GWT modules individually by this way:

```
$ mvn gwt:compile -Dgwt.module=com.openkm.backend.Main
```

You might be interested in running frontend and backend GWT packaging from eclipse. Go to run configurations and add in maven build



To achieve a faster compile we use to development Firefox, uncommenting gecko line on file Main.gwt.xml GWT compilation only for gecko browser runs fine. Don't forget at finish to comment it and try application with IE or other browsers. Now are two files called Main.gwt.xml one for frontend and other to backend.

```
<!-- Compile for Firefox only -->
<set-property name="user.agent" value="gecko"/>
```

Development tips OpenKM 5.0

We recommend downloading JBoss+OpenKM from SourceForge, remove *OpenKM.war* and develop with this JBoss server configuration. For better development we recommend not deploying *OpenKM.war* file it's better setting in *\$JBoss_HOME/server/default/deploy* some alias to *target/OpenKM* folder:

▷ jboss-web.deployer	11 elementos	carpeta	mié 18 mar 2009 11:27
▷ jbossws.sar	15 elementos	carpeta	jue 29 ene 2009 13:11
▷ jms	9 elementos	carpeta	mié 29 oct 2008 12:56
▷ jmx-console.war	11 elementos	carpeta	mié 29 oct 2008 12:56
▷ juddi-service.sar	6 elementos	carpeta	jue 29 ene 2009 13:11
▷ management	1 elemento	carpeta	lun 22 oct 2007 11:43
▷ OpenKM.war	16 elementos	Enlace hacia carpeta	jue 21 ene 2010 12:32
▷ ota_jaxb.war	3 elementos	carpeta	mar 03 feb 2009 13:43
▷ uuid-key-generator.sar	2 elementos	carpeta	lun 22 oct 2007 11:43
bsh-deployer.xml	405 bytes	documento XML	lun 22 oct 2007 11:43
cache-invalidation-s...	2,0 KiB	documento XML	lun 22 oct 2007 11:43
client-deployer-servi...	1,9 KiB	documento XML	lun 22 oct 2007 11:43
database-ds.xml	5,3 KiB	documento XML	mar 25 ago 2009 13:34

It's mandatory registering **sample-5.0-full.jar** into maven repository. File is into **ext-libs** folder. It must be executed some maven command to registering (then compilation will execute right). Take a look at *install.sh*, if sometimes is required some new library, the reference it'll be included on that file.

```
mvn install:install-file -DgroupId=com.openkm.extension -DartifactId=sample-full -Dversion=5.0 -Dpackaging=jar -Dfile=sample-5.0-full.jar
```

In case you're making strong changes in OpenKM UI (GWT) we recommend disabling pom compile directive

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>gwt-maven-plugin</artifactId>
  <version>1.2</version>
  <executions>
    <execution>
      <goals>
        <goal>compile</goal>
        <!-- <goal>generateAsync</goal> -->
        <!-- <goal>test</goal> -->
      </goals>
    </execution>
  </executions>
</configuration>
```

```

<runTarget>com.openkm.frontend.Main/index.html</runTarget>
<modules>
  <module>com.openkm.frontend.Main</module>
</modules>
</configuration>
</plugin>

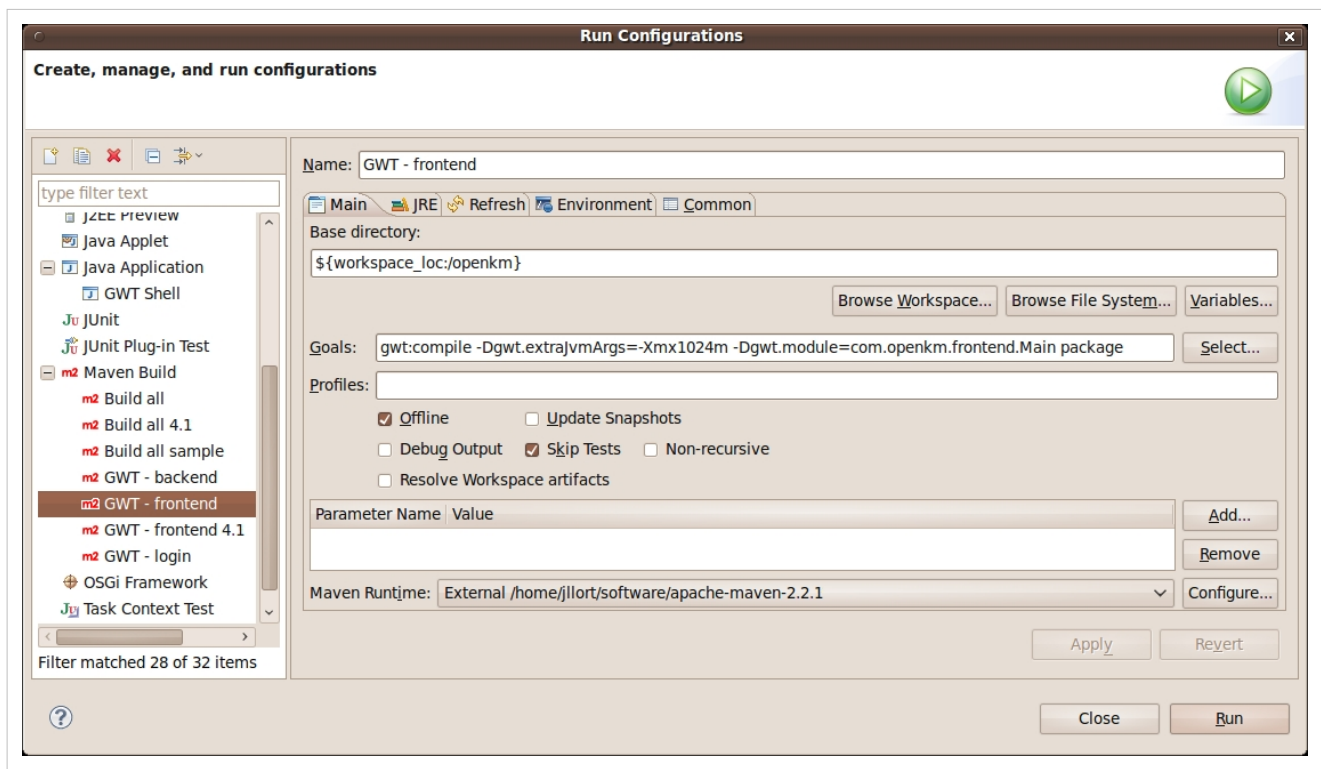
```

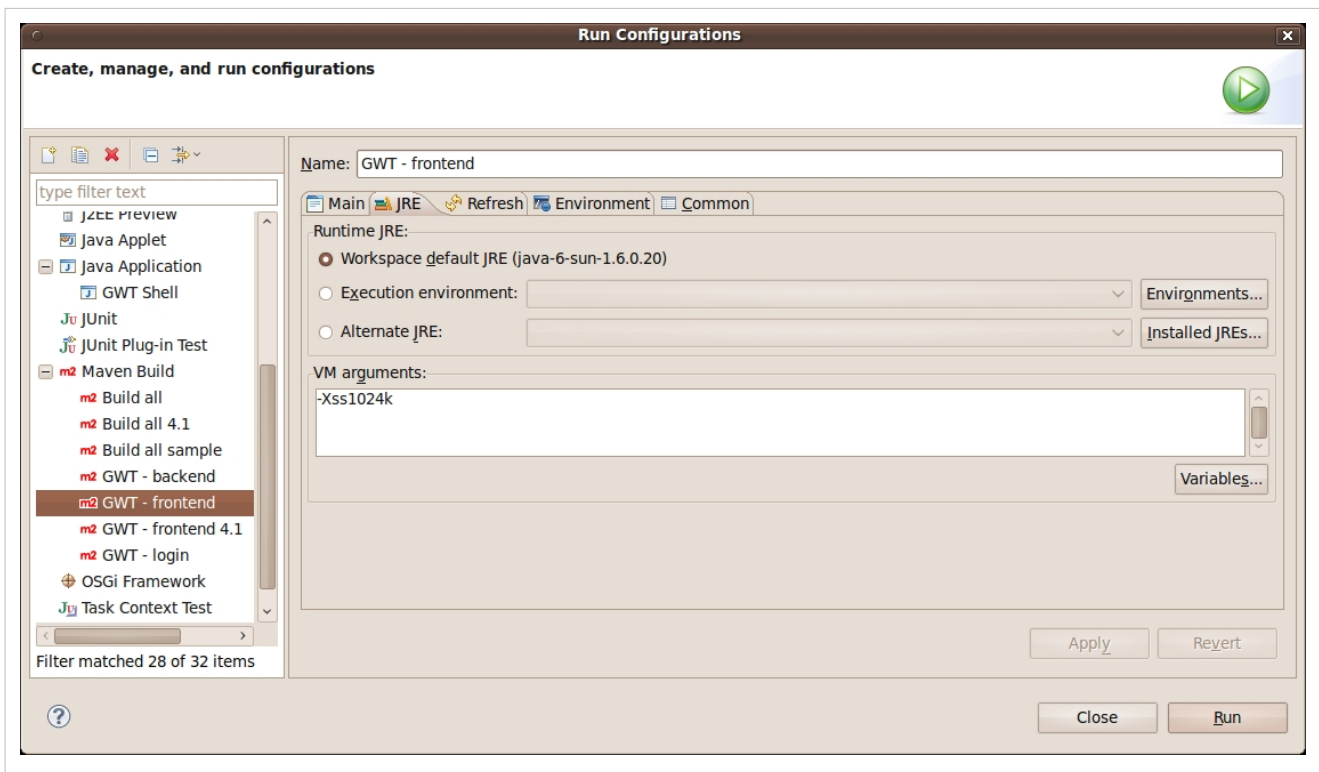


Is possible to compile GWT modules individually by this way:

```
$ mvn gwt:compile -Dgwt.module=com.openkm.frontend.Main
```

You might be interested in running frontend and backend GWT packaging from eclipse. Go to run configurations and add in maven build





To achieve a faster compile we use to development Firefox, uncommenting gecko line on file **Main.gwt.xml** GWT compilation only for gecko browser runs fine. Don't forget at finish to comment it and try application with IE or other browsers.

```
<!-- Compile for Firefox only -->
<set-property name="user.agent" value="gecko"/>
```

User contribution script

```
#!/bin/bash
# @author: Kenneth Walter

#Cleans and re-compiles the source files
mvn -Dmaven.test.skip=true clean gwt:compile install $*

#Set the TARGET_DIRECTORY to the path of your JBoss installation
TARGET_DIRECTORY=~/.Downloads/jboss-4.2.3.GA/
FILE_TO_MOVE=target/OpenKM.war
#This will only attempt to replace the existing WAR if the new WAR
exists
if [ -f $FILE_TO_MOVE ]
then
    echo 'Deploying WAR to JBoss Directory'
    cp -v $FILE_TO_MOVE $TARGET_DIRECTORY/server/default/deploy/.
    echo 'Done'
fi
```

More info about this script at Cannot create or use example extensions ^[1] forum thread.

References

[1] <http://forum.openkm.com/viewtopic.php?f=31&t=5332>

Browsing embeded database

OpenKM comes with FOUR embeded databases that are defined in poll openkm-ds.xml into \$JBoss_HOME/server/default/deploy directory

- OKMActivity (used for log info)
- OKMAuth (user for authentication purpose)
- OKMDashboardStats (used for dashboard info)
- OKMWorkflow (used to store workflow info)

A simply way to browse it

Check the JAVA_OPTS in bin/run.sh. java.awt.headless has to be false to use the embedded database browser.

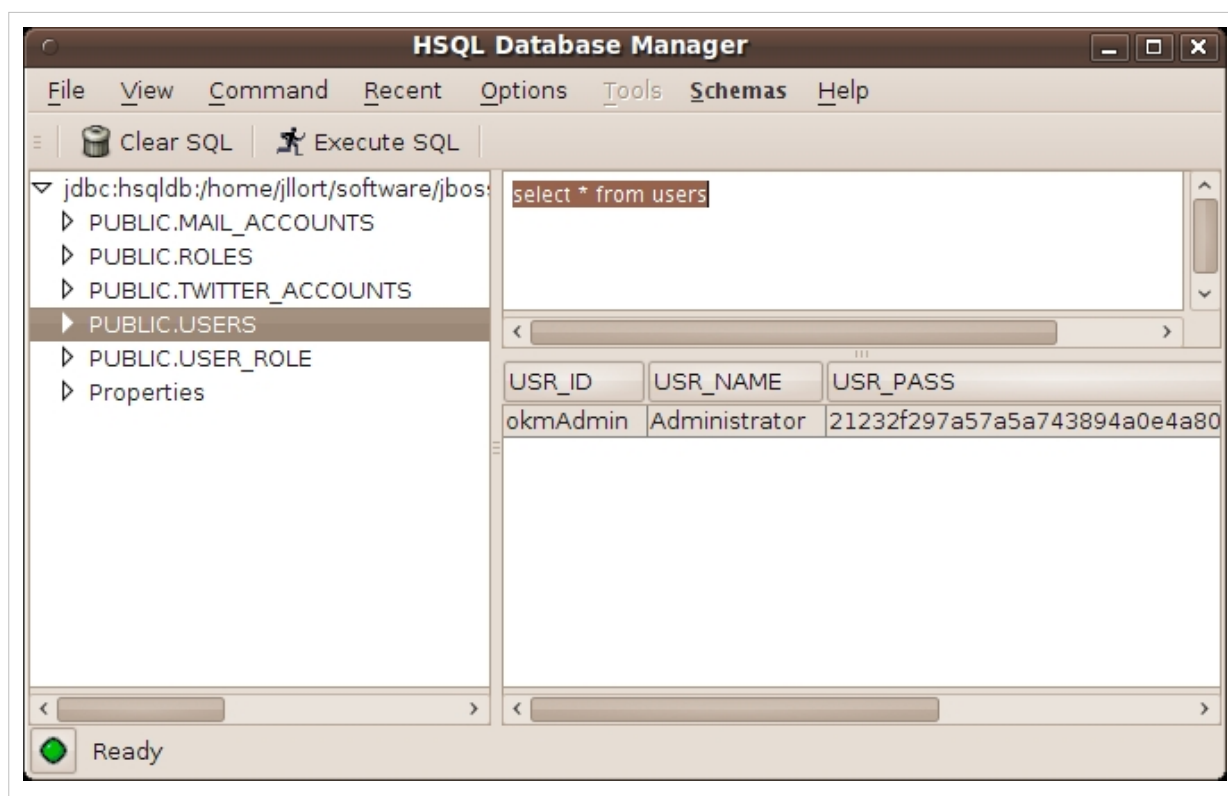
An example JAVA_OPTS could be:

```
JAVA_OPTS="-Xms256m -Xmx1024m -XX:PermSize=64m -XX:MaxPermSize=128m -Djava.awt.headless=false"
```

Open the url from server <http://localhost:8080/jmx-console/>. You'll see there three services defined under JBoss:

```
database=OKMActivity,service=Hypersonic  
database=OKMAuth,service=Hypersonic  
database=OKMDashboardStats,service=Hypersonic  
database=OKMWorkflow,service=Hypersonic  
database=localDB,service=Hypersonic
```

Select one datasource, then look for **void startDatabaseManager()** and press the **Invoke** button. You will see an screen like this:



More info at Starting hsqldb manager on JBoss ^[1].

References

- [1] <http://docs.jboss.org/jbpm/v3/userguide/thejbpmdatabase.html#d0e2677>

Debugging OpenKM

You can debug your OpenKM installation using the JBoss logging facility. This is an useful thing when you have problems with your configuration. Default OpenKM installation tries to log important events like errors and warnings. Is possible to change this configuration editing the file `$JBOSS_HOME/server/default/conf/jboss-log4j.xml`.

Default JBoss log configuration can generate a lot of messages. These files are stored at `$JBOSS_HOME/server/default/log`. It is configured to use the `DailyRollingFileAppender`. This appender create a new log file for every day. This is better than have a unique huge log file, os course. The rollover is performed at midnight each day, but you can configure it to make the rollover every hour (uncomment the proper line).

```
<!-- A time/date based rolling appender -->
<appender name="FILE"
    class="org.jboss.logging.appender.DailyRollingFileAppender">
    <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
    <param name="File" value="${jboss.server.log.dir}/server.log"/>
    <param name="Append" value="false"/>
    <!-- Rollover at midnight each day -->
    <param name="DatePattern" value="'.'yyyy-MM-dd"/>
    <!-- Rollover at the top of each hour
    <param name="DatePattern" value="'.'yyyy-MM-dd-HH"/>
-->
<layout class="org.apache.log4j.PatternLayout">
    <!-- The default pattern: Date Priority [Category] Message\n -->
    <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>
    <!-- The full pattern: Date MS Priority [Category] (Thread:NDC) Message\n
    <param name="ConversionPattern" value="%d %-5r %-5p [%c] (%t:%x) %m%n"/>
-->
</layout>
</appender>
```

You can reduce the amount of log messages produced by OpenKM, or can increase them. In this example we limit the log messages produced by the class `OKMAccessManager` for those of type `ERROR`.

```
<category name="com.openkm.core.OKMAccessManager">
    <priority value="ERROR" />
</category>
```

If you create this configuration:

```
<category name="com.openkm">
    <priority value="DEBUG" />
</category>
```

All the log messages generated by OpenKM will be shown. As you can see, you can increase debug messages in some parts of OpenKM to check a determinate behavior.

Debugging JAAS configuration

If you are trying to setup another authentication source different from the default provided by OpenKM, you can afford some problems. The JBoss login-config.xml is supposed to have the right configuration, but you can't log into the application. The most common case is a bad or missing JAAS configuration. So if you need to debug the JAAS, you can add to the log4j.xml file the following:

```
<category name="org.jboss.security">
  <priority value="TRACE" class="org.jboss.logging.XLevel"/>
  <appender-ref ref="SECURITY_F"/>
</category>
<appender name="SECURITY_F" class='org.jboss.logging.appender.DailyRollingFileAppender'>
  <param name="Append" value="true"/>
  <param name="DatePattern" value="'.'yyyy-MM-dd"/>
  <param name="File" value="${jboss.server.home.dir}/log/jboss.security.log"/>
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d{ABSOLUTE} %-5p [%c] %m%n"/>
  </layout>
</appender>
```

This is more or less in the middle of the file, just where the <category-name> section begins. And you should look at this new log file:

```
$ tailf -f $JBASS_HOME/server/default/log/jboss.security.log
```

Email error notification

Always is good idea to be notified when things goes wrong. There are some log appenders that can help you. The SMTPAppender will mail you log messages with threshold ERROR by default. You can lower this threshold, but you will got lots of useless mail messages. Here you must configure some properties:

- **To:** The mail account where the messages will arrive.
- **From:** You can set it simply as noreply@your-domain.com.
- **Subject:** Here you can specify the subject of the mail. If you have several OpenKM installations, you can create a filter in your mail client using this value.
- **SMTPHost:** The mail server server. Can be localhost if there is a mail server installed in this computer.

```
<!-- EMail events to an administrator -->
<appender name="SMTP" class="org.apache.log4j.net.SMTPAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="Threshold" value="ERROR"/>
  <param name="To" value="admin@myhost.domain.com"/>
  <param name="From" value="nobody@myhost.domain.com"/>
  <param name="Subject" value="JBoss Sever Errors"/>
  <param name="SMTPHost" value="localhost"/>
  <param name="BufferSize" value="10"/>
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="[%d{ABSOLUTE},%c{1}] %m%n"/>
  </layout>
</appender>
```


In Unix / Linux systems there is a centralized log manager called syslog. You can configure Log4J to use this system using the SyslogAppender:

```
<!-- Syslog events -->
<appender name="SYSLOG" class="org.apache.log4j.net.SyslogAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="Facility" value="LOCAL7"/>
  <param name="FacilityPrinting" value="true"/>
  <param name="SyslogHost" value="localhost"/>
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d{ABSOLUTE},%c{1}] %m%n"/>
  </layout>
</appender>
```

- **SyslogHost:** This configuration parameters allows you to specify the local syslog or a remote syslog server used to centralize the network log management.

For more info, visit:

- <http://jboss.org/community/docs/DOC-11280>
- <http://jboss.org/community/docs/DOC-9372>
- <http://jakarta.apache.org/log4j>
- <http://primalcortex.wordpress.com/2007/11/28/jboss-and-jaas-debug>
- [can I get log4j to delete old rotating log files?](#) ^[1]

References

- [1] <http://stackoverflow.com/questions/1050256/how-can-i-get-log4j-to-delete-old-rotating-log-files>How

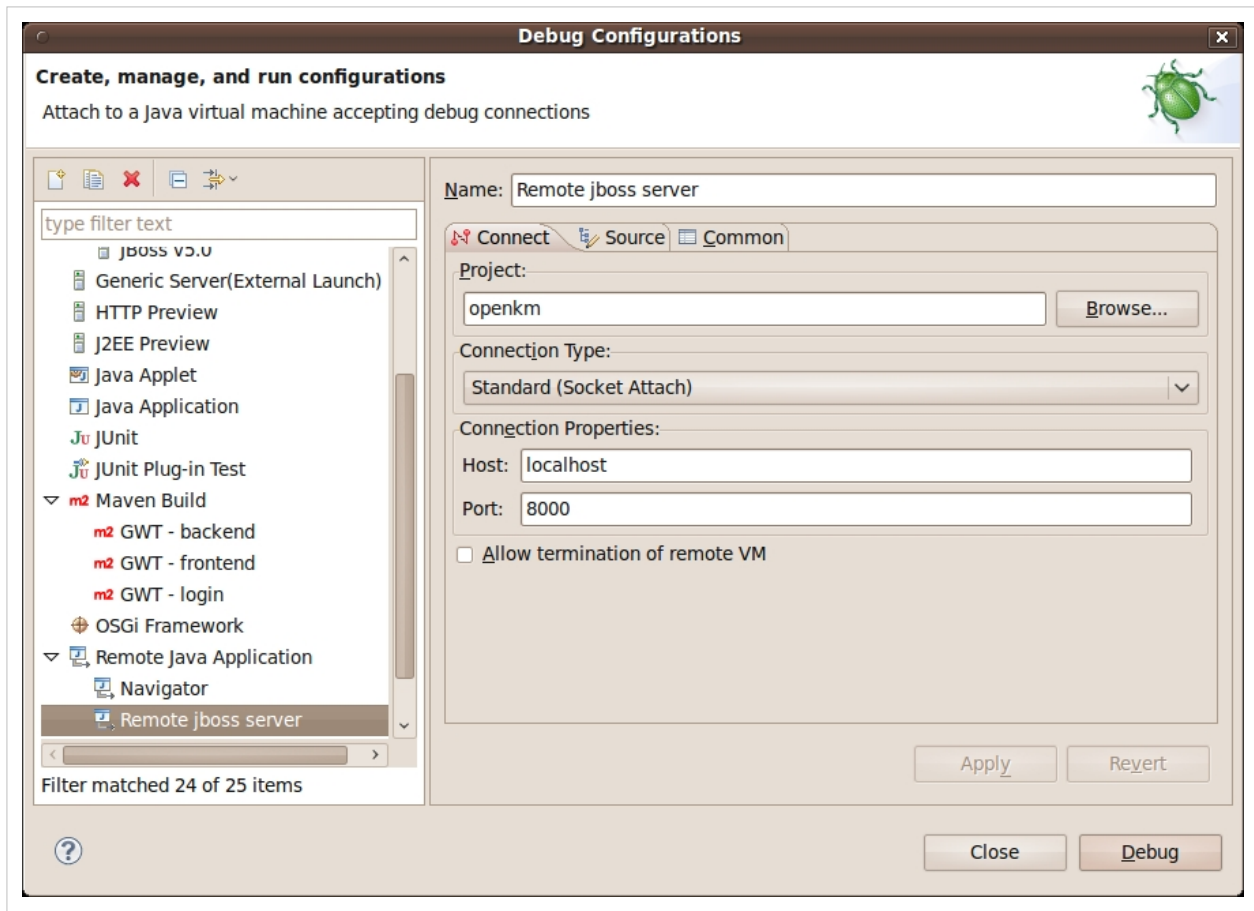
Debugging remote server

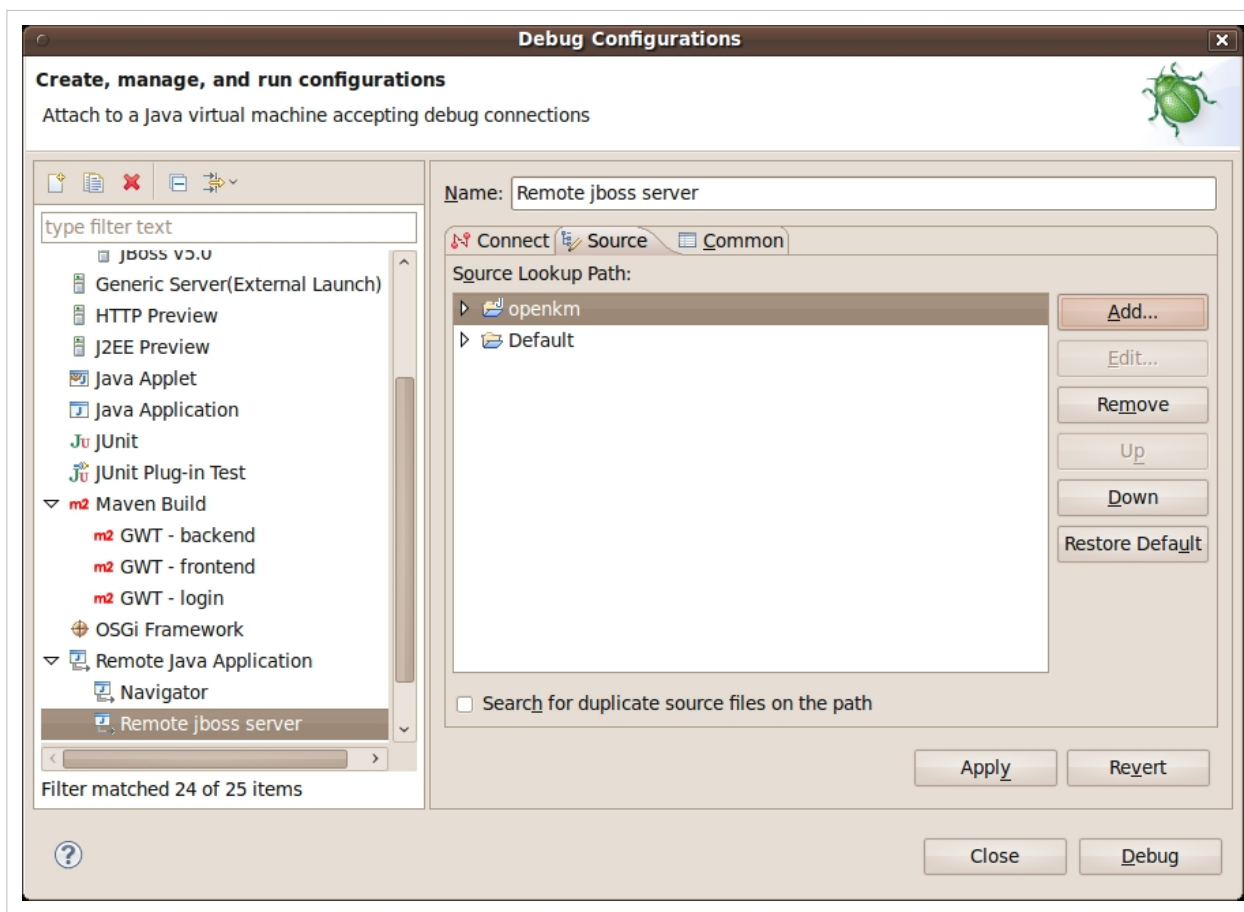
If you got problems on production environment this configuration is specially useful. But it might not be setting if you've not problem in these environments. Add this line at the top of `$JBOSS_HOME/bin/run.sh` script:

```
JAVA_OPTS="-Xmx2000m -Djava.awt.headless=true -Xdebug -Xnoagent \  
-Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=8000"
```

Configure eclipse

Select the project and the host. Include in source your OpenKM java project. You can debug normally, setting breakpoints etc...





Debugging with GWT

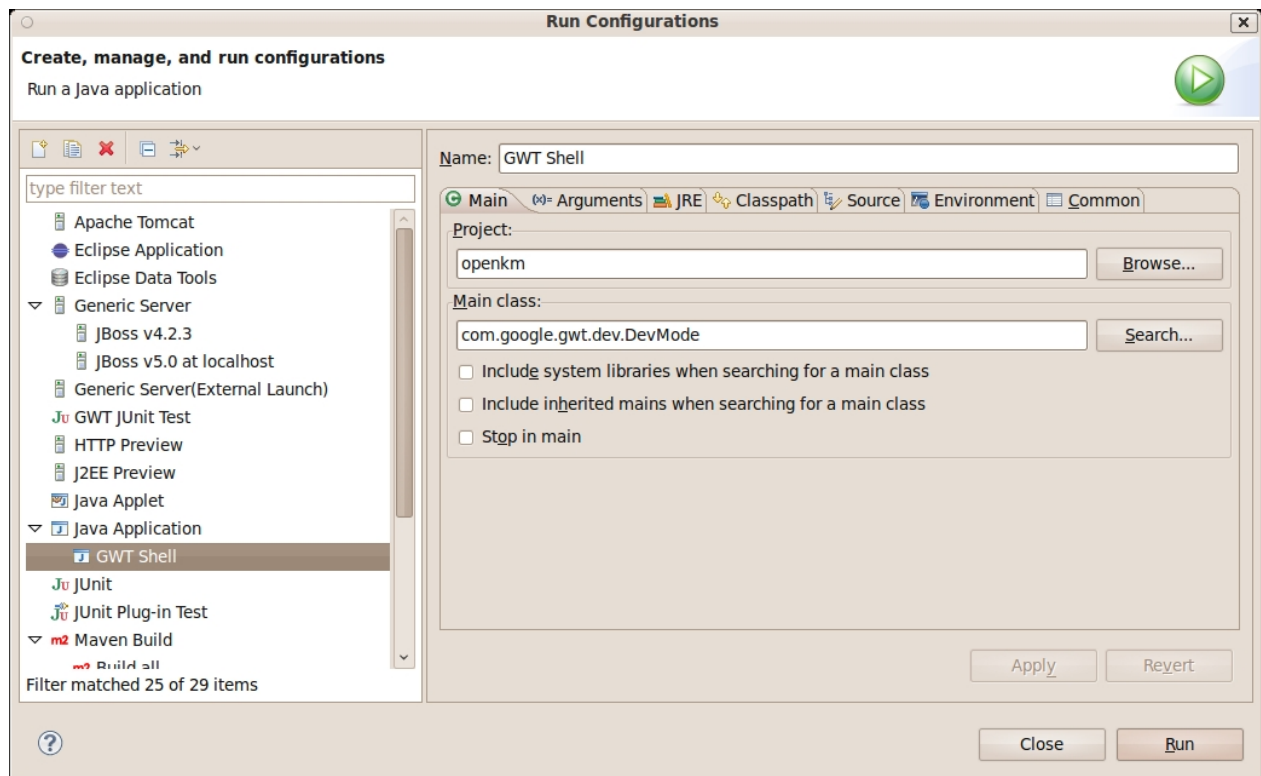
Go to **run configurations** and add new **java application** configuration as you can see in this screenshoot.

First of all, open the pom.xml file and uncomment this section:

```
<!-- Only for development -->
<dependency>
  <groupId>com.google.gwt</groupId>
  <artifactId>gwt-dev</artifactId>
  <version>${gwt.version}</version>
  <scope>provided</scope>
</dependency>
```

Set the Main class:

```
com.google.gwt.dev.DevMode
```

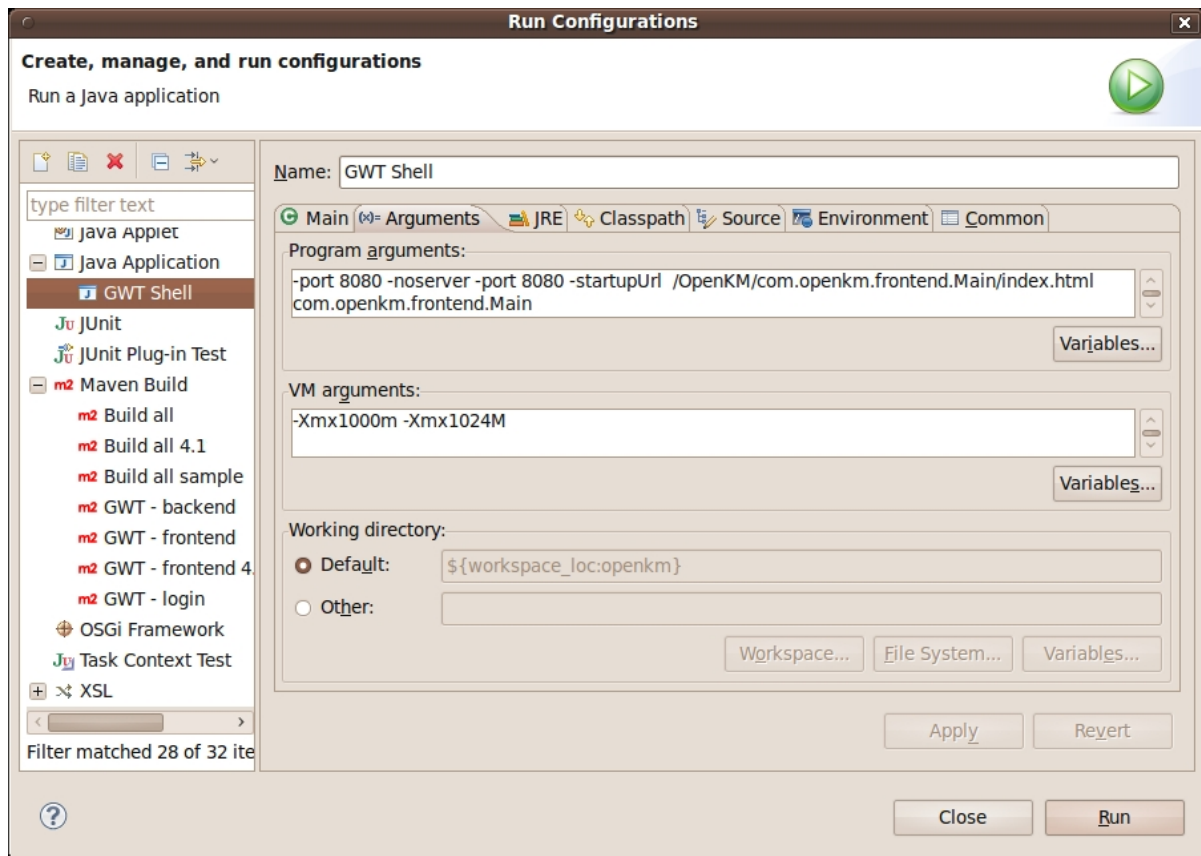


Set the program arguments

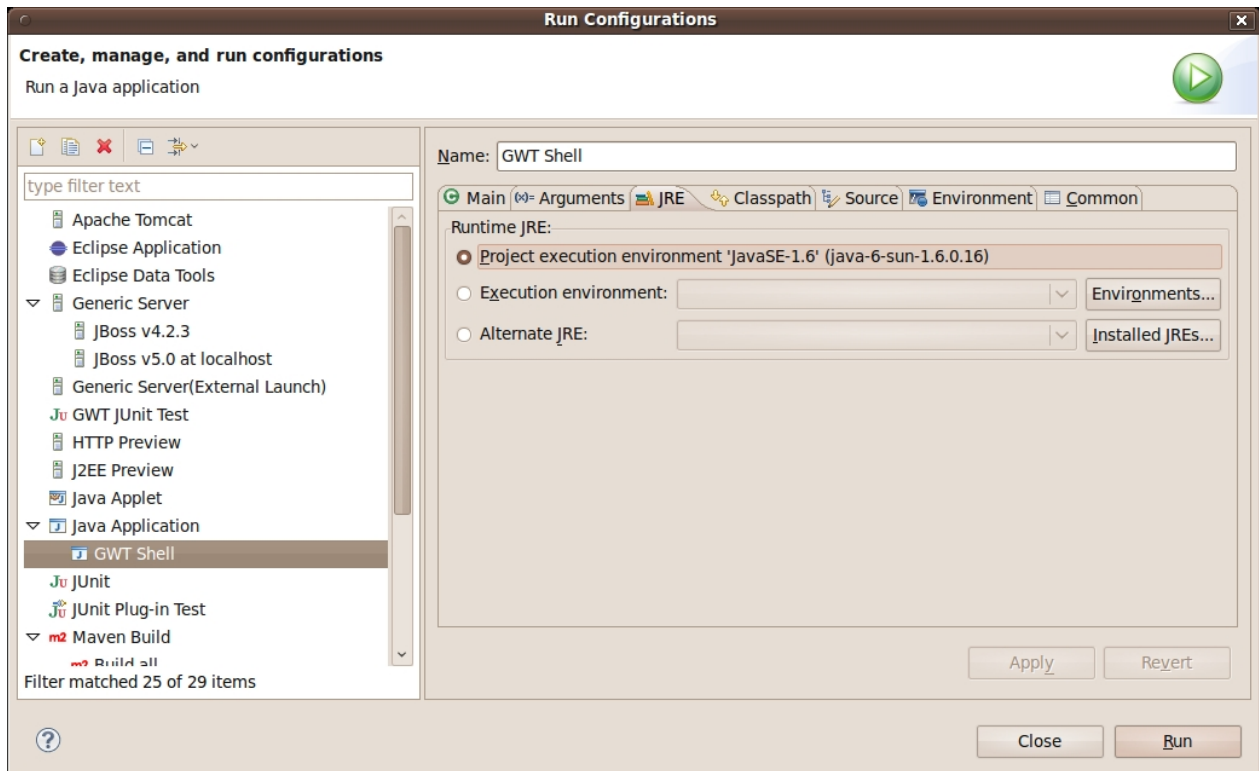
```
-port 8080 -noserver -port 8080 -startupUrl /OpenKM/frontend/index.html com.openkm.frontend.Main
```

and the VM arguments

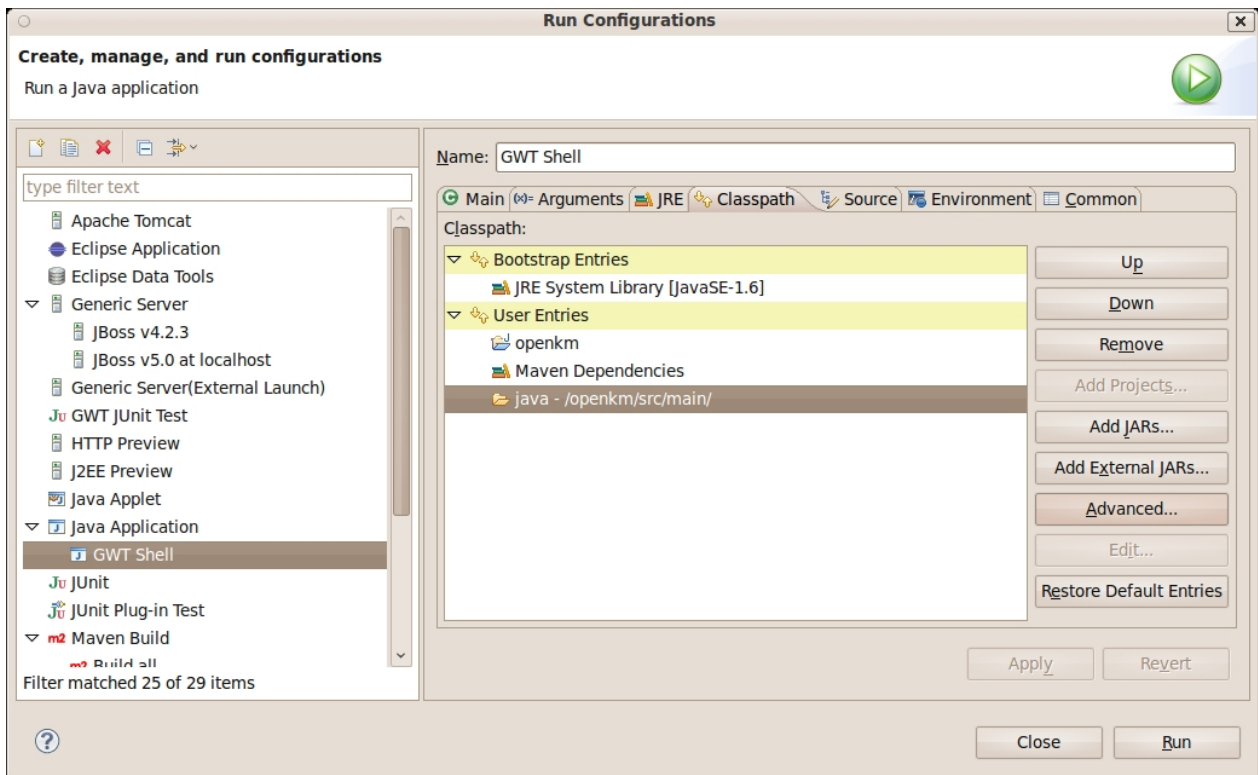
```
-Xms1024m -Xmx1024M
```



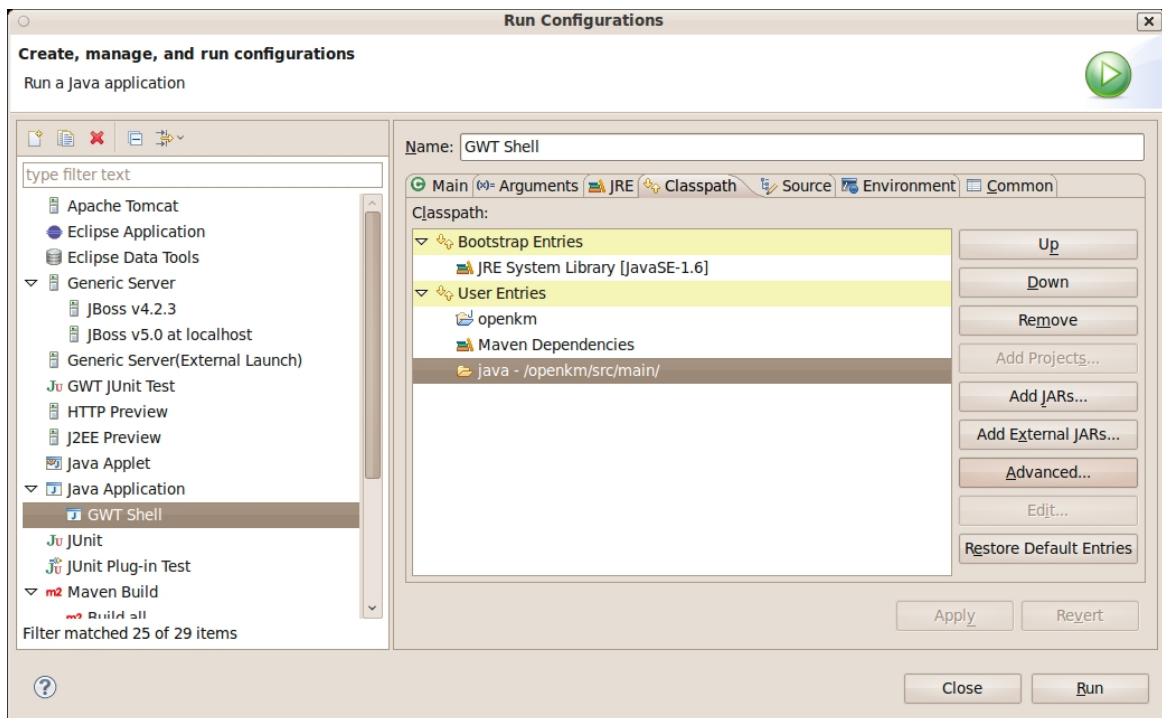
Set the jdk 1.6



Set the class path. Pay special **attention** in **adding** folder `/openkm/src/main` (using button advanced and add folder option)



Now you can run (better if you run in debug mode, you can set breakpoints in your eclipse code). It'll appear some screen like this:

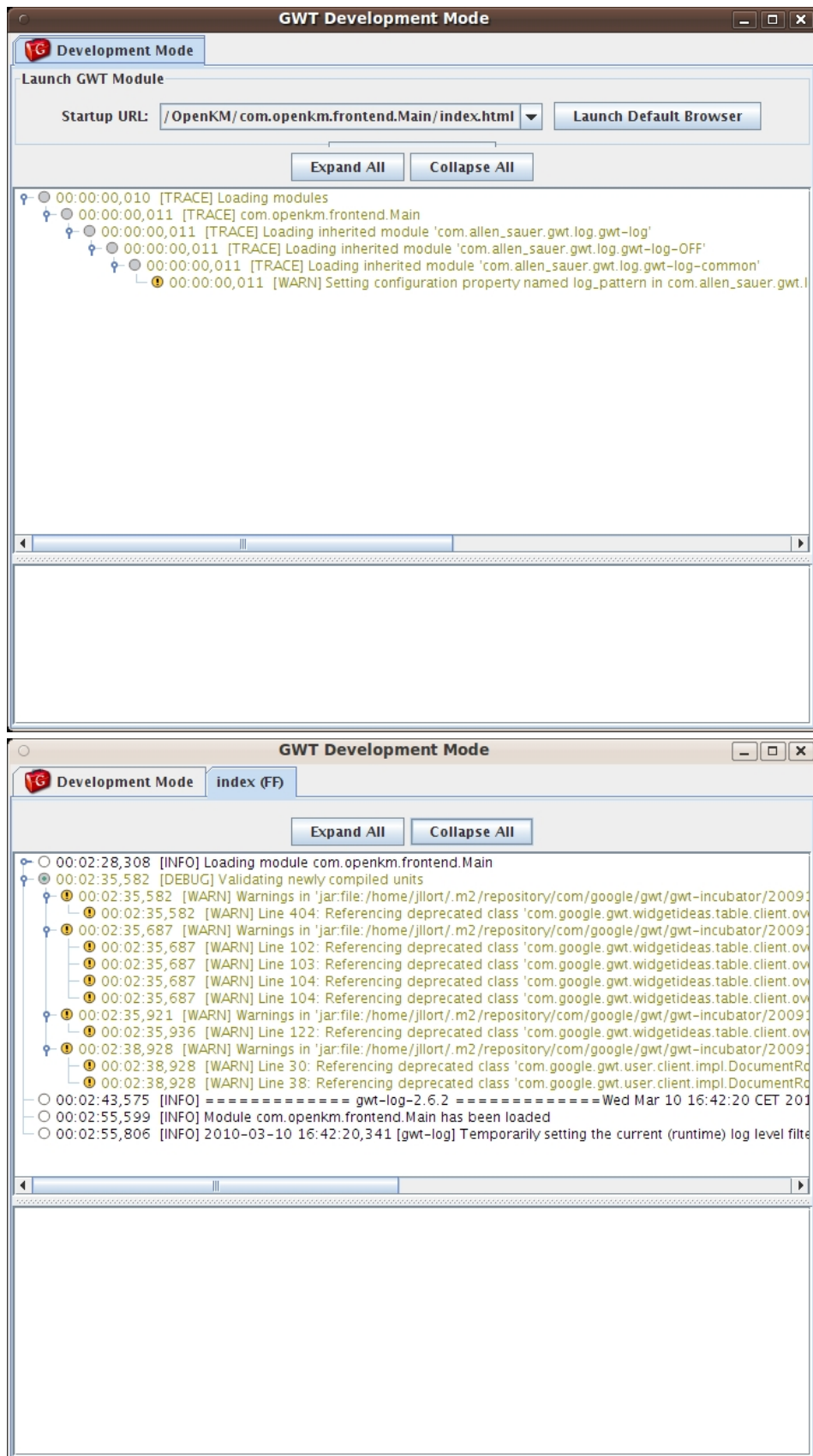


To **debug login** put in your browser

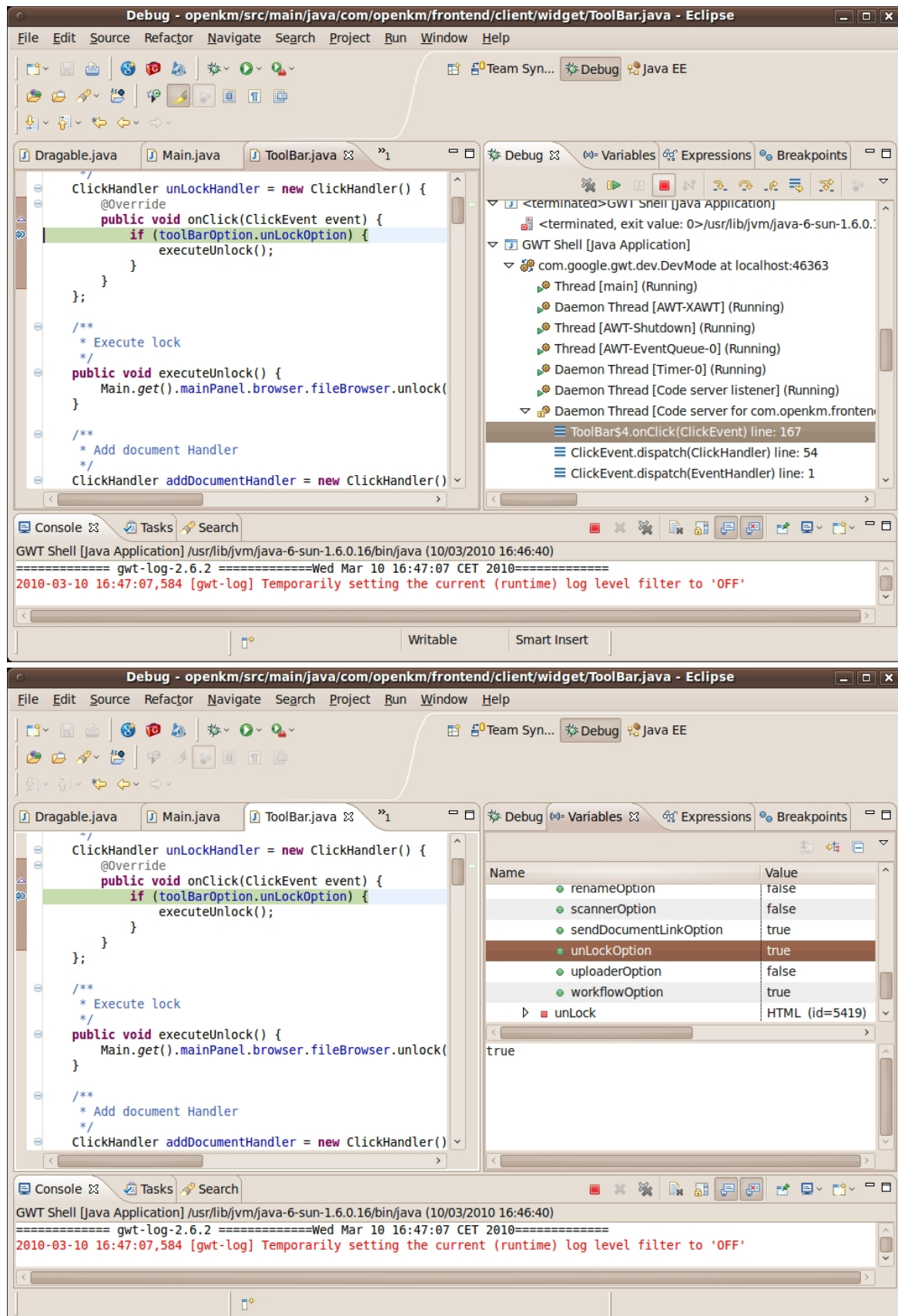
```
http://localhost:8080/OpenKM/com.openkm.login.Main/index.jsp?gwt.codesvr=127.0.1.1:9997
```

To **debug frontend** first authenticate and after it, put in your browser

```
http://localhost:8080/OpenKM/com.openkm.frontend.Main/index.html?gwt.codesvr=127.0.1.1:9997
```



You debug into eclipse, see variables values etc...



OpenKM version 4.1 and older

Set the program arguments

```
-port 8080 -noserver -port 8080 -startupUrl /OpenKM/com.openkm.frontend.Main/index.html com.openkm.frontend.Main
```

Article Sources and Contributors

Developer Guide *Source:* <http://wiki.openkm.com/index.php?oldid=4744> *Contributors:* Jllort, Pavila, Ronny.Fritzsche.ywa

Configure Java in Eclipse *Source:* <http://wiki.openkm.com/index.php?oldid=1027> *Contributors:* Pavila

Configure JBoss server in Eclipse *Source:* <http://wiki.openkm.com/index.php?oldid=928> *Contributors:* Pavila

Maven installation *Source:* <http://wiki.openkm.com/index.php?oldid=2241> *Contributors:* Jllort, Pavila

Checkout source from Subversion *Source:* <http://wiki.openkm.com/index.php?oldid=4534> *Contributors:* Jllort, Liska.aj, Pavila, Ronny.Fritzsche.ywa

Development tips OpenKM 4.0 *Source:* <http://wiki.openkm.com/index.php?oldid=2336> *Contributors:* Jllort

Development tips OpenKM 5.0 *Source:* <http://wiki.openkm.com/index.php?oldid=4952> *Contributors:* Jllort, Pavila

Browsing embeded database *Source:* <http://wiki.openkm.com/index.php?oldid=2443> *Contributors:* Christoph.xmt, Jllort, Pavila

Debugging OpenKM *Source:* <http://wiki.openkm.com/index.php?oldid=3795> *Contributors:* Pavila

Debugging remote server *Source:* <http://wiki.openkm.com/index.php?oldid=1215> *Contributors:* Jllort, Pavila

Debugging with GWT *Source:* <http://wiki.openkm.com/index.php?oldid=4954> *Contributors:* Jllort, Pavila, Wangmj

Image Sources, Licenses and Contributors

[illegible]