

# ETICS Repository Installation

## Installation of ETICS Repository

Here you can find some guidelines how to install and configure a development environment for ETICS repository in Linux, using Eclipse.

### 1. Install Eclipse

Go to [www.eclipse.com](http://www.eclipse.com), get "Eclipse IDE for Java Developers" for your OS and install it.

### 2. Install Subversive

*Subversive* is a *plugin* for *Eclipse* which enables associating projects with a *subversion* repository, making it very easy to checkout, commit and maintain everything synchronised. It can be easily installed from *Eclipse*: Go to Window->Install New Software. Search for subversion in all available sites and install "Subversive SVN Team Provider".

You will also need to install a *connector*. After restarting Eclipse, a window will appear asking you to choose a connector to be installed. You may choose any you prefer.

At the time of writing, full details about installing *subversion* could be found here:  
<http://www.eclipse.org/subversive/documentation/gettingStarted/aboutSubversive/install.php>

### 3. Create a Java Project from SVN repository

Go to File->New->Project. Choose under SVN, "Project from SVN". Create a new repository location, using the location listed above, and finish. In the New Project Wizard that appears, choose "Java Project", choose a project name and finish. You should have now the project, checked out from the repository, in your workspace.

In the next sections, consider {project\_dir} to be the base directory where your project is in your file system.

### 4. Libraries

As you probably notice, there are lots of errors in the project due to missing libraries.

In the next table there are the names of the used jar files, which can help you find out which libraries are used and their version number. You can find which libraries are being used by checking the repository configuration in ETICS. Go to <https://etics.cern.ch/eticsPortal> -> configuration tab. Add org.etics.repository.webservice to the workspace and check its dependencies.

You can also find all the libraries in the externals: Go to <https://etics.cern.ch/eticsPortal> -> repository tab -> registered -> externals.

activation.jar	jackrabbit-spi-commons-1.4.jar
ant.jar	jackrabbit-text-extractors-1.4.jar
axis-ant.jar	jaxb-api.jar
axis.jar	jaxb-impl.jar
bcp4j-1.42.jar	jaxrpc.jar
bcprov-1.37.jar	jcr-1.0.jar
commons-collections-3.2.jar	jsr173_1.0_api.jar
commons-discovery-0.2.jar	log4j-1.2.14.jar
commons-logging-1.0.4.jar	lucene-core-2.2.0.jar
concurrent-1.3.4.jar	mail.jar
derby.jar	saaj.jar
jackrabbit-api-1.4.jar	servlet-api.jar

jackrabbit-core-1.4.9.jar	slf4j-api-1.5.0.jar
jackrabbit-jcr-commons-1.4.2.jar	slf4j-log4j12-1.5.0.jar
jackrabbit-spi-1.4.jar	wsdl4j-1.5.1.jar

Copy all the libraries to some place, for example, {project\_dir}/lib. Then add these libraries to the build path. To do this, in Eclipse go to Project->Properties, click in "Add JARs" and select all the libraries. *Eclipse* should now know where all the needed libraries are.

## 5. Configuration

Under the directory {project\_root}/config/, create local copies of *eticsRepositoryWSConfiguration.xml*, *eticsRepositoryWSLog4j.properties* and *eticsRepositoryWSRepository.xml* (for example *eticsRepositoryWSConfiguration.local.xml*, ...) and add them to svn ignore. In this way, you will be able to test locally the repository with your own configuration files without messing the files in the repository. Here is a brief description of the required changes:

### *eticsRepositoryWSConfiguration.xml*

repositoryDescriptor	Location of local version of <i>eticsRepositoryWSRepository.xml</i>
repositoryHome	Location where repository data is stored. For example, you can create the directory {project-dir}/test-jr-home.
repositoryCND	{project-dir}/config/eticsRepositoryWSNodeTypeDefinition.cnd
FSRepositoryRoot	Location to simulate AFS directory, where registered data is stored. For example, {project-dir}/test-afs-copy/registered.
FSVolatileRoot	Location to simulate AFS directory, where volatile data is stored. For example, {project-dir}/test-afs-copy/volatile.
keyStore	"/etc/grid-security/hostKeystore.pkcs12"
keyStorePassword	"" (empty)
trustStore	/etc/grid-security/caKeystore.jks
trustStorePassword	"" (empty)
buildSystemWebservice	"https://etics.cern.ch:8443/BuildSystemService/services/BuildSystemService"
log4jPropertyFile	Location of local version of <i>eticsRepositoryWSLog4j.properties</i>

### *eticsRepositoryWSRepository.xml*

DataStore "org.etics.repository.webservice.data.AFSDataStore" path	{project-dir}/test-jr-home/datastore

## 6. Tomcat

You need *Tomcat* to deploy the Java application. You also need *Tomcat admin* to have a web interface that helps manage *Tomcat*. It also allows easily deploy applications without restarting *Tomcat* (See section Deployment).

For example:

```
sudo apt-get install tomcat6 tomcat6-admin
```

Check <http://localhost:8080> to see if installation of *Tomcat* was successful.

To configure the access to the web manager, go to /etc/tomcat6/tomcat-users.xml and add the following:

```
<!--role rolename="manager" -->
<!--user username="user" password="pass" roles="manager" -->
```

Choose an appropriate username and password. You can then access the manager through the following url:

<http://localhost:8080/manager/html>

You need to pass a context parameter to indicate to the repository where the configuration file is. The easiest way is to edit `{CATALINA_HOME}/conf/web.xml` and add, inside `<web-app>` tag, the following:

```
&lt;context-param&gt;
&lt;param-name&gt;configurationFile&lt;/param-name&gt;
&lt;param-value&gt;\\{project_root}\\config\\eticsRepositoryWSConfiguration.local.xml&lt;/param-value&gt;
&lt;/context-param&gt;
```

## 7. Build & Deploy

To build and deploy in a easy way, you will need to have installed Ant and curl. If you don't have them:

```
sudo apt-get install ant curl
```

Check the script in TODO.

This script builds the repository by running Ant tool with the `build.xml` in the base directory. You need to edit the command and update the location of the libraries. Next, it deploys the repository's war file using the manager web interface. This method doesn't require restarting *Tomcat*. Finally it tails *Tomcat* log.

After deploying, you should have the repository web interface in the following address:<http://localhost:8080/repository/>

## 8. Debugging

You can debug the repository server. For this, you need to pass some additional flags to the JVM when running Tomcat. In `"/etc/default/tomcat6"`, uncomment the following line:

```
JAVA_OPTS="{JAVA_OPTS} -Xdebug -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n"
```

This will enable remote debugging, and you will be able to connect a Java debugger to port 8000. After restarting Tomcat, you can check if it is running:

```
sudo /etc/init.d/tomcat6 restart
netstat -lt --numeric-ports | grep 8000
```

Now, in *Eclipse*, go to Run->Debug Configurations. Create a new *"Remote java Application"*, setting *"Connection Type"* to *"Standard"* and *"connection"* to `localhost:8000`.

You should now be able to connect to the JVM, using the debug functionality (F11), create breakpoints, etc..`sudo apt-get install ant curl`