Building org.lcsim software

DEPRECATED

Most of the information on this page is now out of date, as it refers to the old Maven 1 build system. The lcsim projects have been ported to use Maven 2 instead.

Prerequisites

To build the org.lcsim software, you should have the following installed.

- Java 1.5 or 1.6
- Maven 2.0.9
- Netbeans 6.5
- CVS client such as command-line on Linux, Tortoise CVS, or just use the Netbeans cvs client

If you are following the lcsim tutorial instructions, these tools should be setup by now.

Building org.lcsim software

The lcsim software consists of three primary packages: GeomConverter, org.lcsim, and lcsim-contrib.

Most users will be primarily interested in building and running lcsim and/or lcsim-contrib.

Here are condensed instructions, which you can execute from any bash-like unix shell.

```
export CVSROOT=:pserver:anonymous@cvs.freehep.org:/cvs/lcd
cvs login (just hit enter when prompted for password)
cvs co lcsim
cd lcsim
mvn
```

The build procedure for lcsim-contrib is virtually the same. Just replace "lcsim" in the above instructions by "lcsim-contrib".

Keeping the Release Current

Oftentimes new directories are added in a new release. In order
to make sure that cvs adds those, please use the command

```
cvs update -PdA
```

which will

P: prune empty directories
d: create directories
A: reset sticky tags

```
man cvs
```

will give more information on the cvs program.

Installing Custom JARs into JAS3

By default, your lcsim and lcsim-contrib builds install their current jar files into **~/.JAS3/extensions**.

Edit the pom.xml file to turn off this behavior.

Testing the installation

Test Directory

By default, maven will also run the JUnit tests associated with this distribution.
Test reports are written to

```
target/test-reports/
```

If you wish to build without testing, issue the following command:

```
maven -Dmaven.test.skip=true
```

To execute a single test case, e.g. for org.lcsim.recon.cluster.fixedcone.FixedConeClusterer:

```
maven test:single -Dtestcase=org.lcsim.recon.cluster.fixedcone.FixedConeClustererTest
```

Additional Component Tests

There are also other component tests for many of the org.lcsim classes. Until they can be ported to the JUnit structure they need to be manually executed. Do so with the following ant target:

```
ant -f ctest.xml
```

Running Classes from the Command Line

We have defined a Maven goal that allows users to run a Java class in the project from the command line using the freehep-run-plugin.

For example:

```
 maven run -Drun.class="org.lcsim.geometry.compact.converter.heprep.Main" -Dargs="test/org/lcsim/geometry
/compact/sdjan03_compact.xml tst.heprep"
```

The above command run the **Main** class to convert the given compact description to a HepRep.

Compiler Warnings

By default, compiler warnings about deprecation and unchecked method or class usages are turned off.

To turn on these warnings, remove the "#" from these two lines in *lcsim/project.properties*

```
#maven.compile.deprecation=on
#maven.compile.compilerargs=-Xlint:unchecked
```

Now you should see lots of warning messages, which will effect both Netbeans and command-line builds.

To customize Netbeans, only, you can change how the Netbeans Build Mapping works.

Right-click on the org.lcsim project and select "Properties". Now double-click on "Goal to IDE Mappings". Where it says "Build", you should see "jar:install".

To see all warnings, change the Build line to this:

```
jar:install -Dmaven.compile.deprecation=on -Dmaven.compile.compilerargs=-Xlint:unchecked
```

This should have the same effect as changing the *project.properties* file, but command-line builds using maven only will unaffected.

Building the Documentation

Generating the documentation with maven is quite easy. Just execute this command.

```
maven site
```

This builds the site at the location *./target/docs*. You can use this site for local documentation by browsing to *./target/docs/index.html*.

The reference copy of the lcsim documentation is kept at http://www.lcsim.org/software/lcsim. This usually corresponds to the CVS head version.

Increasing the Java Heap and Perm Size

The LCSim tests can take up more memory than the default heap/perm size assigned to the java VM.

Use this environment variable setting to allocate more memory, i.e. 1 gigabyte heap / 256 megabyte perm, to maven.

```
MAVEN_OPTS="-Xmx1024m -XX:MaxPermSize=256m"
```

If you are building the lcsim documentation, including running all the tests, you are required to increase the heap size beyond the initial default of 256 megabytes. Otherwise, the target will fail due to an "out of memory" error.