

Installing Netbeans with Maven Support

Installing Maven with Netbeans Support

This tutorial will show you how to install Netbeans including support for the Maven build tool.



The tutorial assumes that you have [Java JDK \(version 1.5.x\)](#) (or higher version number) and [Maven](#) already installed.

Install Netbeans

Netbeans 5.0 can be downloaded from <http://www.netbeans.info/downloads/download.php?type=5.0>. Windows users should just click on the netbeans-5_0-windows.exe after download to install netbeans.

Netbeans 5.5 can be downloaded from: <http://www.netbeans.info/downloads/index.php>

Install Mevenide

Download (do a "save as" on the link) this small [autoupdate module](#).



Warning

If you use IE on windows it will save the file with extension .zip. You must change this back to .nbm before proceeding

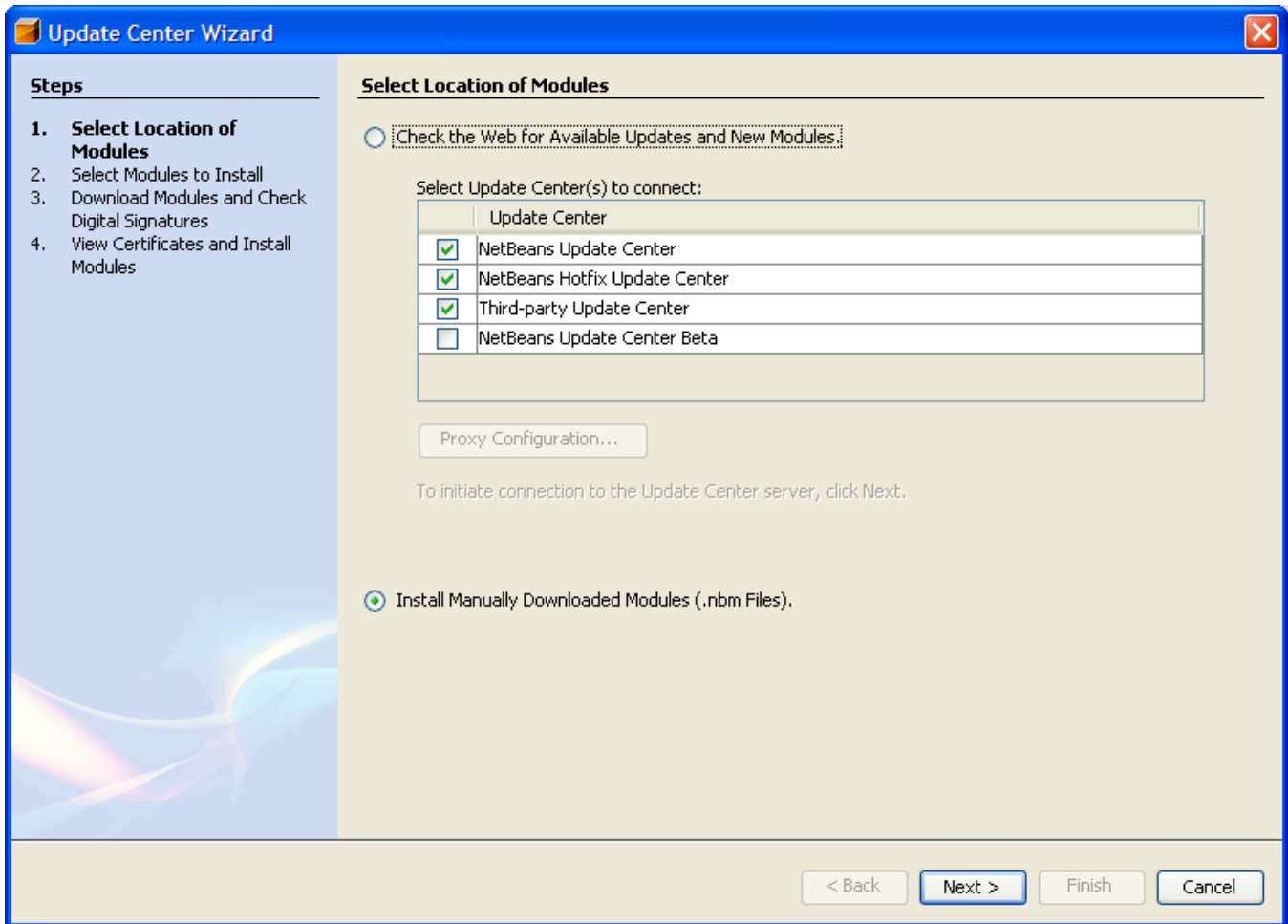


Warning

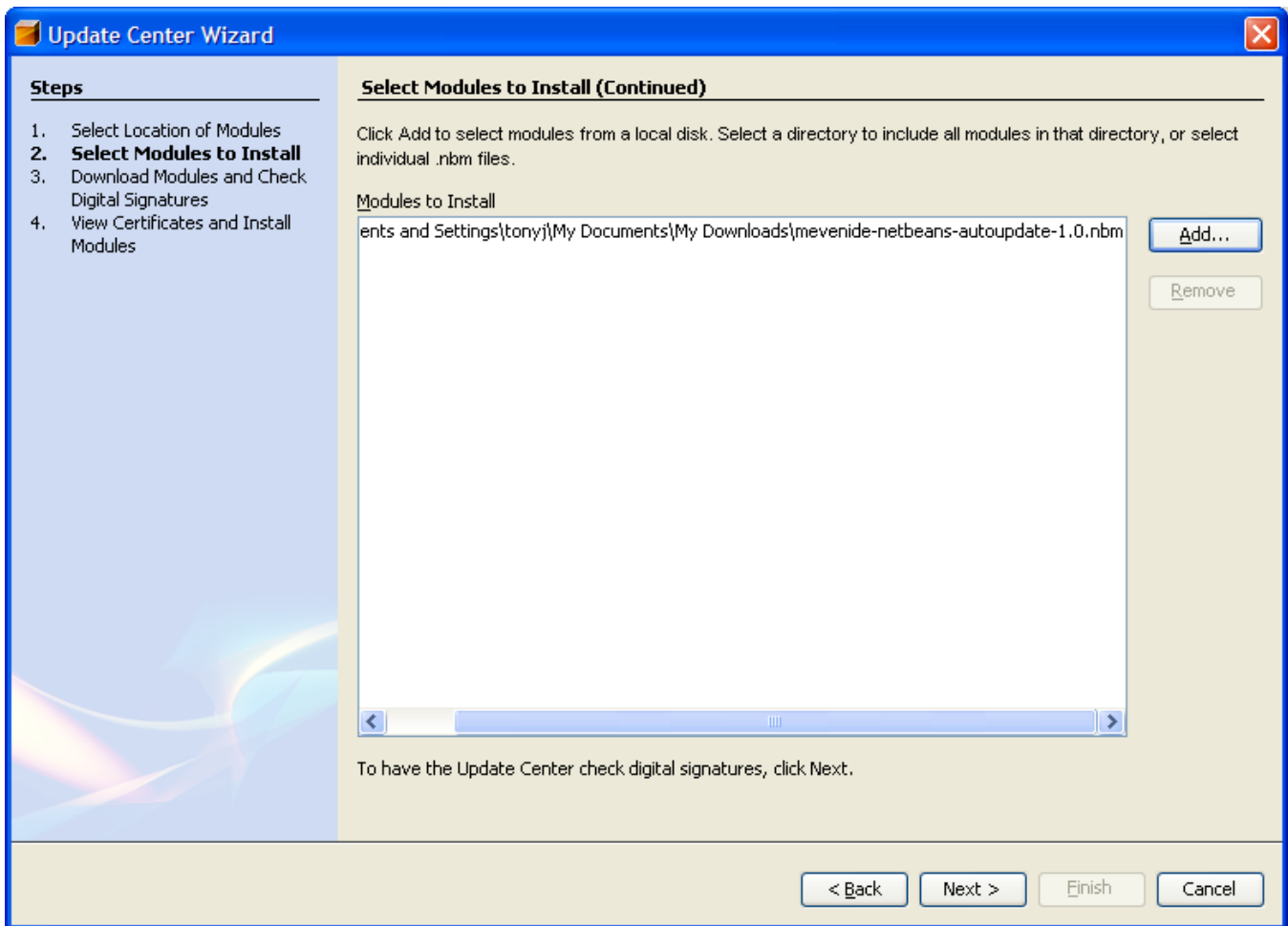
We use maven 1.0.2 and consequently we use mevenide 1.0, not mevenide 2.x. (You can install both mevenide 1.0 and 2.x simultaneously if you are using maven 2 for some other projects.)

To install, start NetBeans and select the **Tools -> Update Center**.

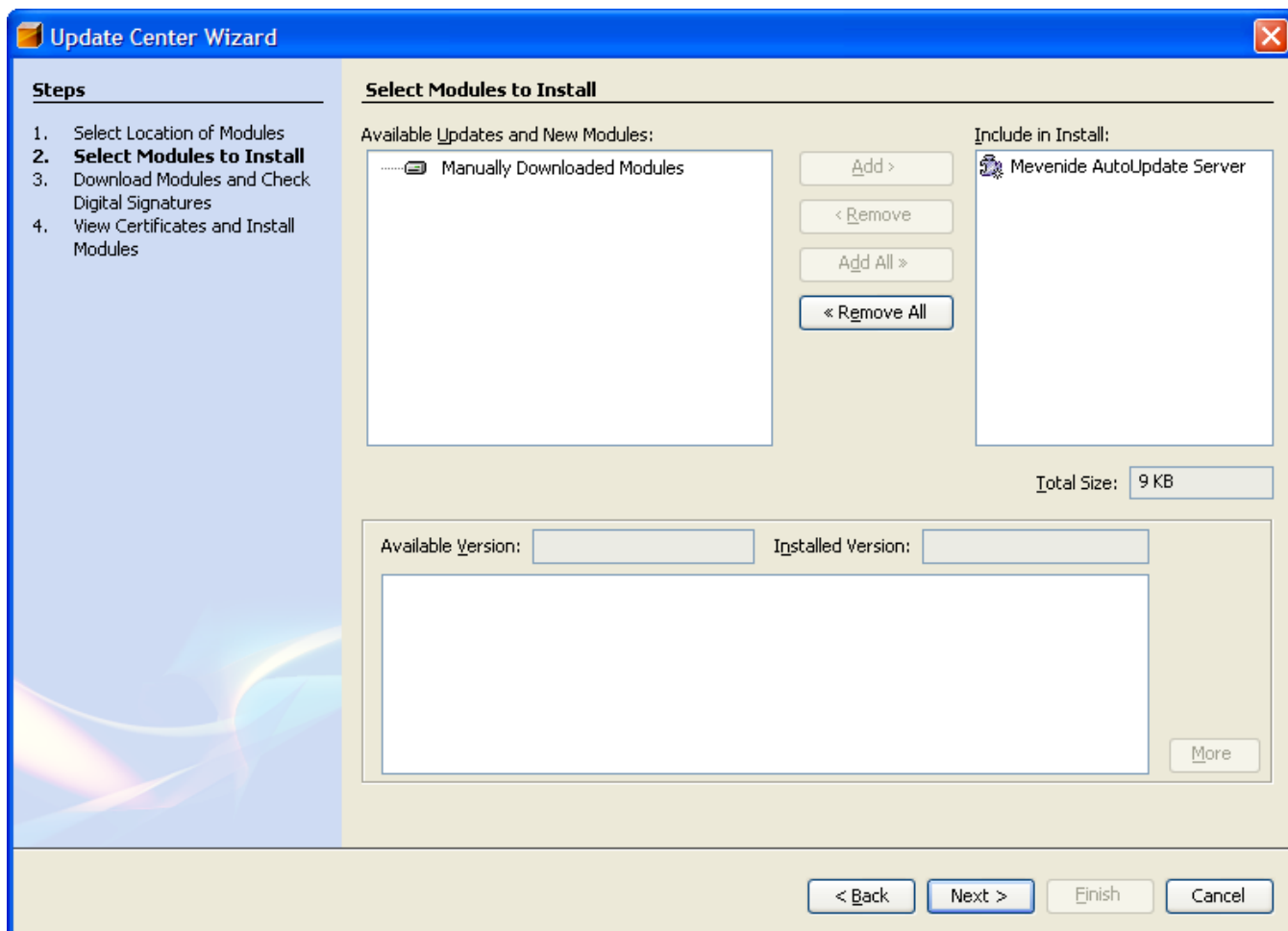
Select the "Install Manually Downloaded Modules (.nbm Files)."



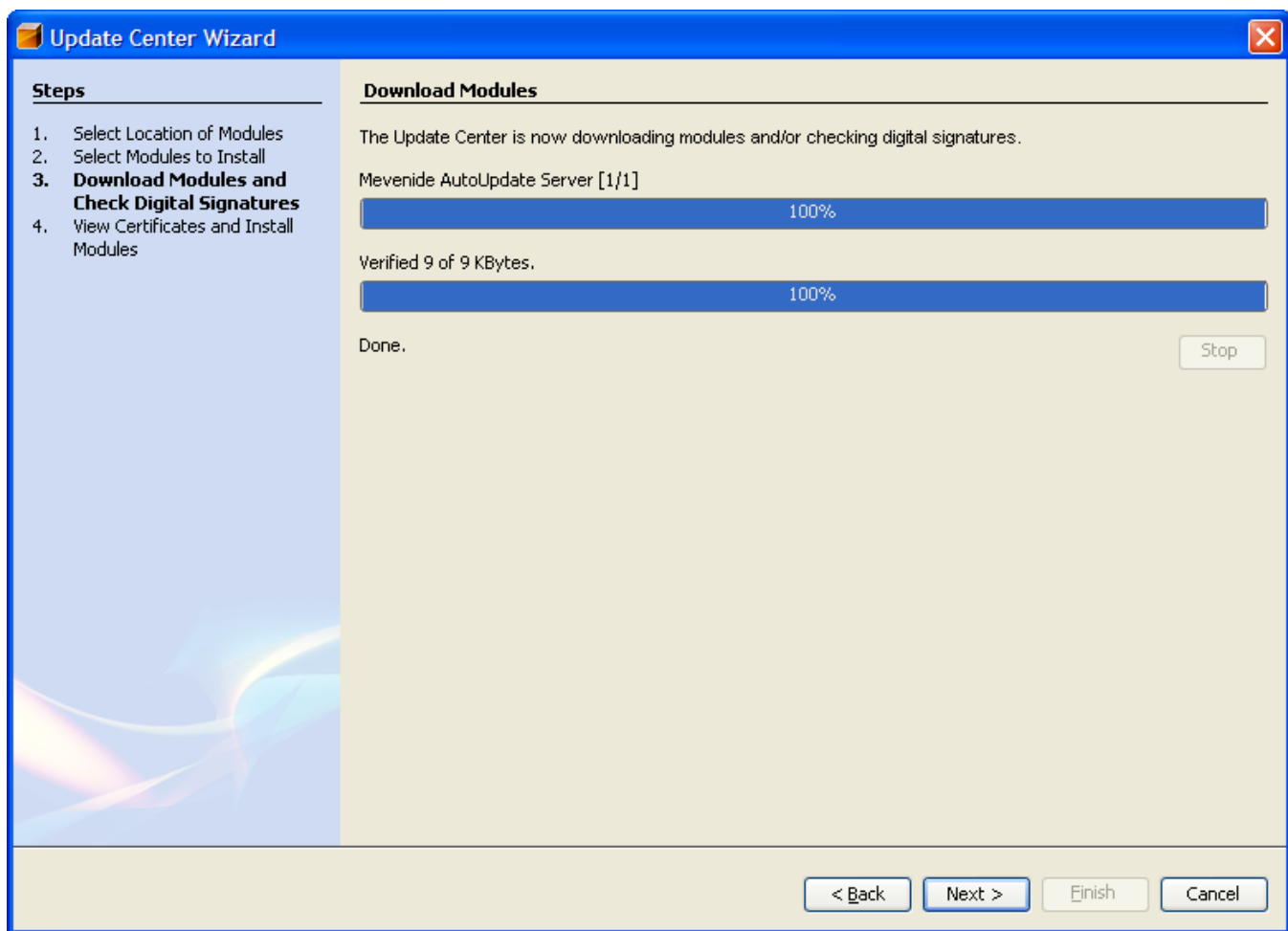
On the next page, select "Add..." from the "Modules to Install" panel and navigate to the location of the downloaded file.



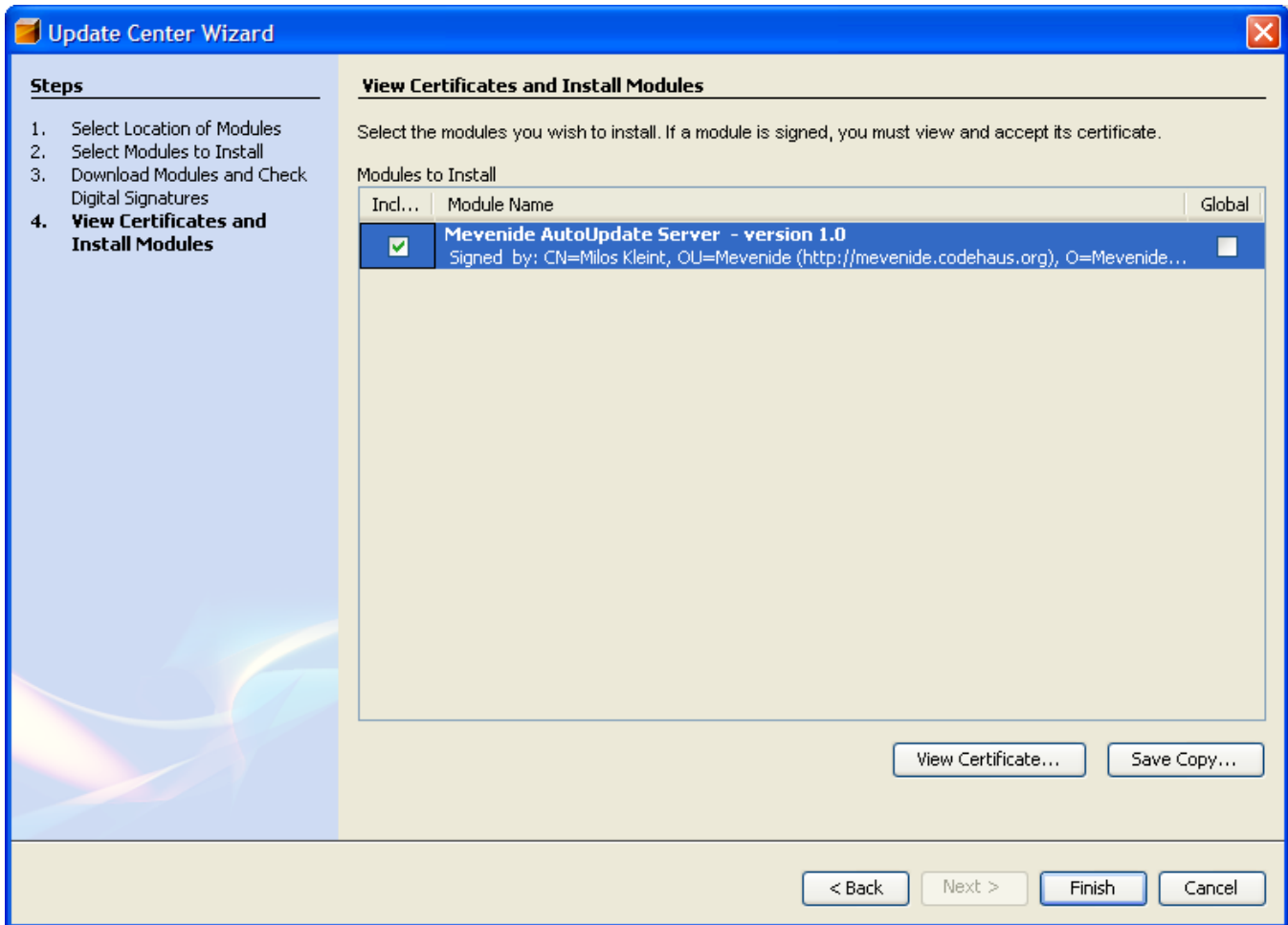
On the next page, transfer that file from the "Available Updates and New Modules:" panel to the "Include in install:" panel and click on the "Finish" button.



On the next panel the module will be downloaded.

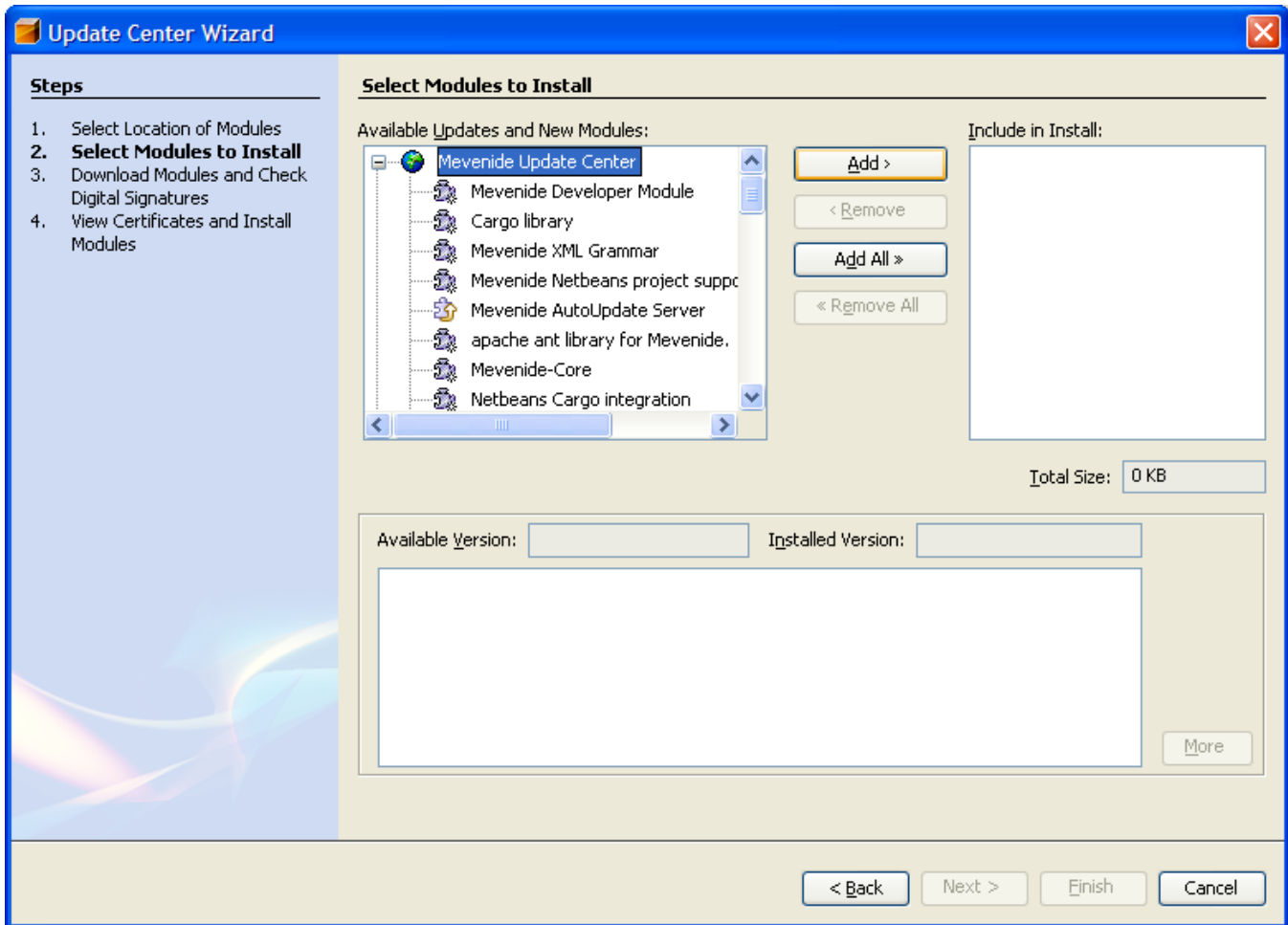


On the next panel select the checkbox next to the downloaded module, and accept any licenses that appear.

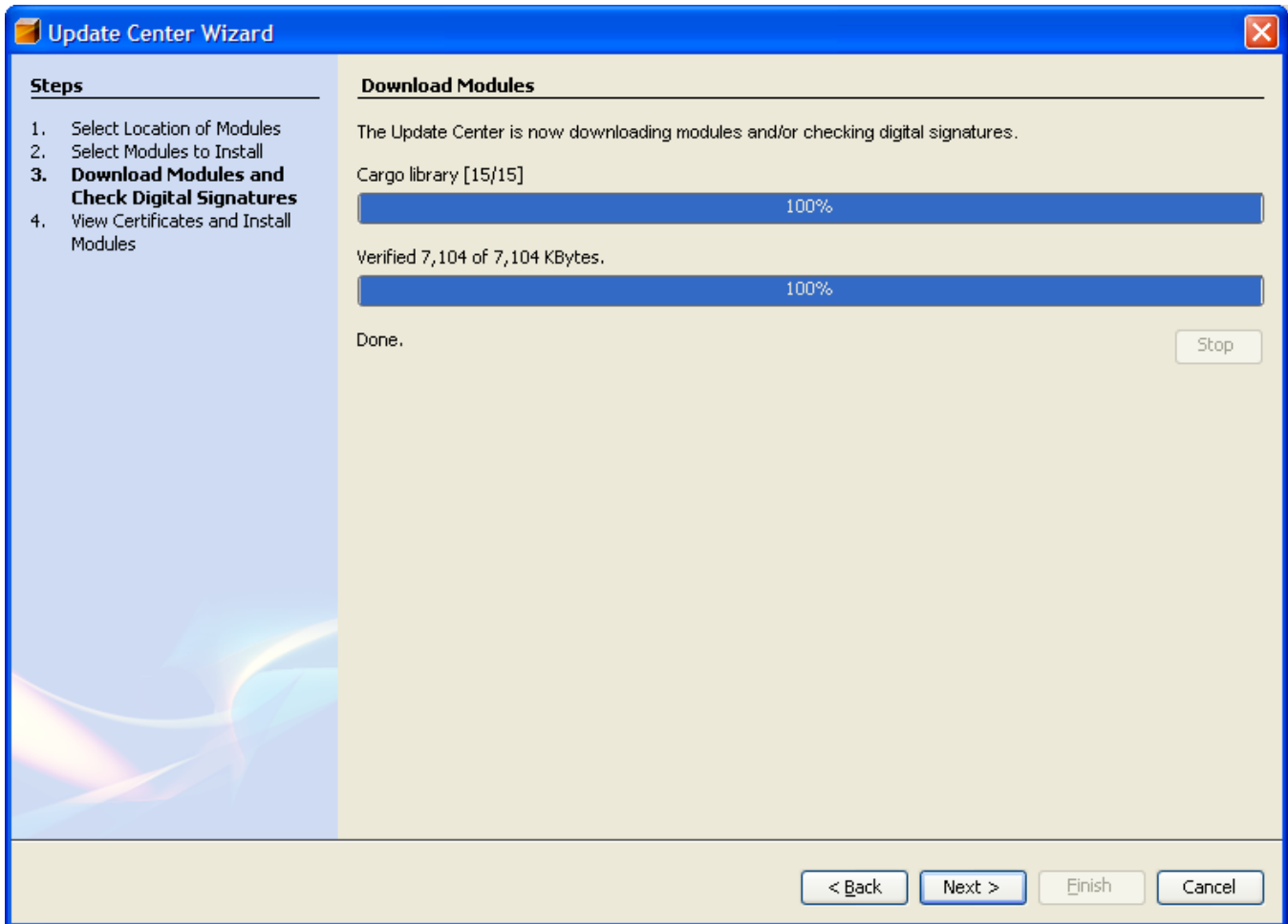


Finally click finish to dismiss the dialog.

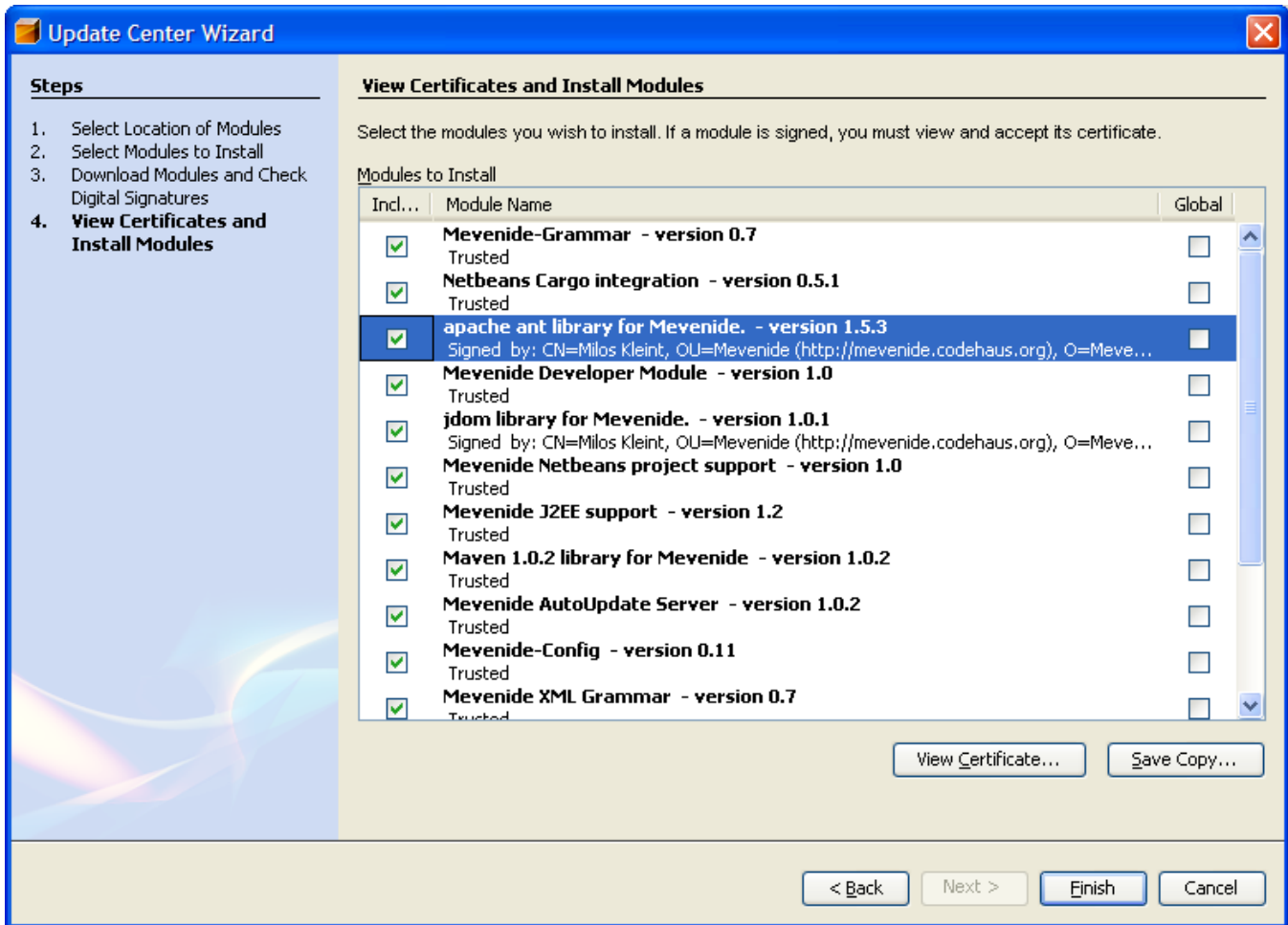
Now go back to **Tools -> Update Center**. Select "Check the web ..." and the "Mevenide Update Center" box, only. Click next. (Note: NetBeans must have access to the Internet for this step to work; make sure proxies and firewalls are configured correctly). Now select the "Mevenide Update Center" node, and click the "Add>" button to move all maven plugins into the righthand box.



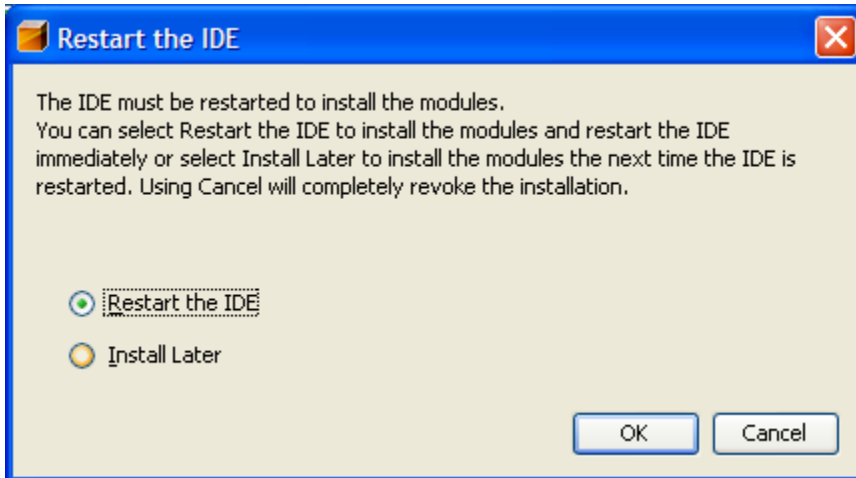
Wait while the modules download, the click "Next>" to continue.



Click the "Include" checkbox for each plugin and then the "Always Accept" button. All of the Maven plugins should have the "Include" box selected, now. Click next.



Now restart Netbeans.

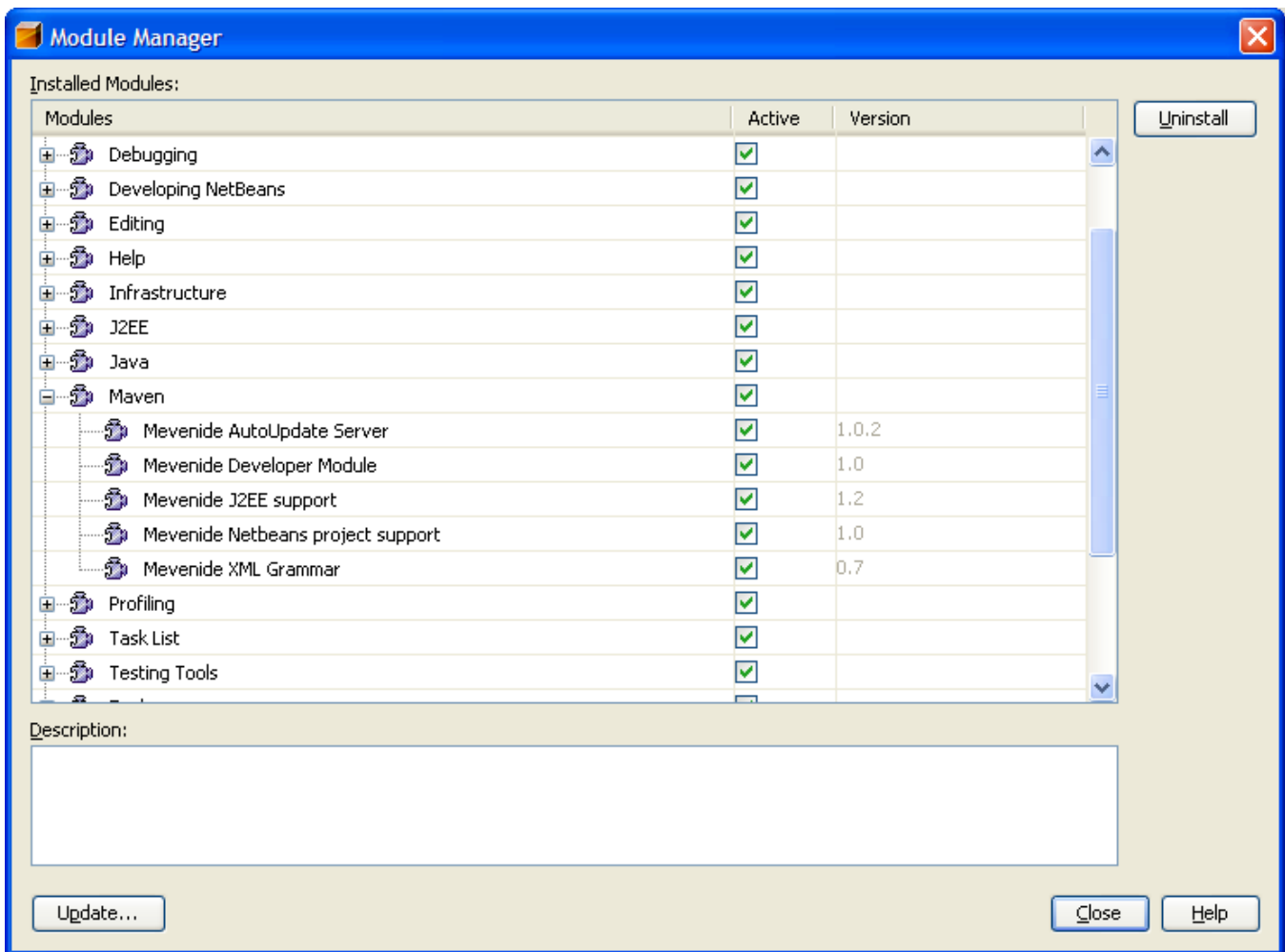


Test the Installation

Now check whether all plugins were loaded successfully.

Go to **Tools -> Module Manager -> Maven**.

Maven Plugins



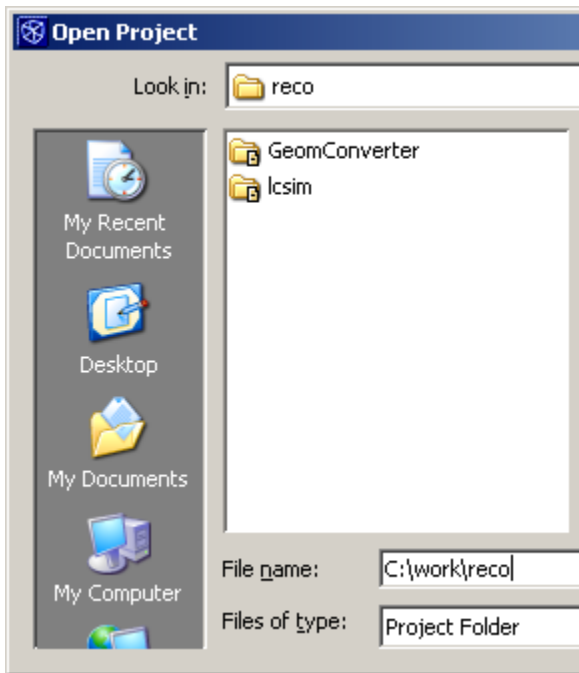
All the Maven plugins should have the **Enabled** box checked.

Set these all to **Enabled** if they are not already. This may clear up errors that can occur during the installation.

Open Project

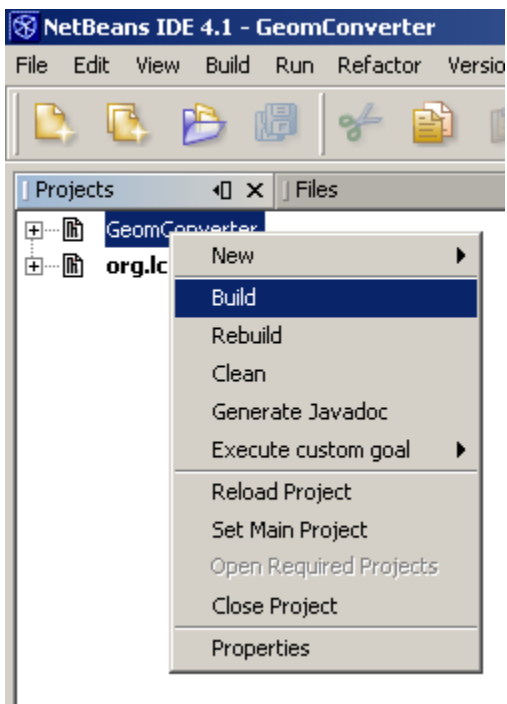
To open the cvs projects which you previously checked out, use **File -> Open Project**, then navigate to your cvs installation directory.

If all went well, you should see a small box next to Maven projects that Netbeans can open.



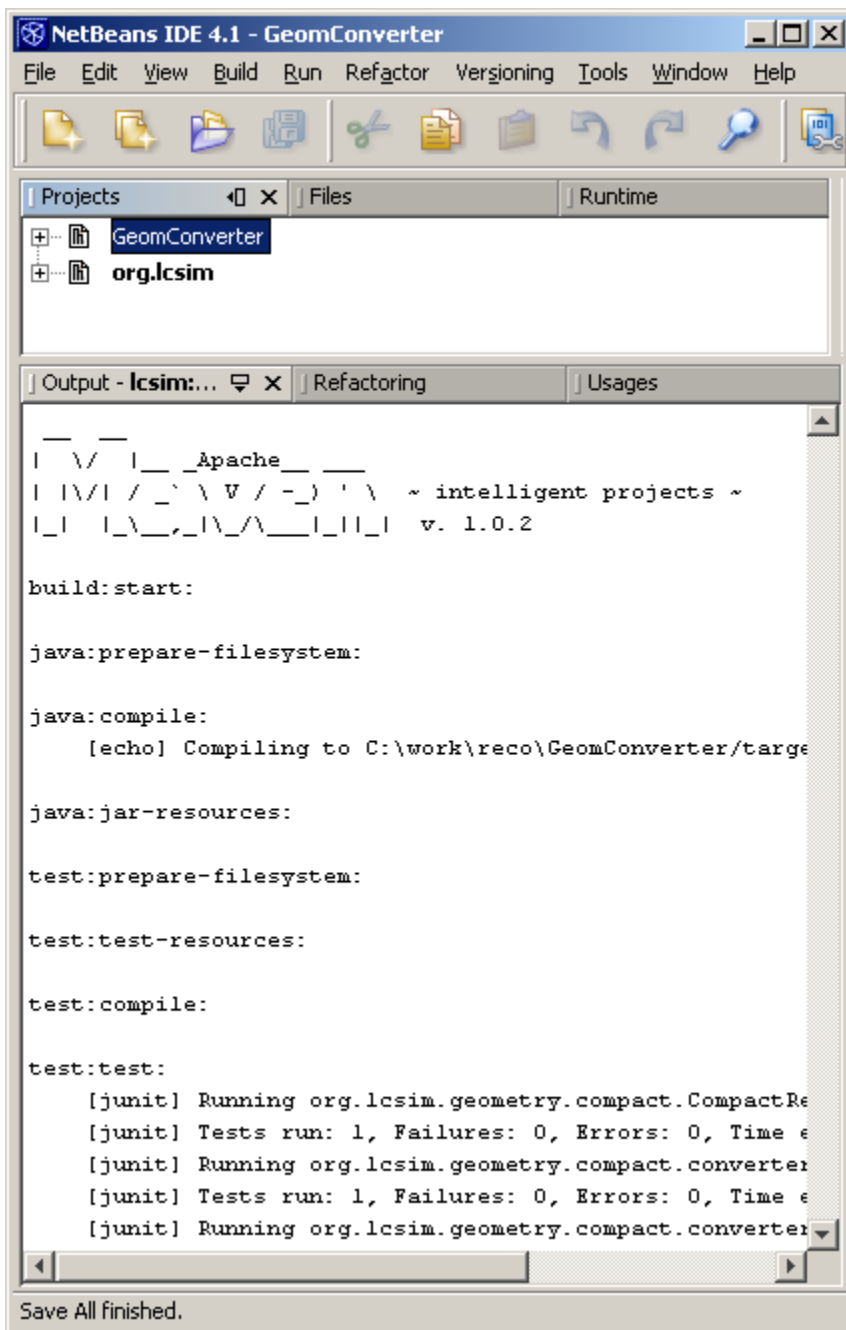
To open a project, click on it and select "Open Project Folder".

The project itself will have the following options, which will now call Maven commands.



To build the project, right-click on build in this menu.

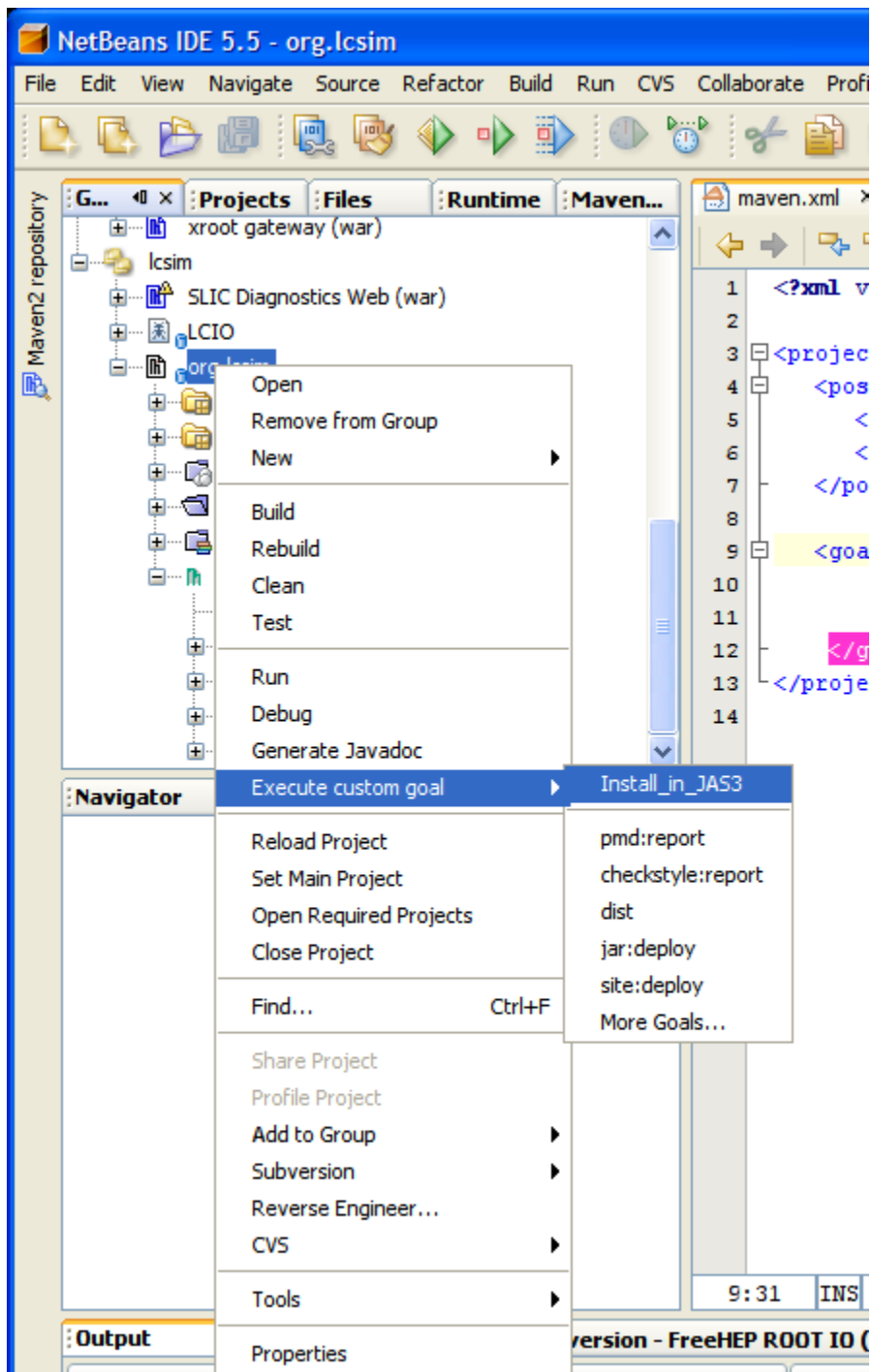
The console should show the Maven splash screen.



Now you have successfully installed Maven support into Netbeans.

Installing updated code into JAS3

Building the code in netbeans will not automatically cause JAS3 to pick up the latest version. To install the latest version into JAS3 right-click on the org.lcsim project, choose "Execute Custom Goal" then choose "Install_in_JAS3".



Note, running this goal on org.lcsim will build org.lcsim, and copy the jar files for org.lcsim, GeomConverter, and other required libraries into your local JAS3 extensions directory. You will need to restart JAS3 for it to pick up these changes.

Turning off tests

If you want to build lcsim.org without running the tests, you can do so by selecting the project in the netbeans project tree, right clicking to get the popup menu and then select "Properties". Then choose "Goal to IDE Action Mappings" and add -Dmaven.test.skip=true to the build goal. This is how I have my settings:

org.Icsim

General

Description

Organization and Site

Repository

Lists

Team

Reports

Dependencies

Build

Java/Jar

Unit Tests

Project Files

Goal to IDE Action Mappings

Plugins

Build :

Clean :

Rebuild :

Javadoc :

Run :

Run Single :

Debug :

Debug Single :

Test :

Test Single :

Debug Test Single :

Multi Project Build :

Multi Project Clean :

-Dmaven.test.skip=true jar:install jas:install

clean

clean jar:install

javadoc

mevenide:run-artifact

-Dmaven.mevenide.run.classname=%CLASS% mevenide:run-class

mevenide:debug-artifact

-Dmaven.mevenide.run.classname=%CLASS% mevenide:debug-class

test

-Dtestcase=%TESTCLASS% test:single

-Dtestcase=%TESTCLASS% mevenide:test-single-debug

-Dgoal=jar:install multiproject:goal

multiproject:clean

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

Resolve Values

Apply

OK

Cancel