Modifying and Building CacheWolf on Linux

Table of contents

Scope of this document

This document describes how to set up an environment for building and modifying CacheWolf on a Linux system. It will show you how to set up the software needed. In case you want to use the eclipse IDE (integrated development environment) for modifying the sources, it will show you how obtain the sources of CacheWolf and how to set up an initial environment for further development.

This document will not tell you how to set up a Linux system in the first place. It is assumed that you already have a computer running Linux with a GUI (graphical user interface), that you know how to start a program on your computer and that you know the administrator password needed to install any software packages that might be missing.

All screen shots in this document were made on a computer running Fedora 10 with KDE. If you use a different flavor of Linux or another desktop environment, you mileage may vary.

Install missing software packages

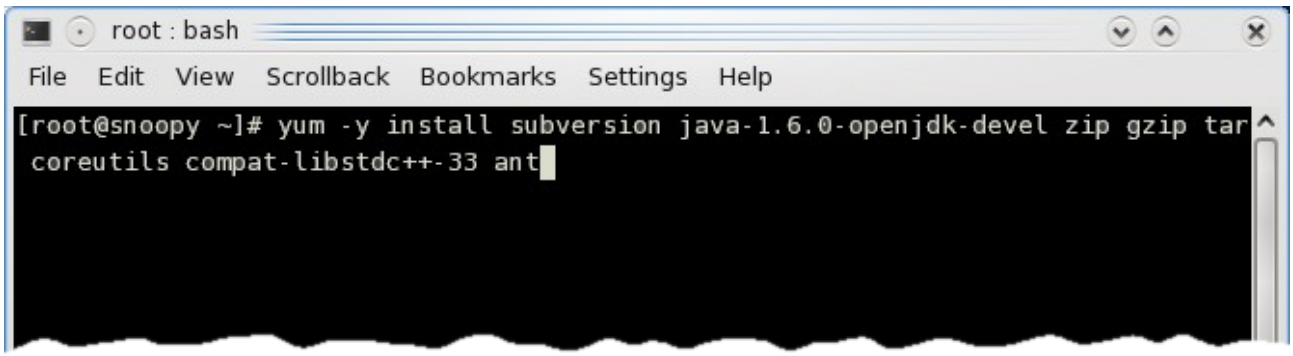
Please open a shell window and become superuser by typing `su -` pressing enter, and entering the administrator password.

Next we will install the missing software for building CacheWolf and optionally for the eclipse IDE. If you want to, you can also install everything in one go.

On Fedora 10 you can use yum to install any missing packages. To install the software packages

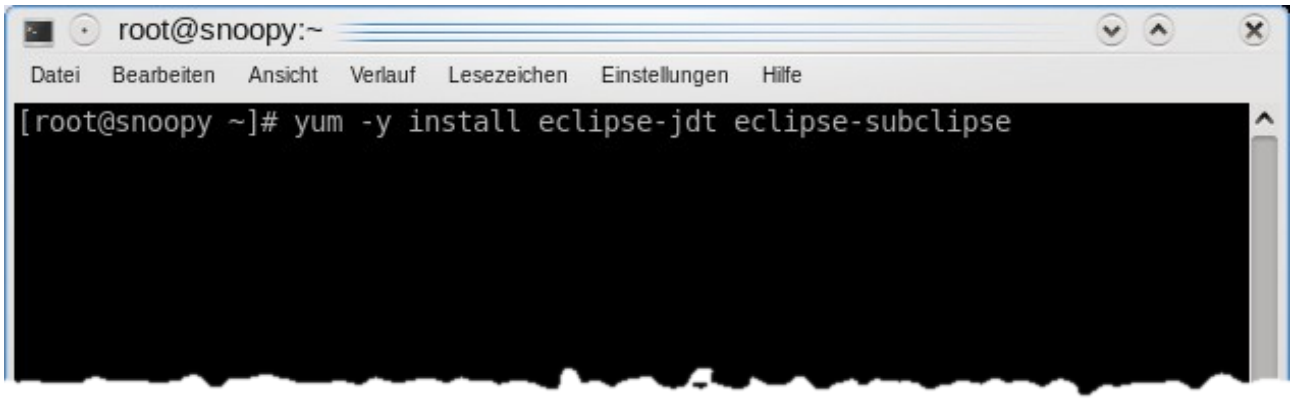
For the build

```
yum -y install subversion java-1.6.0-openjdk-devel zip gzip tar
coreutils compat-libstdc++-33 ant
```



For the eclipse IDE

```
yum -y install eclipse-jdt eclipse-subclipse
```



Make sure all arguments are on one line, press enter and wait for the installation to finish.

After that, type `exit` and press enter to leave the administrator mode.

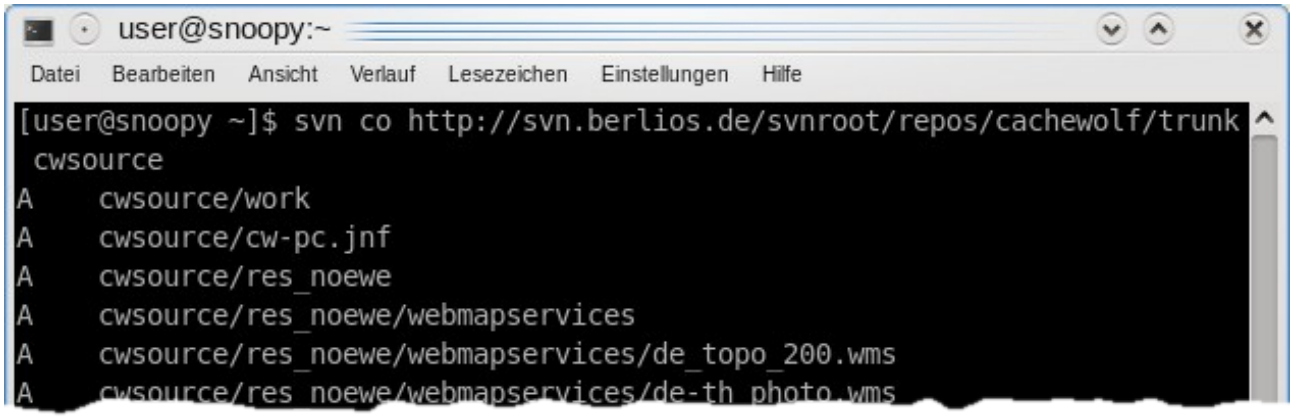
Getting the source code without eclipse

Please skip this chapter if you want to use the eclipse IDE for CacheWolf development. You will only need the commands described here, if you want to use an other IDE / editor or if you just want to compile CacheWolf from source.

Download CacheWolf source code with the command (everything on one line)

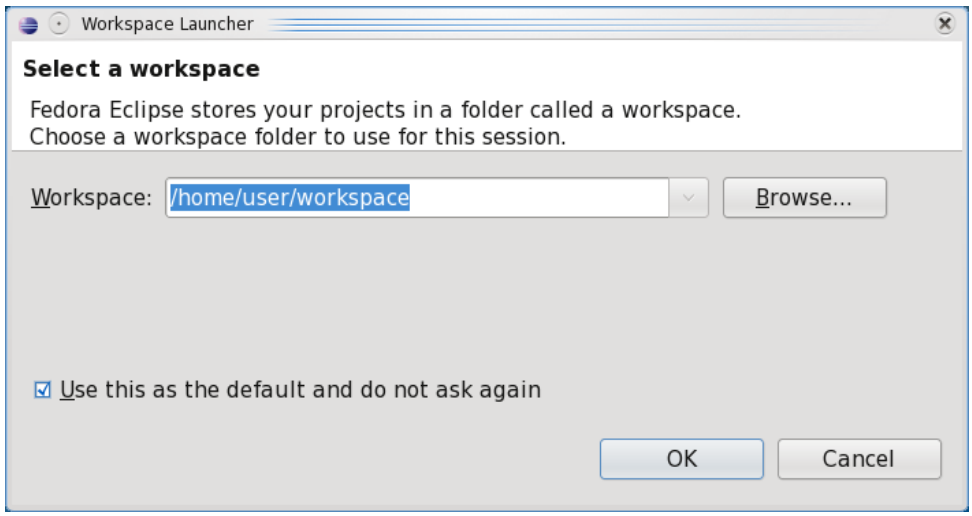
```
svn co http://svn.berlios.de/svnroot/repos/cachewolf/trunk cwsource
```

press enter and wait for the download to finish. After that you will find the sources in the directory cwsource where you can use any editor to make the changes you want.

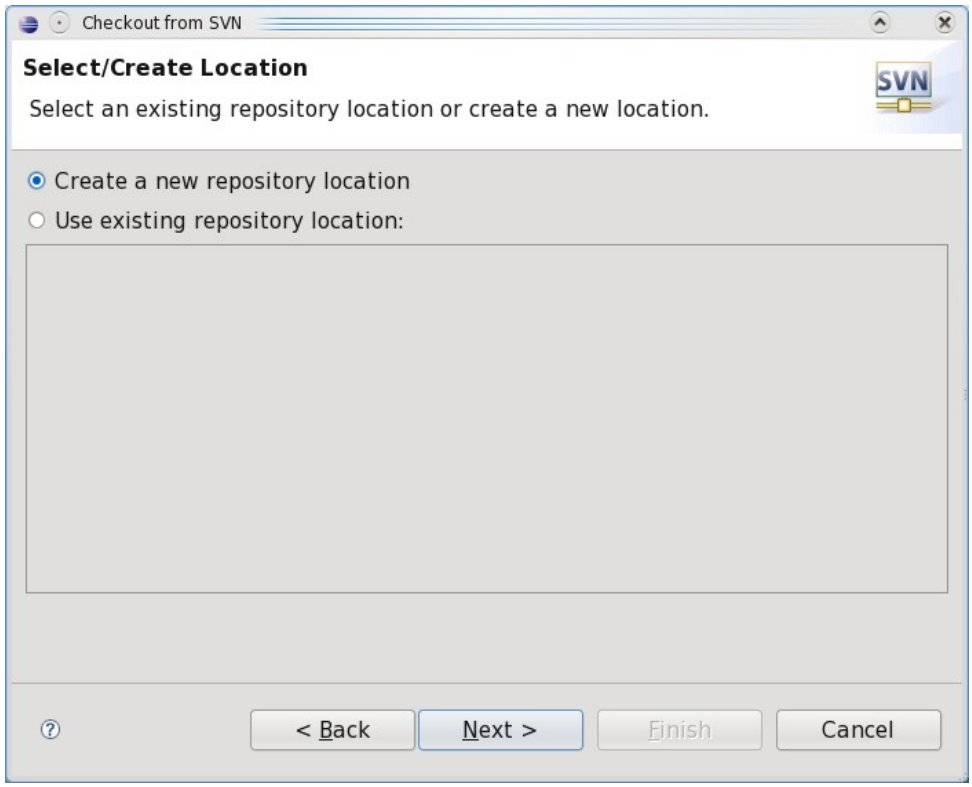


Getting the source code with eclipse

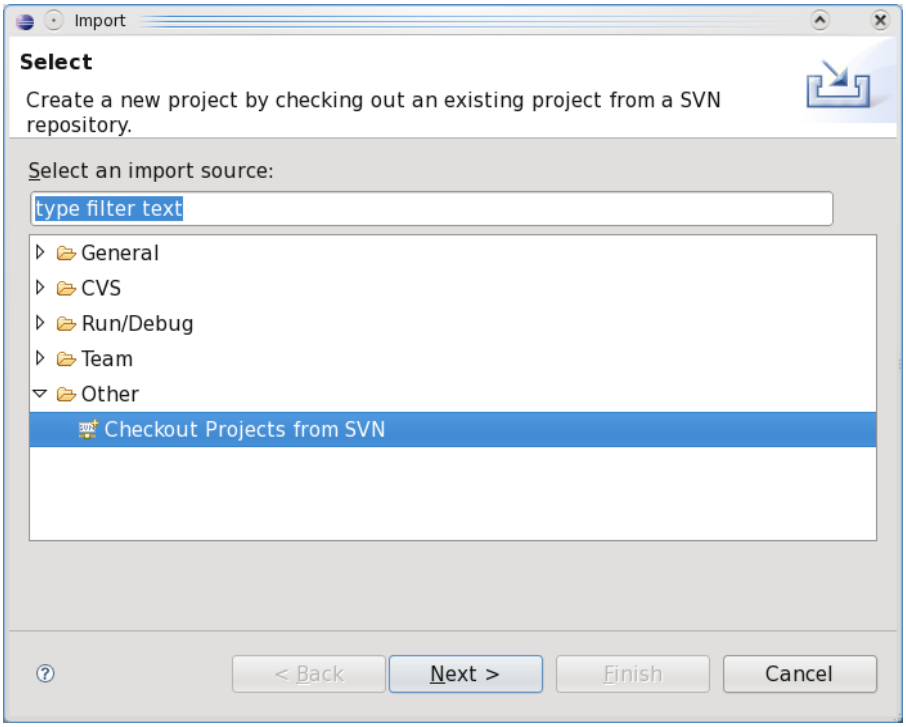
If you are using eclipse for the first time, you have to specify where you want to place your workspace to manage your projects. Accepting the default value usually is sufficient for most purposes. If you choose to enter anything else than the default directory please take a note where you put your projects, sine we will need this information again later on.



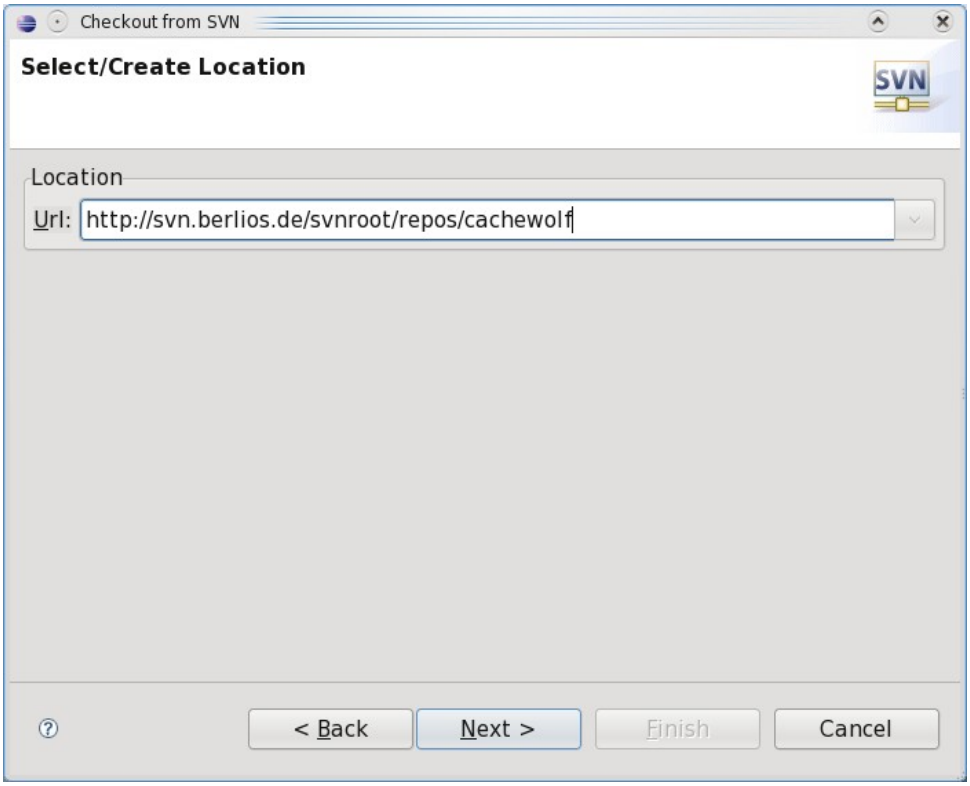
Next select the menu *File* and there the Option *Import* to start the import of the sources.



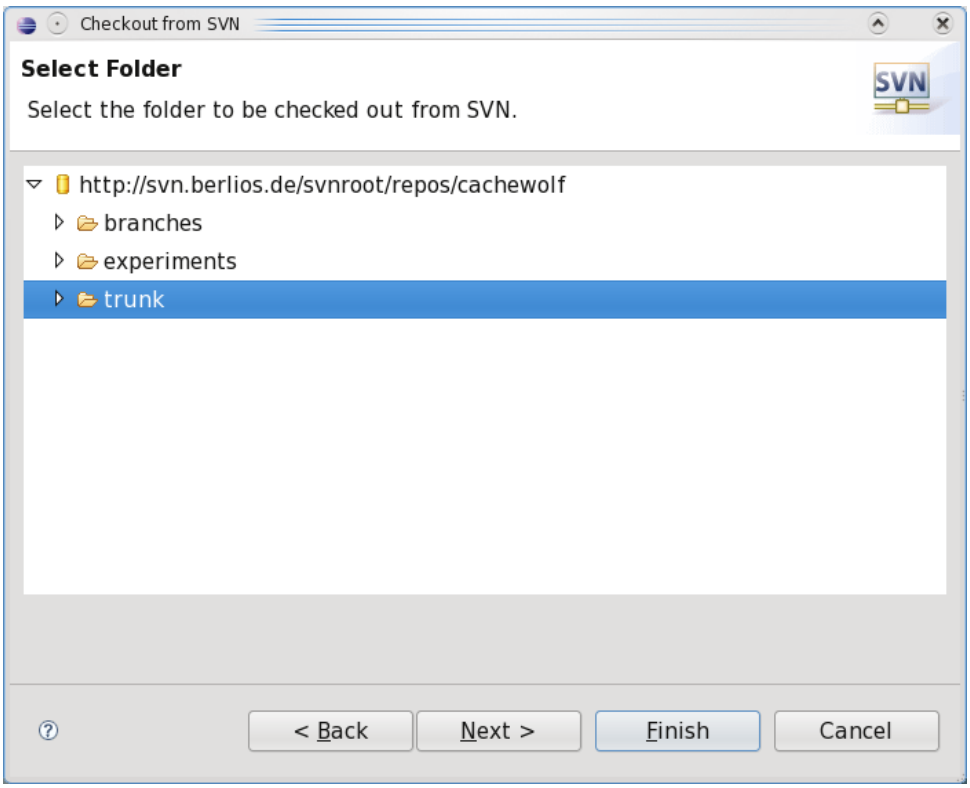
Select the input source *Other* and there *Checkout Projects from SVN*. The click on *Next*.



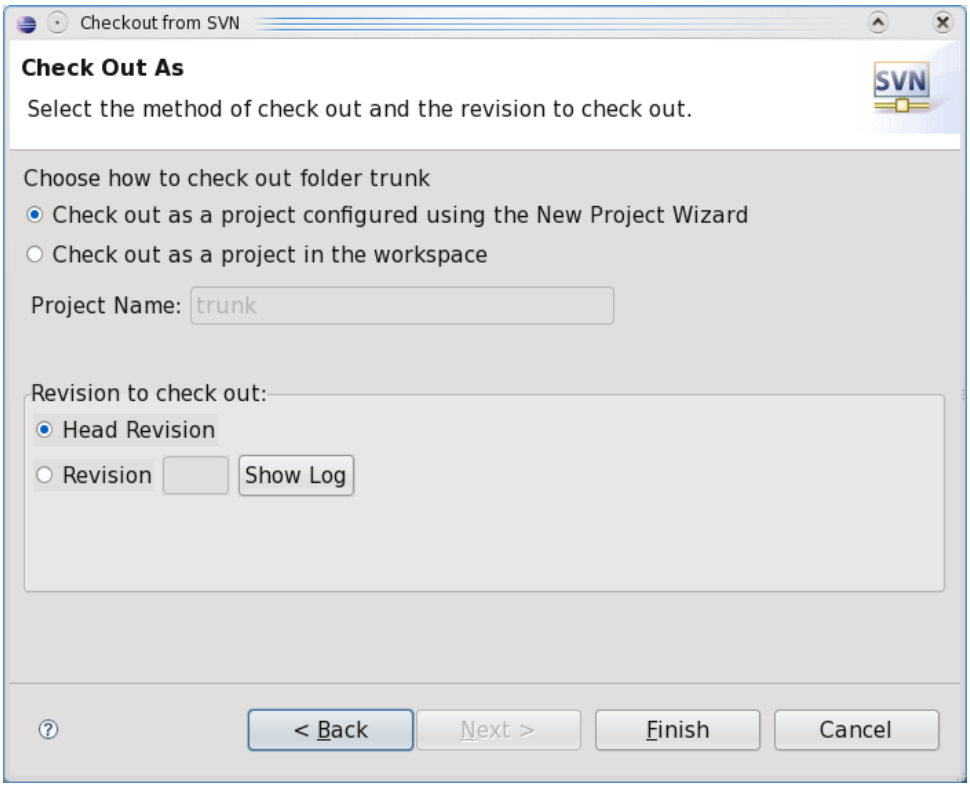
Select *Create new repository location* and click on *Next*.



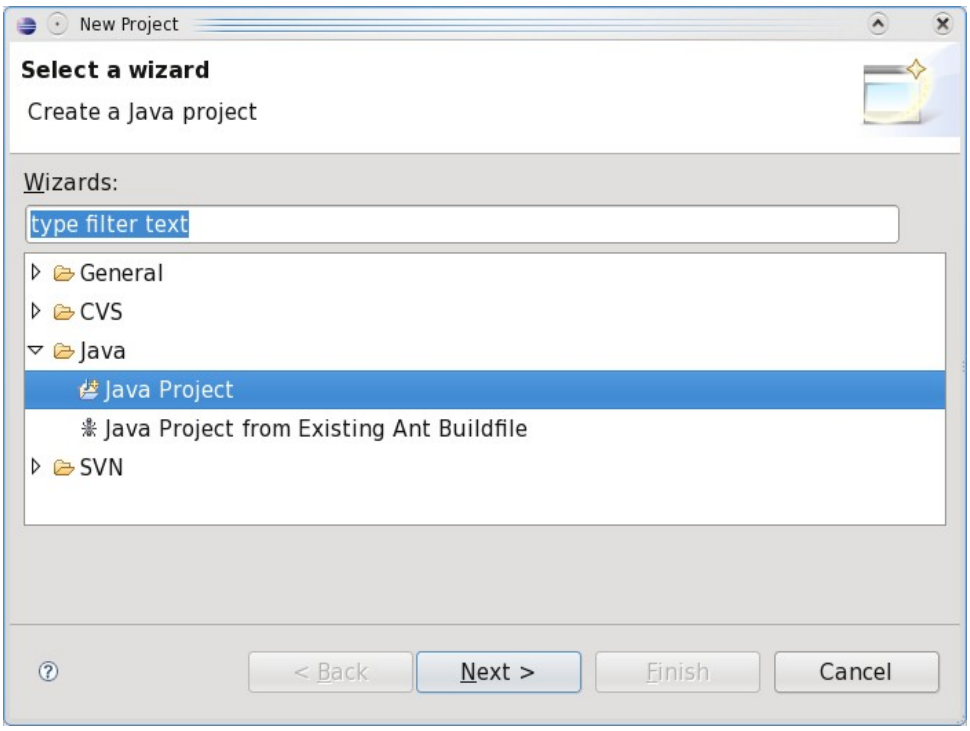
Enter http://svn.berlios.de/svnroot/repos/cachewolf as Location URL and click *Next*.



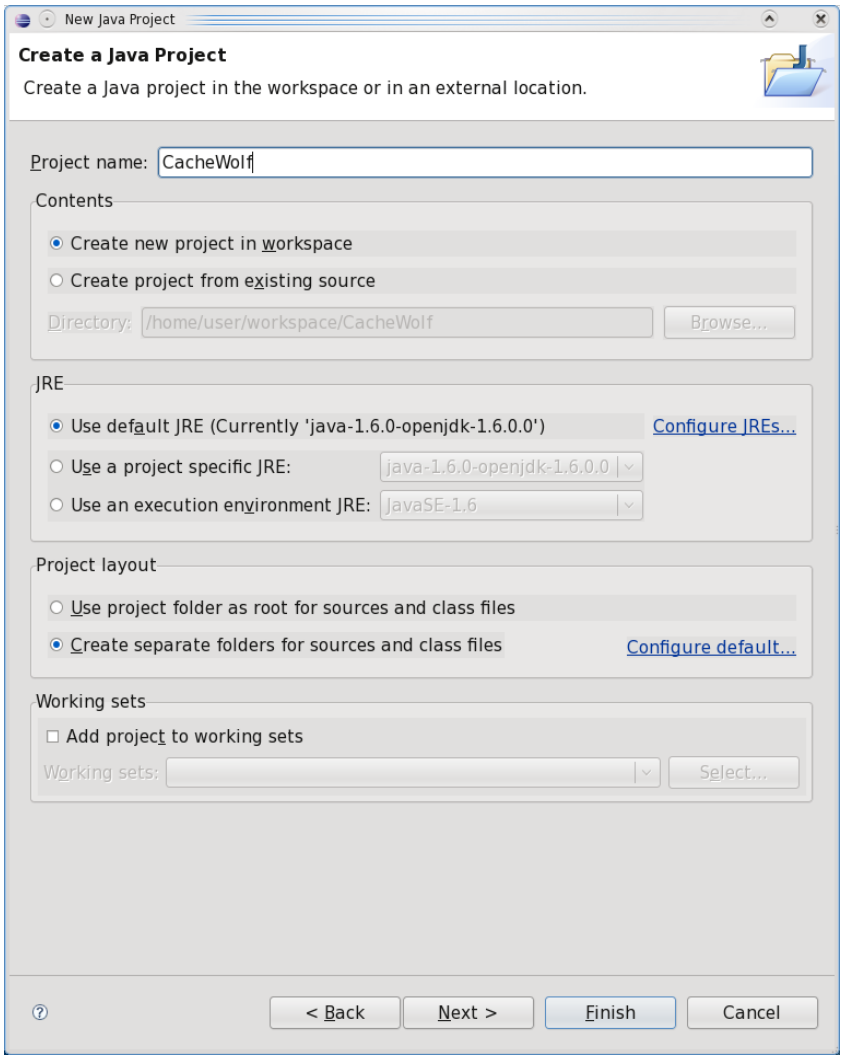
Select the Folder `trunk` to get the sources of the current CacheWolf version and click *Next*.



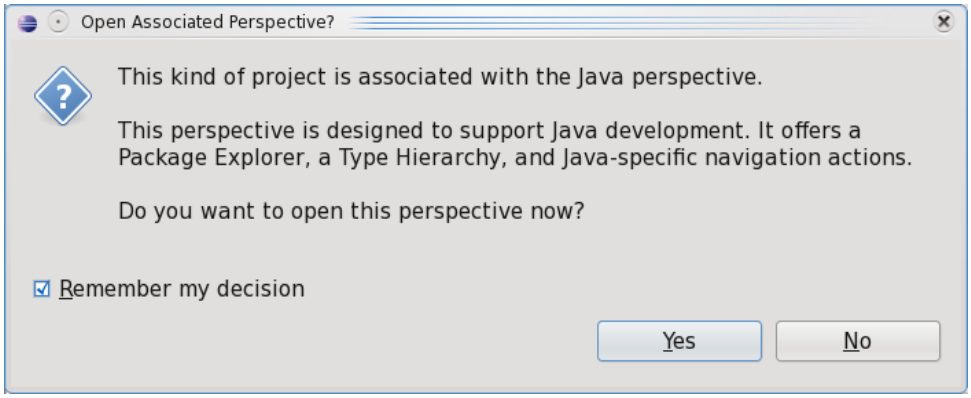
Check out the sources as *Project using the New Project Wizard* and select the *Head Revision* to get the most recent version of the source code. Click *Next* to continue.



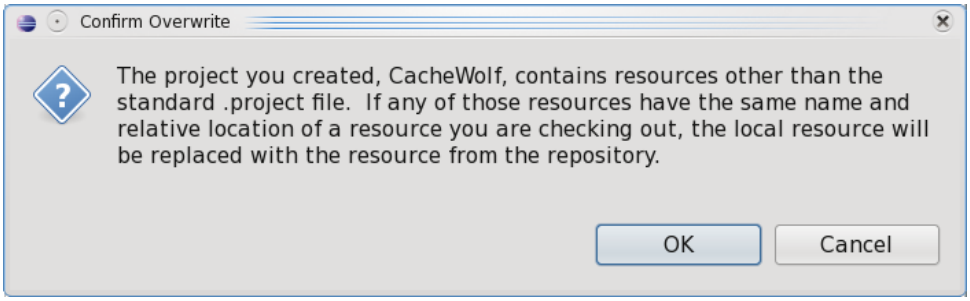
Select *Java Project* and then move on by clicking *Next*.



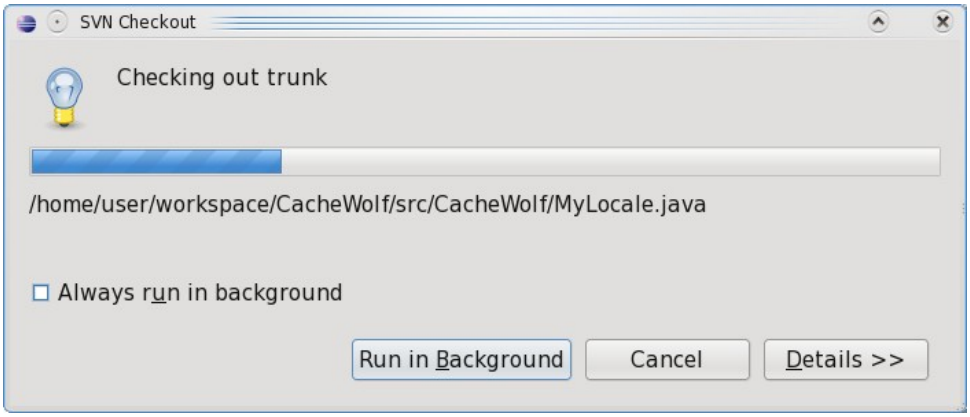
Give the Name CacheWolf to your project and select the other options according to the screen shot. The click *Finnish* to have your Project set up in your workspace.



Select `Yes` to open the project in the Java perspective.



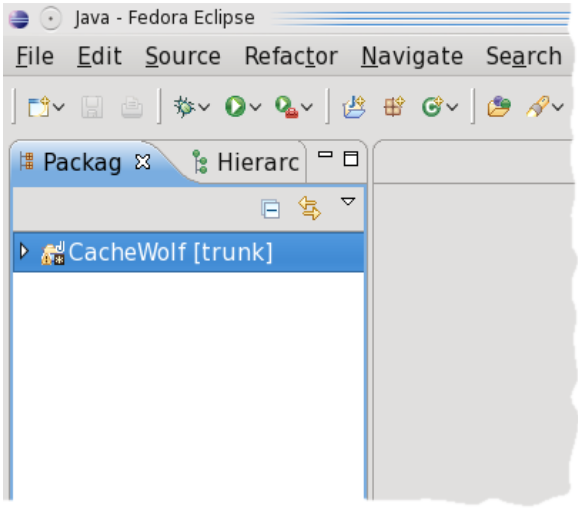
Please confirm, that the checkout of CacheWolf from the main repository may replace some resources in your freshly created project by clicking *OK*.



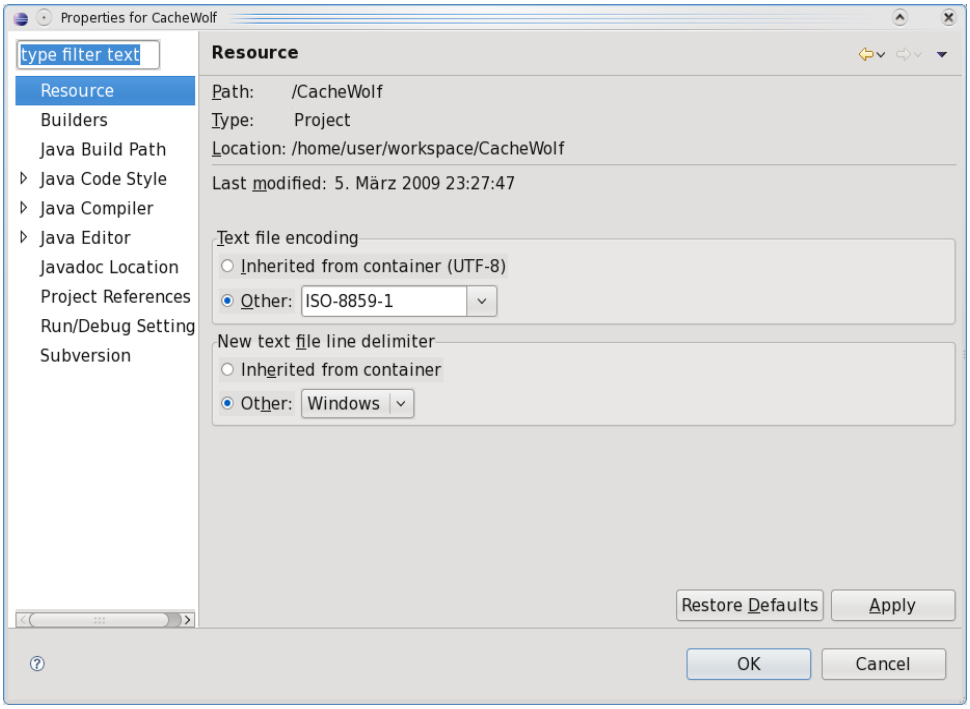
Now wait until the checkout has finished (time depends on the speed of your internet connection) and the main eclipse window reappears.

Configure your eclipse project

Next you have configure some properties of your newly created project to set up your environment for development and to ensure a smooth integration of changes you make back into the main CacheWolf repository.



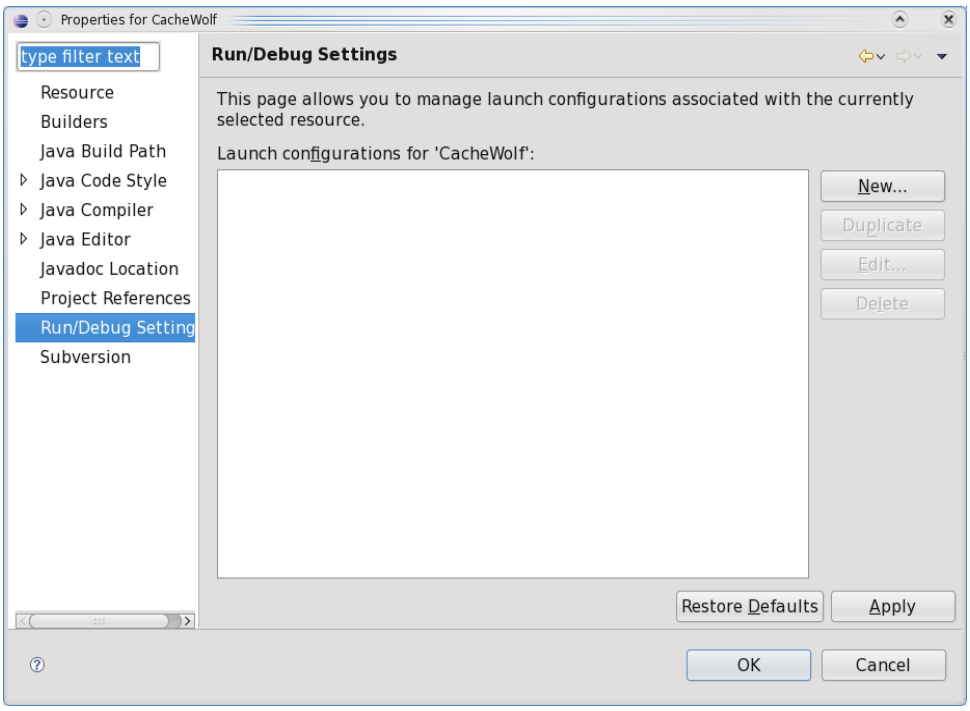
Select your CacheWolf project and press *Alt+Enter*.



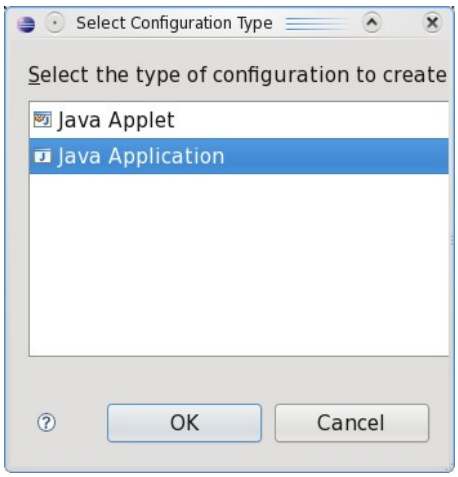
Select *Resources* and change your *Text file encoding* to *ISO-8859-1* and set your *line delimiter* to *Windows*. Click *Apply* and wait a couple of seconds for the software to catch up.

Important note:

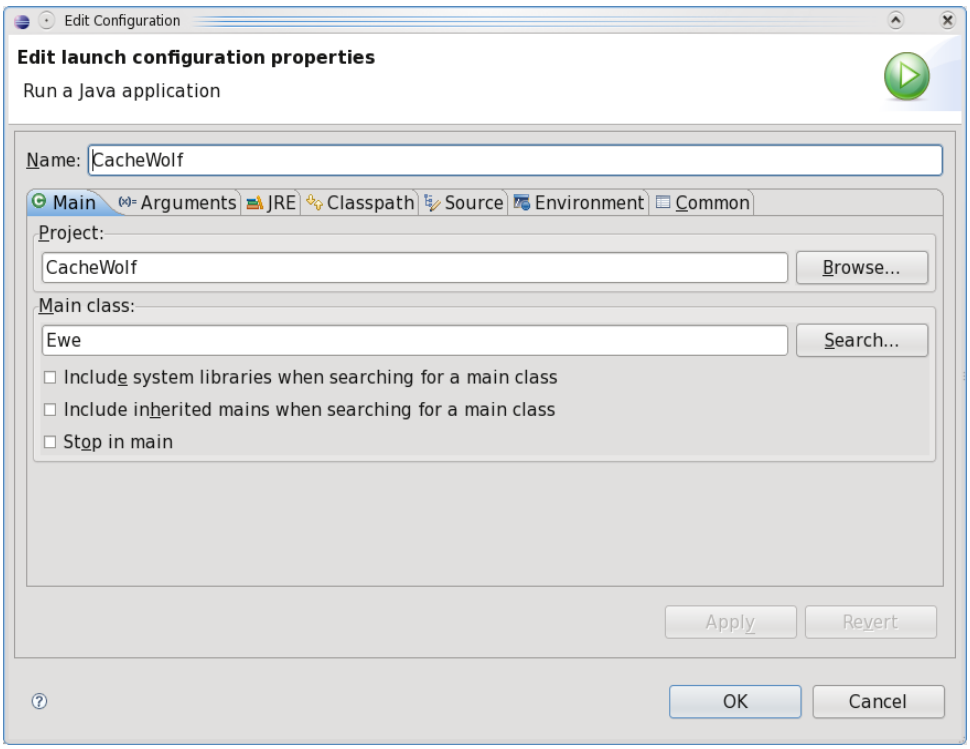
Most CacheWolf developers use windows and a cp1252 encoding. cp1252 support on Linux is somewhere between terrible and non existent. ISO-8859-1 will get you quite close, but still some differences remain. At the time of this writing this mainly concerns the files CWPoint.java and SafeXML.java. If you need any changes of files with characters not rendering OK on your screen, please ask a fellow developer using Windows to commit the changes to the repository to avoid any glitches. See http://en.wikipedia.org/wiki/ISO_8859-1#ISO-8859-1_and_Windows-1252_confusion for further information. This will not affect compilation and building of binary packages, but in eclipse this will be as close as you get for the time being.



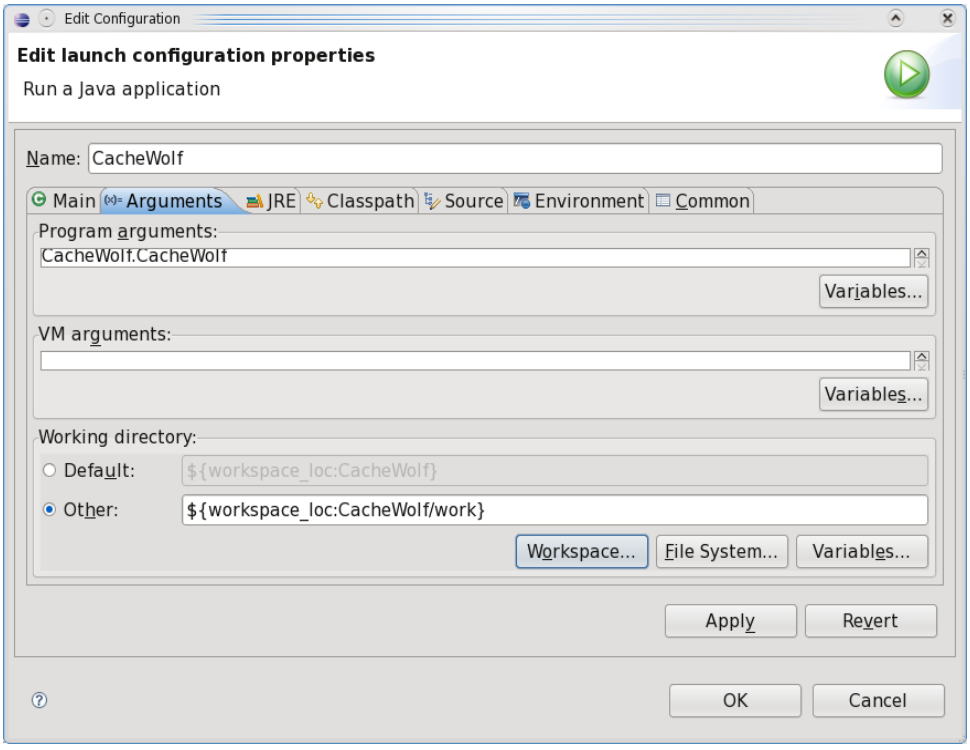
Next we will have to configure the Run/Debug Settings by creating a *New* configuration.



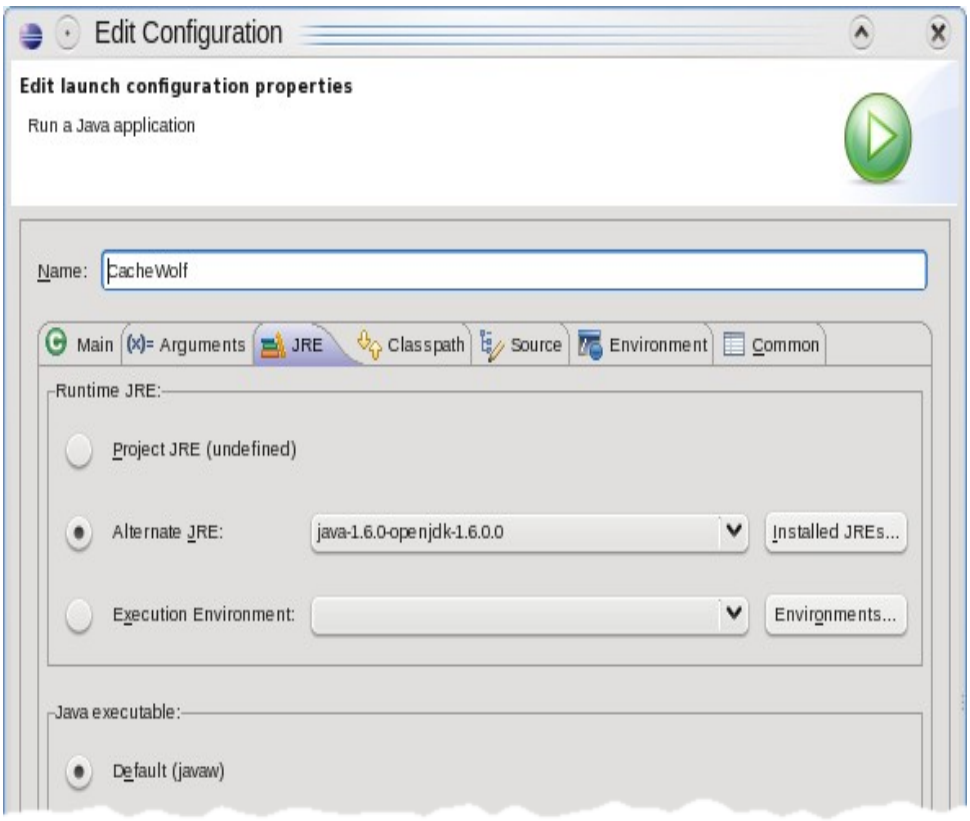
Select *Java Application* and Click *OK* to continue.



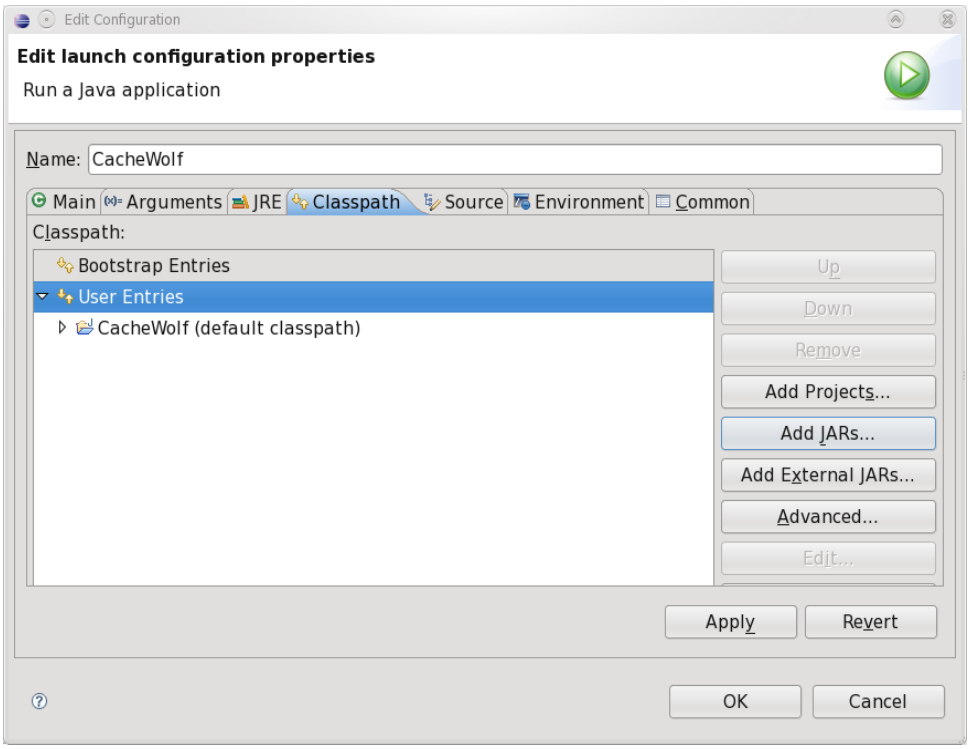
Enter the *Name* CacheWolf for your configuration and set the *Main class* to Ewe. Then click *Apply* and select *Arguments*.



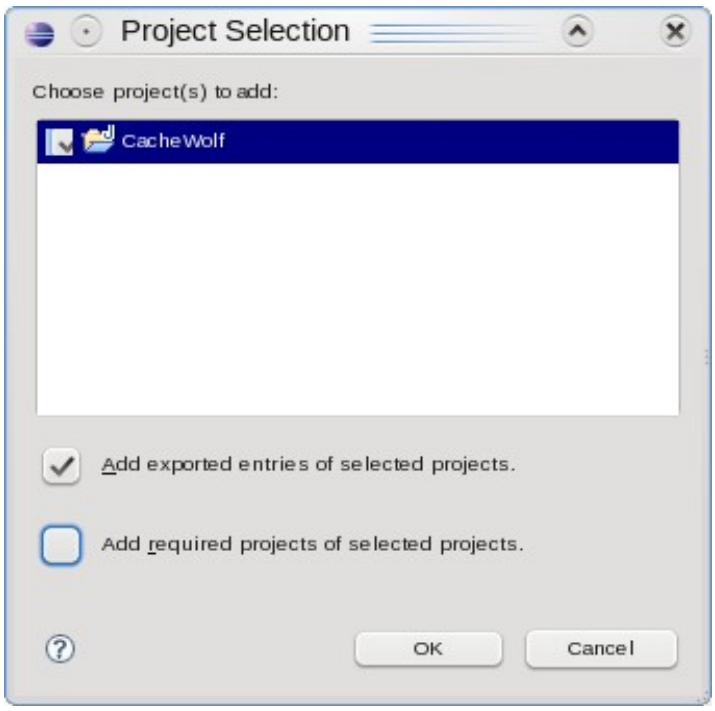
Enter `CacheWolf.CacheWolf` as *Program arguments* and change the *Working directory* by clicking on *Workspace* and selecting *work*. Click *Apply* before you select *JRE* to continue.



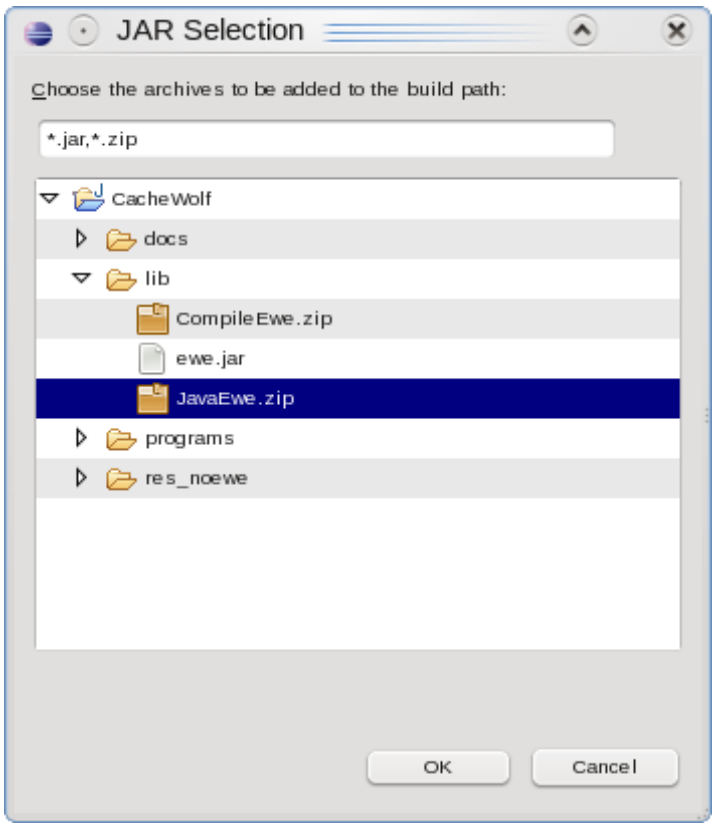
Make sure to select a valid JRE and move on to *Classpath*.



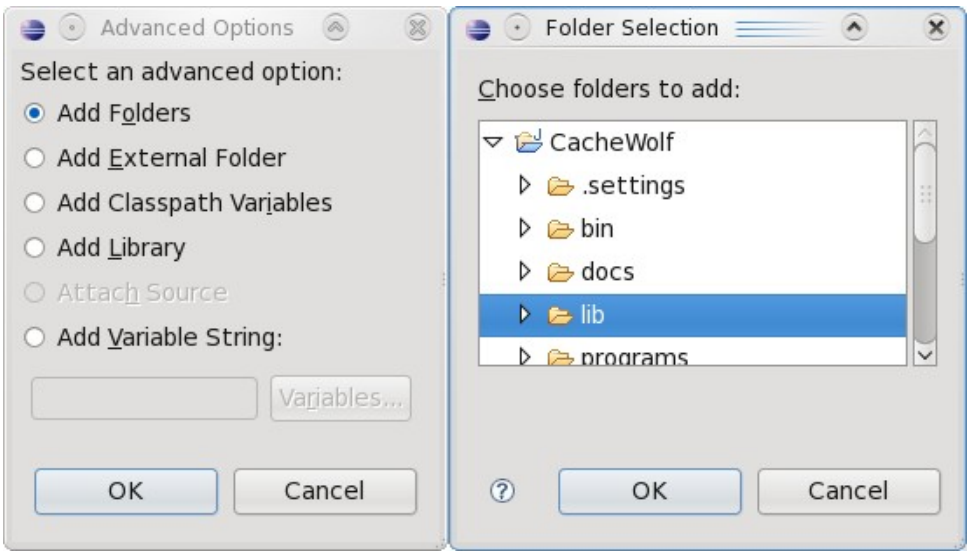
Select *User Entries* and the click *Add Projects*.



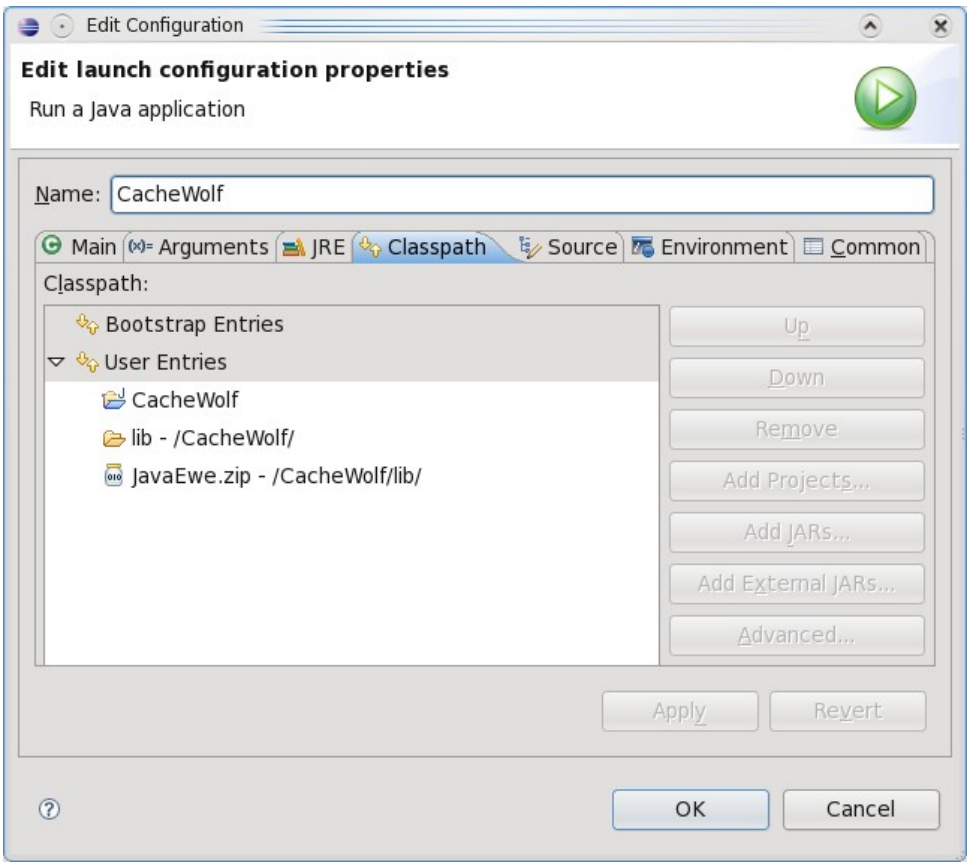
Select the *CacheWolf* project and make sure the *Checkbox* in front of it is checked. Click *OK* to proceed and then select *User Entries* and click *Add Jars*.



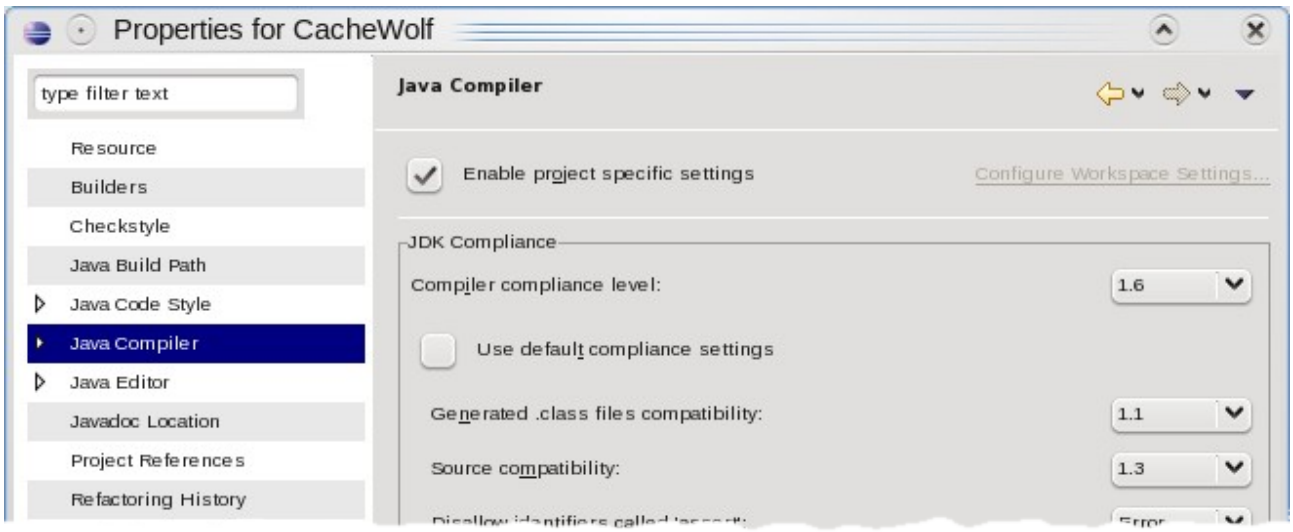
Select *JavaEWE.zip* in the *lib* directory and click *OK*. Again select *User Entries* and click *Avanced*.



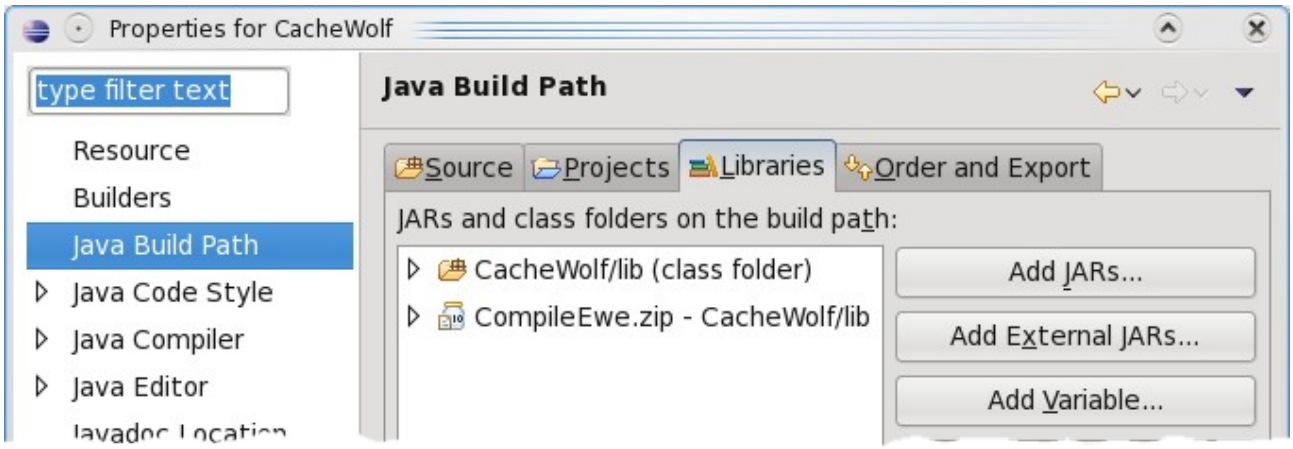
Select *Add Folders* and select the *lib* folder in your in your CacheWolf project. Click OK to confirm. Finally select *CacheWolf (default classpath)* and click *Remove*.



Your Classpath should now look like this. Click *Apply* and *OK*.

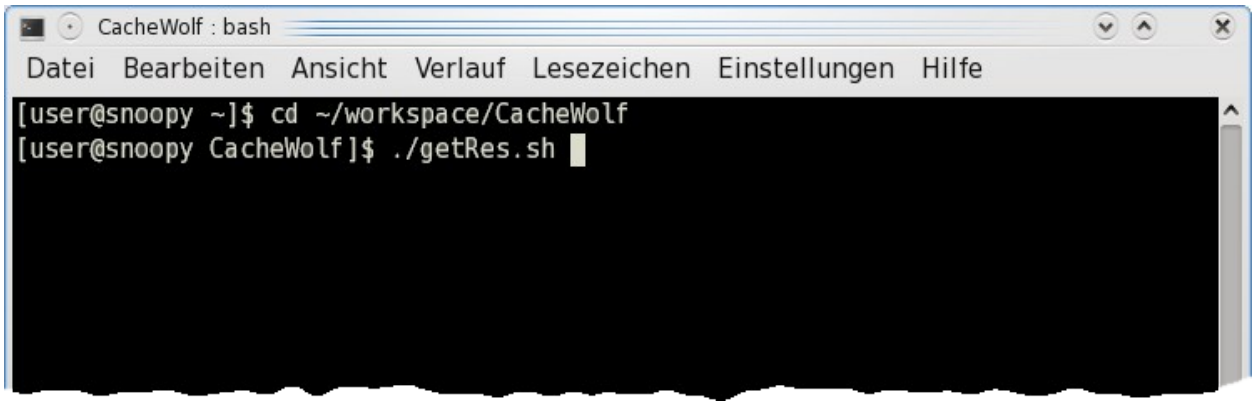


For the Java Compiler you should enable project specific setting. Classes should be compatible to 1.1 and Source to 1.3. These are the same settings which will be used later by the build mechanism for the packages and they enable eclipse to prohibit you from writing Java Code that could no be compiled later on anyway.



As a last step check the *Libraries* on your *Java Build Path*. If it does not match the screen shot above, add the missing with the buttons *Advanced* and *Add JARs* and remove anything which is surplus just as we did when setting the Classpath for Run Settings.

Now we need a shell to do the final step in setting up our environment:



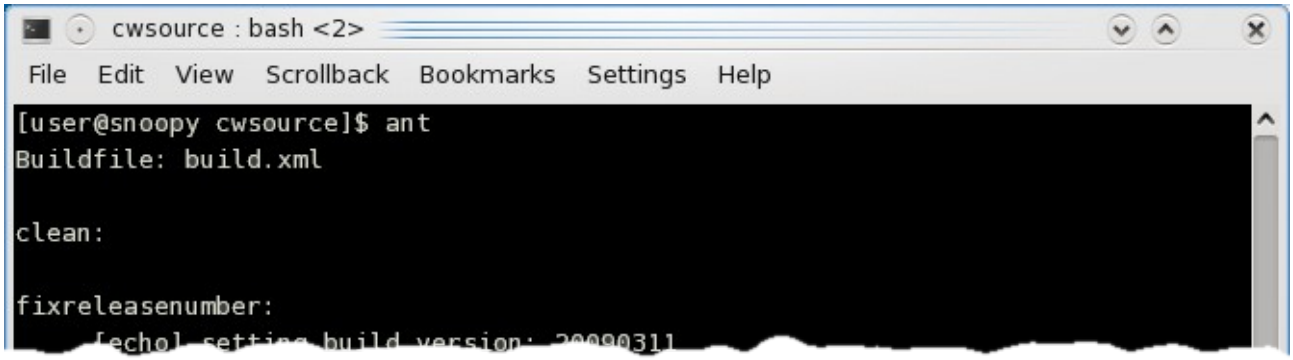
Change your directory to point to you CacheWolf directory within your workspace (the directory I asked you to remember quite at the beginning of the process) and call the script getRes.sh. Don't be confused if some error messages or warnings appear.

Now you can close the shell and make all the changes you want to make using the eclipse IDE. For an introduction to the various features of eclipse for software developers please refer to one of the many eclipse tutorials (http://www.google.com/search?q=eclipse+tutorial).

Building binaries

After you have done all the modifications you wanted to make to the CacheWolf sources, it's time to build some binaries.

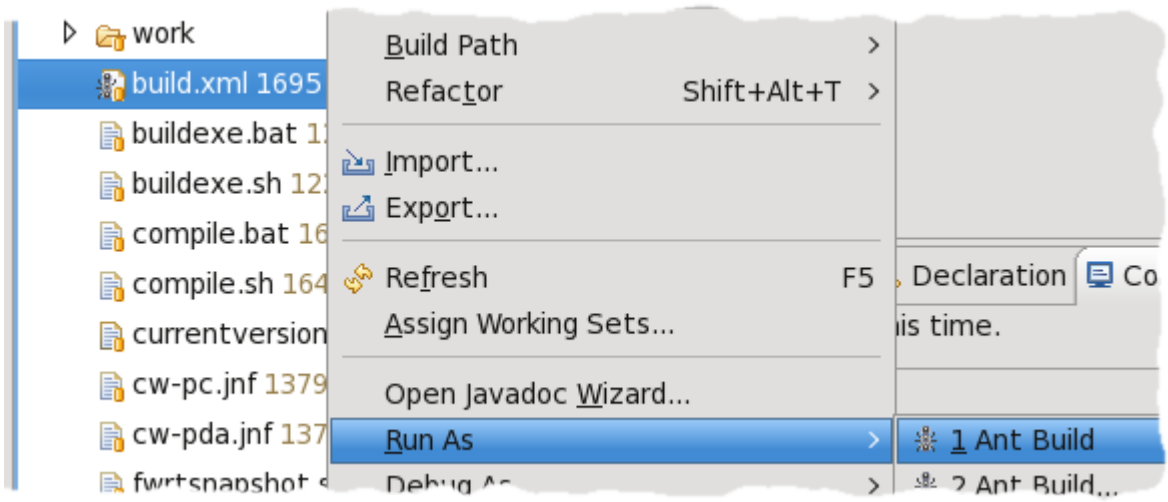
Command line



Just type the command *ant* in your main CacheWolf source directory.

After the build has finished you can find the generated packages in the published directory.

Eclipse



Right click on *build.xml*, select *Run As* and *Ant Build*. After the build has finished you can find the generated packages in the published directory of your CacheWolf directory.