

# HOW Tos

## Table of contents

1 Tweak Eclipse to emit code compatible with $\leq 5.0$ even if source level is set to 5.0 .....	2
2 Make annotations work.....	2
3 Make generic collections work.....	2
4 Tweak the batch compiler.....	2

## 1. Tweak Eclipse to emit code compatible with $\leq 5.0$ even if source level is set to 5.0

Add a file called `org.eclipse.jdt.core.prefs` in the `.settings` subfolder of your java project. TODO: add image. From within eclipse, you may also achieve this through setting the Java Compiler JDK Compliance settings to something not default, e.g. `source=1.4, target=1.3`

```
org.eclipse.jdt.core.compiler.codegen.targetPlatform=1.4
org.eclipse.jdt.core.compiler.compliance=1.5
org.eclipse.jdt.core.compiler.source=1.5
```

This will generate such a file which you can then alter (in text!) to the settings mentioned above. Note that eclipse will claim that these settings are not valid - you may ignore the message. I will provide a simple patch at some time which will remove the error message.

## 2. Make annotations work

Simply add the file `"java_lang_Annotations.jar"` to your build classpath. This way the eclipse compiler will see them and handle them correctly.

## 3. Make generic collections work

Simply add the file `"java_util_Collections14-5_0.jar"` to your build classpath. Make sure it is in **front** of any JRE Libraries. This way, the compiler will see them first and allow you to use them parametrized.

The reason it works is that the ported classes are just an "interface copy" of the 1.4 collection classes, but beefed up with parameter information. Eclipse will happily compile against them, but at runtime the normal JDK1.4 classes will be used. **Make sure these classes are only on the build path, but not on the classpath when running your application.**

### Note:

This will only work when running on JRE1.4

## 4. Tweak the batch compiler

**Note: this has become obsolete with Eclipse 3.2**

Tweaking the compiler in eclipse to spit out 1.4 code when in 5.0 source mode is one thing,

but running it in batch mode is a slightly different thing. Like in the UI, there are also some checks which disable source=5.0 and target < 5.0. Applying [this](#) patch simply disables these checks for 5.0->1.4.

Example:

```
<javac ...>
  <bootclasspath>
    <pathelement
path="{some_path}/java_util_Collections1_4-5_0.jar;{some_path_to_JRE14Runtime}/rt.jar"/
    </bootclasspath>

    <compilerarg compiler="org.eclipse.jdt.core.JDTCompilerAdapter"
line="-1.5 -source 1.5 -target 1.4"/>
</javac>
```