

2.3 Compiling an Aspect and Multiple Java Files

Problem

You want to conveniently compile a selection of aspects that are to be applied to multiple Java classes.

Solution

Create an AspectJ build configuration file titled *<appname>.lst* containing the names of all of the class files and aspects to be included in the compilation, similar to the example *.lst* file in Example 2-4.

Example 2-4. The contents of an example AspectJ build configuration .lst file

```
// File files.lst
com/oreilly/aspectjcookbook/MyClass.java
com/oreilly/aspectjcookbook/MyAspect.java
com/oreilly/aspectjcookbook/AnotherClass.java
com/oreilly/aspectjcookbook/AnotherAspect.java
```

Use the following command to instruct the ajc compiler to apply the aspects to the classes:

```
> ajc -argfile files.lst -classpath %MY_CLASSPATH% -d %MY_DESTINATION_DIRECTORY%
```

Discussion

The process by which the ajc tool completes the compilation of aspects and classes is largely transparent to the developer and can be treated as a black box. You shouldn't really worry about the interim steps that may be taking place inside the AspectJ compiler, short of a desire to get into development work on ajc itself.

It is important to consider the inputs to the build process; this is handled through the creation of the *.lst* build configuration file. The ajc compiler does not search the source or class path for files to compile; it must be told which files are to be involved in the compilation. This means that all of your source that is to be compiled with *aspects* must be fed directly to the ajc compiler. There are three ways to supply the files to be compiled to the ajc compiler (two of which are semantically equivalent):

The -argfile option

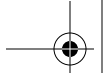
You can supply all the files within a *.lst* file by specifying the filename on the ajc command line with this option.

The @ option

This option is equivalent to the `-argfiles` option.

Directly list the files

You can simply specify the files on the command line when the ajc compiler is invoked.



See Also

Recipe 2.1 shows how to prepare a simple command-line build area for development using AspectJ; the AspectJ Development Environment Guide is available at <http://www.eclipse.org/aspectj> and provides more details on the runtime options and flags that the ajdoc tool supports; a full description of the AspectJ compiler process is available at <http://hugunin.net/papers/aosd-2004-cameraReady.pdf>; Recipe 2.8 shows how the *.lst* file can be used to vary the aspects that are woven for a specific particular build configuration.

