The code for the exception handler is then produced by traversing the chosen paths graph. The nodes shared in common with the main path are reset back to their state at the exact point mid main path traversal when the exception is thrown. This means that all cached and calculated values from the main path can be reused in the exception handler, thus avoiding repeating any work.

Bytecode manipulation

Converting our compiled sections of bytecode into loadable classes is something for which there are a number of established and well-engineered solutions. Apache's Byte Code Engineering Library (BCEL) considers itself to conveniently "analyze, create, and manipulate (binary) Java class files."* ASM is an "all purpose Java bytecode manipulation and analysis framework."†

Unfortunately, all we want to do is modify a single method (always the same one) in a single skeleton class. We can only generate a small subset of the possible set of bytecode sequences, and we don't need to provide any analysis tools. It would appear then that both BCEL and ASM are overkill for our needs. Instead, we developed a custom bytecode manipulation library with very limited capabilities that exactly matched what we needed. For example, our stack-depth algorithm is tuned to rapidly assess the maximum stack depth of our methods (so that they can pass verification). Although this algorithm does not work for general class compiling, it is sufficient and more efficient for our purposes.

TIP #9: BEWARE EXTERNAL LIBRARIES

Avoid using external libraries that are overkill for your purposes. If the task is simple and critical, then seriously consider coding it internally; a tailor-made solution is likely to be better suited to the task, resulting in better performance and less external dependencies.

Class Loading and Unloading, but on a Big Scale

So now we have classes, and they need to be loaded. The obvious first question is, "How many?" Figure 9-11 gives an illustration of the scale of the problem. Loading 100,000 classes into a single JVM can prove to be a slight challenge.

* *http://jakarta.apache.org/bcel*

† *http://asm.objectweb.org/*

Download at WoweBook.Com