CHAPTER 5

The Abstract Factory Pattern

The Abstract Factory pattern is one level of abstraction higher than the Factory Method pattern. You can use this pattern to return one of several related classes of objects, each of which can return several different objects on request. In other words, the Abstract Factory is a factory object that returns one of several groups of classes. You might even decide which class to return from that group by using a Simple Factory.

One classic application of the Abstract Factory pattern is when your system needs to support multiple look-and-feel user interfaces, such as Windows *9x*, Motif, and Macintosh. You tell the factory that you want your program to look like Windows, and it returns a GUI factory that returns Windows-like objects. Then when you request specific objects, such as buttons, check boxes, and windows, the GUI factory returns Windows instances of these visual interface components.

In Java 1.2, the pluggable look-and-feel classes accomplish this at the system level so that instances of the visual interface components are returned correctly once the program selects the type of look and feel. In the following code, we find the name of the current windowing system and then tell the pluggable look-and-feel (PLAF) Abstract Factory to generate the correct objects.

```
String laf = UIManager.getSystemLookAndFeelClassName();
try {
        UIManager.setLookAndFeel(laf);
    }
    catch(UnsupportedLookAndFeelException exc){
        System.err.println("Unsupported L&F: " + laf);
  }catch(Exception exc)
        System.err.println("Error loading " + laf);
  }
```