Why Developers Should Not Write Programs That Call 'sun' Packages

The classes that Sun includes with the Java 2 SDK, Standard Edition, fall into package groups java.*, javax.*, org.* and sun.*. All but the sun.* packages are a standard part of the Java platform and will be supported into the future. In general, packages such as sun.*, that are outside of the Java platform, can be different across OS platforms (Solaris, Windows, Linux, Macintosh, etc.) and can change at any time without notice with SDK versions (1.2, 1.2.1, 1.2.3, etc). Programs that contain direct calls to the sun.* packages are not 100% Pure Java. In other words:

> **The java.*, javax.* and org.* packages documented in the Java 2 Platform Standard Edition API Specification make up the official, supported, public interface.**
> If a Java program directly calls only API in these packages, it will operate on all Java-compatible platforms, regardless of the underlying OS platform.
>
> **The sun.* packages are not part of the supported, public interface.**
> A Java program that directly calls into sun.* packages is *not* guaranteed to work on all Java-compatible platforms. In fact, such a program is not guaranteed to work even in future versions on the same platform.

For these reasons, there is no documentation available for the sun.* classes. Platform-independence is one of the great advantages of developing in the Java programming language. Furthermore, Sun and our licensees of Java technology are committed to maintaining backward compatibility of the APIs for future versions of the Java platform. (Except for code that relies on serious bugs that we later fix.) This means that once your program is written, the class files will work in future releases.

Each company that implements the Java platform will do so in their own private way. The classes in sun.* are present in the SDK to support the Sun implementation of the Java platform: the sun.* classes are what make the Java platform classes work "under the covers" for the Sun Java 2 SDK. These classes will not in general be present on another vendor's Java platform. If your Java program asks for a class "sun.package.Foo" by name, it may fail with ClassNotFoundError, and you will have lost a major advantage of developing in Java.

Technically, nothing prevents your program from calling into sun.* by name. From one release to another, these classes may be removed, or they may be moved from one package to another, and it's fairly likely that their interface (method names and signatures) will change. (From the Sun point of view, since we are committed to maintaining the Java platform, we need to be able to change sun.* to refine and enhance the platform.) In this case, even if you are willing to run only on the Sun implementation, you run the risk of a new version of the implementation breaking your program.

In general, writing java programs that rely on sun.* is risky: they are not portable, and are not supported.