Java Survival Guide

by Craig Newmark and Kandy Arnold

Sponsored by Apple Computer, Inc.
Developer Relations Group
for the Apple Media Program

media

Java Survival Guide

By Craig Newmark and Kandy Arnold

As much as it is an object-oriented programming language inspired by C++, Java is a culture for the wired and caffeinated 1990s. If you're not convinced, just take a look at its designer-squiggle logo. And while Java may be as trendy as sipping lattes at your local coffehouse, it's an Internet phenomenon that's here to stay. In this article, we offer you an overview of the Java environment and new developments to watch for.

Java was designed from the ground up for the things the Internet needs, such as greatly enhanced security, reliability, and maintainability. In addition to being a programming language, Java is a complete development environment, including a Java virtual machine (JVM), the architecture and class libraries that run on many platforms from the Mac OS to Windows 95. Developers that write to this new environment can write to these libraries and create applications that run unmodified on any platform—just as HTML is a standard, crossplatform way to create content for the Internet, Java is a standard, crossplatform way to write software for the Internet.

The first mainstream version of the Java standard, known as "JDK 1.0.2" had some problems typical of a first release, including bugs in the libraries and poor performance, but these problems are being addressed very rapidly. Apple's internal test package for Java now includes more than 8,500 tests to ensure the maximum degree of compatibility. This doesn't mean that Java isn't moving forward—progress on Java is moving rapidly. JavaBeans is probably the most significant of these areas, and that is part of the "JDK 1.1" standard. JavaBeans allows you to stich together prebuilt Java components quickly and easily into large applications. These components pave the way for a whole new generation of visual Java tools, so Java development gets easier and faster. If you need a component that doesn't exist, you can either program a JavaBean yourself or hire a Java programmer to write the needed JavaBean for you. Either way, you save time by using existing code for at least part of any given project. For more details, visit http://java.sun.com/features/1997/feb/jdkbeans.html.

It doesn't stop with JDK 1.1, JavaSoft plans to introduce several additional improvements in JDK 1.2, coming in the third quarter. The biggest part of JDK 1.2 is the Java Foundation Classes, which are being jointly developed by Apple, IBM, Netscape, and Sun, are graphical user interface libraries based on Netscape's Internet Foundation Classes (IFC) and layered on Java's AWT. See http://java.sun.com/pr/1997/may/p r970513-01.html and http://www.javaworld.com/javaworld/jw-06-1997/jw-06-javafuture.html for further details. To learn more about Apple's support for Java Foundation Classes, check out the announcement at http://java.sun.com/pr/1997/may/pr970513-01.html.

Additional Java technologies under development include...
The Java Electronic Commerce Framework—a set of financial- and transaction-oriented classes and interfaces, which are being released as a series of "cassettes," or small commerce components that are similar in function to JavaBeans. The first cassette in the series is the Java wallet. For details, see http://developer.javasoft.com/developer/readAboutJava/JECF.html and http://developer.javasoft.com/developer/readAboutJava/JECF2.html.

The HotJava browser offers a more secure environment in which to run Java applets. Currently only available for the Solaris and Windows platforms, a Mac version is scheduled for second quarter release. See http://java.sun.com/products/hotjava/index.html for more information.

The PersonalJava and EmbeddedJava APIs are part of Sun's JavaCard API environment, which are designed to port Java functionality to information appliances and network PCs and to hardware devices with limited display capabilities, respectively. Visit http://java.sun.com/pr/1997/april/pr970402-02.html for details.

Java's media and communications APIs promise increased support for dynamic media types such as:

animation, 2-D graphics
3-D graphics
media players
sound, and telephony

Apple is integrating Java throughout it's system software. For the Mac OS, it has developed the Mac OS Runtime for Java (MRJ). Mac OS 8, available this summer, will include the MRJ 1.0, with support for the JDK 1.0.2. This integration lets you know that you'll have support for Java right out of the box. MRJ 1.5, available online at <http://applejava.apple.com/> later this summer, is a performance update for the MRJ that features dramatic speed increases, with some tests showing over ten times the performance! MRJ 2.0, which provides support for JDK 1.1 and JavaBeans, is already in developer seeding. For more information about OS 8.0 visit http://www.macos.apple.com/macos8/index.html.

Apple is also working on Rhapsody, the next generation OS based on NeXT technology. Java developers can rest easy - Rhapsody will fully support Java. Not only that, but developers will be able to write any Rhapsody application in Java and take advantage of Rhapsody specific features, making it that much easier to create great applications.

This summer both Netscape Communications of Mountain View, California, and Microsoft Corporation of Redmond, Washington, are releasing new versions of their browsers that are designed to take advantage of improvements introduced to Java through the release of its JDK 1.1. Along with the JavaBeans component software innovation, JavaSoft has improved the Abstract Window Toolkit, or AWT, which is a part of the Java Foundation Classes that Apple is incorporating into the Mac OS 8.0 and Rhapsody.

With these changes, Sun addresses the existing problem areas in Java and maps out a plan that supports its battle cry to write applications once and run them anywhere. Java's "sandbox" feature that limits an applet's access to system files offers the most secure component architecture now available, which should prove attractive to the enterprise market. The language's platform-independence ensures its longevity. For prognostications, check out David Orchard's JavaWorld article at http://www.javaworld.co m/javaworld/jw-06-1997/ jw-06-javafuture.html.

Resources

Gamelan
http://www.gamelan.com

JavaSoft
2550 Garcia Ave.
Mountain View, CA 94043
(800) 528-2768
(512) 434-1591
http://java.sun.com

JavaWorld Magazine
http://www.javaworld.com