REVIEW: Development Software

**The latest release of the Java Development Kit, version 1.1, charts the future of the language. Java Development Kit, Take Two**

By Rick Grehan

1162 letters

1997．4 .1

[BYTE](javascript:void(0))

BYTE

145

Vol. 22, No. 4

(Copyright 1997 McGraw-Hill, Inc.)

Although you certainly can use Sun's Java Development Kit (JDK) to construct Java applets and applications, that has never been its main purpose. The JDK is, first and foremost, a functioning specification for the Java environment, although it also addresses much in the Java world that has nothing to do with Java as a language.



I recently tried the much-anticipated beta version of the latest JDK, version 1.1, which is available for downloading from [http://java.sun.com](javascript:void(0)). I examined not so much the quality of the supplied development tools but rather the new and improved APIs and their accompanying specifications.

Function over Form The full JDK consists of two downloads. One is an archived file of all the command-line development tools and run-time support files; the other is the documentation. As of this writing, JDK 1.1 is available for the Solaris and Windows 95/NT platforms. Other platforms will certainly be available soon.

The JDK comes with all the tools you need to build and run Java applets and applications. You get a compiler, a debugger, Java VM, an applet viewer, and other accessory executables that support some of the new security features and the Java archive file format (JAR, explained later). The JDK also includes all the necessary Java class files.



But the JDK is not a visual development system, like Microsoft's J++ and Symantec's Cafe, nor is it supposed to be. Rather, it's a hands-on laboratory for the evolving Java environment. It is simultaneously robust and primitive--robust in the sense that you can construct and run full-blown and ready-to-distribute Java applets and applications, and primitive in the sense that the package's tools all run from the command line. (That includes the debugger--and there are few things that are more primitive than a command-line debugger.)

New and Improved The package's list of enhancements and additions is long (see the text box ``New in JDK 1.1'' on page 146), but some of them stand out as truly significant. Most striking are the changes to the Abstract Windowing Toolkit (AWT), whose Win32 version's internals have been complete-ly rewritten. In addition, the AWT's new event model promises to improve performance as well as help programmers write more-comprehensible code (see the Tech Focus on page 146).

The aforementioned Java archive file specification, JAR, spells out the format for bundling an applet's components--classes, images, and sounds--into a single compressed and optionally secure file. (One of the supplied tools builds JAR files.) Because each individual file transfer through HTTP requires its own separate TCP virtual circuit, archiving applet components into a single compressed file means that 1.1-compatible servers will deliver applets to browsers faster.

Java Database Connectivity (JDBC), the documentation for which is a sep-arate file that prints out to a small book, brings Open Database Connectivity (ODBC)-style database access to Java. The JDK's supplied JDBC-ODBC bridge software implements JDBC via ODBC, allowing you to experiment with JDBC using ODBC drivers.

Rounding out the more important additions to this release are new wrapper classes that the Java language should have had to begin with: wrappers for byte, short, and void primitives. A new numeric class, bignum, enables programs to manipulate numbers of arbitrary precision. You can alter the rounding behavior of the bignum class, and therefore optimize its precision, depending on whether your requirements are toward scientific or business computing.

Good News and Bad News Java programmers and users have plenty to be happy about with the release of JDK 1.1. The improvements in the AWT's performance should be apparent to anyone running applets in a 1.1-compatible browser, and the AWT's new event model will make Java developers' lives easier. The inclusion of JDBC is already spurring the appearance of JDBC development systems (e.g., Symantec's Visual Cafe Pro), and the remote method invocation (RMI) API opens the possibility of distributed Java applications.

Tarnishing the flip side of the coin are the usual corrosives accompanying any software upgrade: Programmers who invested time and mental energy in those elements of Java that are either changing or being eliminated will have to shift some mental gears. Happily, the JDK 1.1 documentation offers some guidelines for upgrading existing source code, and JavaSoft has taken substantial steps toward providing reasonable backward compatibility. For example, the new AWT will continue to support the previous event model. I was able to compile and run BYTE's Java benchmark suite (jBYTEmark) with no problem, even though jBYTEmark was built using the old event model.

By all means, if you have not yet downloaded JDK 1.1, do so (it's free). But recognize it for what it is: not a commercial Java development package, but a source of valuable documentation and a well-engineered testing ground for the next generation of Java programs.

Java Development Kit 1.1

Free for noncommercial use

(runs on Windows 95/NT or Solaris)

JavaSoft

(a business unit of Sun Microsystems, Inc.)

Cupertino, CA

(408) 343-1684

[http://www.javasoft.com](javascript:void(0))

RATINGS

TECHNOLOGY H H H H H

IMPLEMENTATION H H H

HHHHH Outstanding HHHH Very Good HHH Good HH Fair H Poor

New in JDK 1.1

Improved Abstract Windowing Toolkit (AWT)

-- Rewritten Win32 version.

-- New event model.

-- Additional visual classes.

Java archive file format (JAR)

-- Packages applet components for fast downloading.

-- Can incorporate digital signatures to

protect against intentional third-party

corruption of JAR file components.

-- Based on the PKZip format.

Java Database Connectivity (JDBC)

-- Database connectivity specification

similar to ODBC.

-- JDK includes JDBC-ODBC bridge for

building test programs.

Security API

-- Classes that support digital signatures,

data encryption, and key management.

-- Security support in JAR files.

Internationalization

-- Objects can be sensitive to language,

currency, and so forth.

-- Enhanced string class can collate based

on language.

-- TextBoundary class can determine

word and sentence boundaries based

on language.

Extended Network API

-- Sockets are no longer final and can

easily be extended.

-- Exceptions are more accurate.

Remote method invocation (RMI)

and object serialization

-- With RMI, an object can call another

object's methods remotely--on a machine

next door or in the next country.

-- The object-serialization API can convert

an object into a stream of bytes for

``beaming'' a copy of that object across the network.

Reflection API

-- Java applications can gather information

about loaded classes at run time.

-- Useful for debuggers; necessary for

JavaBeans.

Rick Grehan is a senior technical editor for BYTE reviews. You can reach him by sending e-mail to rick\_g@bix.com.

Illustration: Screen: It's not pretty, but it works: JDK 1.1 compiles our jBYTEmark suite without a hiccup.