

# Eclipse Becomes the Dominant Java IDE

David Geer

**S**ince Sun Microsystems introduced Java in 1995, proponents have sought ways to boost the technology's fortunes. One approach has been to create an integrated development environment that would make working with Java easier.

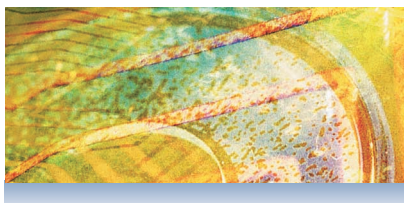
Supporters hoped an IDE would make Java more competitive with Microsoft's popular Visual Studio .NET, which provides an environment for integrated, easy-to-use software tools that appeal to the many business-application developers who aren't hard-core programmers.

This has set off a battle among several Java IDEs, including Borland's JBuilder, Microsoft's Visual J#, Oracle's JDeveloper, and Sun's NetBeans.

One contender has been Eclipse, which IBM developed and turned over in 2001 to the nonprofit Eclipse Foundation ([www.eclipse.org](http://www.eclipse.org)) to manage as an open-source platform.

In addition to providing an IDE, Eclipse automates numerous functions that developers would otherwise hand code, said Alan Zeichick, editor in chief of *SD Times*, a newspaper for software-development managers.

Eclipse has garnered so much support that many industry observers say it is now the key Java-tools player. Today, the Eclipse Foundation has 98 member companies, including most of the largest software vendors. The technology even



has its own annual conference, EclipseCon, which sold out this year.

"Eclipse has truly won," said Zeichick. It is inexpensive to use and makes it much easier to integrate their tools with one another, he added.

## ECLIPSE HISTORY

Object Technology International developed the Java-based technology behind Eclipse before IBM bought the company in 1996. IBM began working on Eclipse internally in 1998 to integrate its many development programs.

IBM designed the Eclipse platform in accordance with standards set by the Object Management Group ([www.omg.org](http://www.omg.org)), which produces and maintains specifications for interoperable enterprise applications.

Although the Eclipse Foundation now manages the platform, nonmembers can also build applications using the technology.

## HOW ECLIPSE WORKS

Multivendor IDEs are a key factor in software design. They let a project's developers select their preferred tools

from different vendors without worrying about making them work together or learning multiple interfaces and programming environments.

Like other IDEs, Eclipse is a programming environment packaged as an application. It consists of a code editor, compiler, debugger, GUI builder, and other tools.

For example, the Eclipse Foundation has included refactoring tools, which conduct a series of small transformations to restructure an existing body of code—for example, to make it smaller and less buggy—without changing its external behavior, noted Ian Skerrett, the organization's director of marketing.

The foundation has also added intelligence to the text editor, which is used for hand coding, Skerrett added.

Eclipse offers a set of APIs that connect tools into one unit, the Generic Workbench, that works as a single development environment with one set of behaviors and interfaces.

Eclipse uses the Standard Widget Toolkit to provide programs' interfaces. The IBM-created SWT is a class library for creating GUIs in Java. It lets developers build portable applications that directly access the user-interface facilities of the operating systems on which they are implemented. The Java programs thus look like native desktop applications.

Proponents say that because the SWT works with the operating system, it will perform better than techniques that bring their own UI features and thereby create user interfaces that look the same regardless of the host OS.

Meanwhile, Eclipse automates functions, such as the creation of buttons and dialog boxes, that developers might otherwise have to hand code.

Eclipse is built with Java and thus runs on multiple platforms. However, it will also help build applications in other languages such as C, C++, Cobol, and HTML.

## REASONS FOR ECLIPSE'S SUCCESS

"Among the top Java IDEs, Eclipse is the only one gaining market share in

Europe, the Middle East, Africa, Asia-Pacific, and North America,” said analyst Albion Butters with Evans Data, a market research firm.

“The power of Eclipse is the common platform that you can integrate different tools into,” said the Eclipse Foundation’s Skerrett.

Eclipse was created as a platform for plug-in tools that extend the IDE’s capabilities so that it can work with numerous programming languages and applications, as Figure 1 shows. Anyone can write plug-ins for Eclipse and have them work directly with any other plug-ins for the platform. Some other IDEs limit plug-in creation to company partners.

There is thus a “huge number” of interoperable third-party plug-ins, which has made Eclipse very popular, said John Andrews, Evans Data’s chief operating officer.

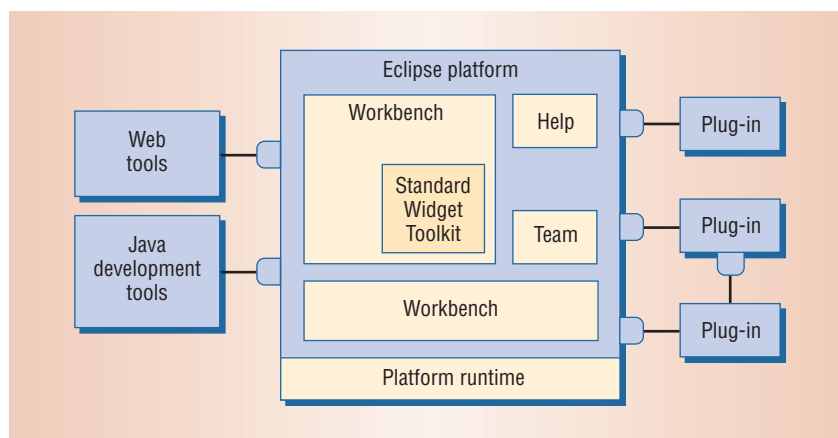
IBM’s release of Eclipse to the Eclipse Foundation made the technology independent of any company, which fueled its broader adoption by businesses that don’t want to be tied to a specific vendor, noted Rob Cheng, Borland’s director of product marketing. “The more independent Eclipse is, the more comfortable companies and developers feel using it,” he explained.

For example, Oracle is working to ensure that any developer using Eclipse can build applications for the vendor’s application server and database, explained Ted Farrell, chief architect in the company’s Application Development Tools Division.

### Lower costs

The entire Eclipse development platform is free. Proprietary IDE systems such as JBuilder, JDeveloper, and JetBrains’ IntelliJ IDEA, on the other hand, can cost up to \$3,500 each.

Users seeking to add plug-ins that aren’t part of Eclipse can get some tools for free and pay for others. Either way, it can be less expensive than buying an entire proprietary development platform.



**Figure 1. Eclipse was created as a platform for plug-in tools that extend its capabilities so that it can work with numerous programming languages and applications. The tools plugged into the platform operate on regular files in the user’s workspace. Eclipse can place a project in the workspace under version and configuration management with an associated team repository. On startup, the platform runtime discovers the set of available plug-ins, reads their manifest files, and builds an in-memory plug-in registry. The workbench provides Eclipse’s user-interface personality and includes the SWT general-purpose UI toolkit.**

Eclipse’s popularity has led many tool developers to make their products compatible with it. This competition has reduced the price of the plug-ins that aren’t free, according to Evans Data’s Andrews.

### Fast-moving innovation and development

Because Eclipse is open source, Borland’s Cheng said, developers have ready access to the source code and can modify it and innovate quickly to meet users’ needs.

And, Cheng added, companies like the technology’s open development process. “It is a very transparent process. Most of the communications, milestones and plans are public, and the builds are available for public download. Interim builds come out every couple weeks or every month so that people can try it out and give feedback quickly. There is a lot of community involvement,” he said.

### Elegant architecture

According to Cheng, Eclipse is a small, modular IDE with an elegant architecture that starts from a basic but

powerful foundation. “There is a layer that lets you integrate applications without worrying about drawing dialog boxes, buttons, and widgets or property pages and project trees,” he explained.

Thus, he elaborated, developers can hand code the new elements they need or want and disregard the elements that stay the same from program to program, such as dialog boxes.

### NOT A TOTAL ECLIPSE

Although it appears to be the Java IDE of choice, Eclipse still faces competition from alternatives such as JBuilder, Visual J#, JDeveloper, and NetBeans.

“Eclipse is certainly a very popular IDE and very successful,” said Tim Cramer, software engineering director for Sun’s NetBeans. “However, NetBeans is seeing a resurgence with the developer community. Eclipse has been great competition, and because of this, we’re all going to improve.”

“With NetBeans 4.1,” he added, “we now have a number of features that add value above and beyond what Eclipse might have: visual development of



# REACH HIGHER

Advancing in the IEEE  
Computer Society can  
elevate your standing in  
the profession.

Application to Senior-  
grade membership  
recognizes

- ✓ ten years or more  
of professional  
expertise

Nomination to Fellow-  
grade membership  
recognizes

- ✓ exemplary  
accomplishments  
in computer  
engineering

GIVE YOUR CAREER A BOOST

UPGRADE YOUR MEMBERSHIP

[www.computer.org/  
join/grades.htm](http://www.computer.org/join/grades.htm)

## Industry Trends

J2ME (Java 2 Platform, Micro Edition) applications, debugging on a live cell phone, and an advanced profiler.”

Thus, he said, “Our active users have gone up dramatically according to our internal measures, and we’re also seeing a surge in traffic to our Web site.”

### Concerns

According to Cramer, because Eclipse and the SWT are not going through Sun’s Java Community Process for introducing new features, they don’t create applications with true Java functionality.

In addition, he explained, developers must port the SWT to all platforms on which Eclipse runs, which can be complex, time consuming, and expensive. NetBeans, on the other hand, runs natively anywhere there’s a 1.4 or later version of Java, he noted.

NetBeans originally used the Abstract Window Toolkit, an API for Java-application GUI development. Sun discovered performance and extensibility limitations with AWT and thus developed Swing, explained Cramer.

AWT uses the operating system’s graphics code for GUIs while Swing brings its own, creating GUIs that look the same on any OS. In addition, Cramer said, Swing has about 500 classes of GUI-related objects and thus offers richer graphics and more components than AWT, which has only about 50 classes.

Because AWT and Swing are part of the Java specification, proponents say, they offer better Java functionality.

### Oracle’s compromise

Concerned that Eclipse and NetBeans might create incompatible technologies that would split Java and make it less attractive to developers, Oracle has offered a compromise designed to enable compatibility.

The company has submitted Java Specification Request 198, “A Standard Extension API for Integrated Development Environments,” to Sun’s JCP. Rather than introduce yet another IDE, JSR 198 would provide a stan-

dard API that would work with all Java IDEs that support it.

**T**he Eclipse Foundation has submitted for review Eclipse 3.1 Release Candidate 1, which features an updated SWT that offers more capabilities and interoperability with a greater number of browsers. The new version would also be faster, include more wizards, and enable automatic coding of additional features.

In addition, the foundation is expanding its activities. For example, the group’s Web Tools Platform Project plans to begin releasing tools this summer. The organization has also developed business intelligence and reporting tools for generating reports from Java servers and is working on a rich-client platform for developing robust desktop and workstation applications.

According to *SD Times*’ Zeichick, Eclipse will be the leading Java IDE for at least five years because of vendor support.

However, Cheng noted, it remains to be seen how sophisticated Eclipse’s functionality and features will get. “It’s not clear where different groups within Eclipse will move and evolve with their projects. It may be that Java development will only reach a certain level on Eclipse,” he explained.

Said Oracle’s Farrell, “Eclipse’s success is tied to how good a product it is. If it starts to deviate from the main development base, it will begin to lose favor. Now that Eclipse is expanding, there are a lot more people contributing different types of technologies to it. As the base starts to grow, there is a danger of it losing some of its appeal as being lightweight, fast, and focused on the developer.” ■

*David Geer is a freelance technology writer based in Ashtabula, Ohio. Contact him at [david@geercom.com](mailto:david@geercom.com).*

Editor: Lee Garber, *Computer*,  
[l.garber@computer.org](mailto:l.garber@computer.org)