Eclipse: A Platform Becomes an Open-Source Woodstock

Alexander Wolfe, Science Writer

Call it a platform. Call it a tool. Call it the hottest open-source software movement since Linux. All those descriptions currently surround Eclipse,¹ the language-agnostic code base that's billed as a "universal platform."

According to its creators—who are developing a very un-Microsoft-like knack for issuing bland pronouncements that downplay the growing industry buzz surrounding Eclipse—it's more precisely a metaplatform. That is, it's a building tool out of which developers can create IDEs (integrated development environments), or indeed anything else they might desire.

"Eclipse is common plumbing that everyone can build on,"² says John Wiegand, Eclipse platform lead. "We think of it as a platform for tool integration."

In practice, users coming upon Eclipse for the first time will tend to think of it as a Java IDE, since that's the most mature implementation of the technology. (People think of Eclipse as a Java IDE because that's the download that's easiest to obtain, opines Wiegand.)

Equally important to its success is that users quickly notice Eclipse is free—a factor that has become a big draw as comparable IDEs regularly sell for at least \$1,000. That's probably the way IBM planned things when it decided to seed the platform in the software community.

As work on the software commenced in 1999, IBM began to dole out some \$40 million in funding. IBM also spun off Eclipse as a stand-alone operation and offered up key personnel to staff the project. For one, platform lead Wiegand is officially an IBM employee, but he works on the Eclipse project full time. (Initially, 40 people were assigned to Eclipse; today, there are developer teams in Ottawa; Minneapolis; Research Triangle Park, N.C.; Zurich; Toronto; and St. Nazeaire, France.)

The first fruits of the project, Eclipse 1.0, premiered in October 2001. The buzz reached major proportions when version 2.1 shipped in 2003. The Eclipse software development kit for 2.1 includes a full-featured Java IDE upgraded with significant improvements to editor and debugger. The SDK (Software Developer's Kit) provides editors for Java files and for plug-in XML files. The 2.1 release also marked the debut of a Mac OS implementation

This free metaplatform IS GOING

TO CHANGE THE WAY WE
WRITE CODE.

of Eclipse. A multiplatform player, Eclipse is available in builds that run on, among others, Windows 98 and XP, the Mac OS, IBM's

AIX Unix, Sun's Solaris 8, Hewlett-Packard's HP/UX, and the Red Hat and SuSE versions of Linux.

With 2.1, the Eclipse organization announced the addition of five new corporate members, including mobile-phone powerhouse Ericsson, bringing the total number to 34. High-profile members include Fujitsu, Hitachi, database vendor Sybase, embedded software houses QNX and MontaVista, Hewlett-Packard, and enterprise code provider SAP, among others.

Now, work is well underway on the 3.0 upgrade, which is expected early next year (beta builds are already rolling out). "We're pushing on to things like scalability—how to handle hundreds of plug-ins—and improved responsiveness," says Wiegand.

Eclipse is also beginning to broaden out from its Java roots toward the widely used C++ language. In 2002, an Eclipse subgroup called the C/C++ development team (CDT) announced³ the availability of CDT 1.0. The release is a basic C/C++ development environment—complete with a debugger and GNU tools and compilers—that the team says is a big step toward its ultimate goal of a fully functional cross-platform C/C++ IDE.

Despite its IDE-centric associations, Wiegand is adamant about wanting to make Eclipse attractive to developers who aren't necessarily interested in an IDE. Rather, he likes to position Eclipse as a language-neutral base with a virtual machine on the bottom, upon which anyone can add software components. "There are many organizations building commercial projects on top of Eclipse, and there are more in the queue," he says. IBM's Webster Studio is the biggest example of an Eclipse-based commercial product. But the list of currently available plug-ins also includes Timestorm from Timesys, Borland's Together Edition for Websphere Studio, Interstage from Fujitsu, and CodePro Studio from Instantiations.

Eclipse features a slick user interface, which may explain its growing appeal to commercial users as a base

Eclipse Overview

FIG 1

upon which to field plug-ins. It also has what's widely acknowledged to be a powerful and well-documented back end, an additional impetus for development.

The commercial world aside, plug-ins seem to be the key reason Eclipse is catching fire with independent developers. Under the terms of the CPL (Common Public License),⁴ under which it's released, developers are allowed to charge for any third-party extensions and plug-ins they create. To ease that task, Eclipse comes equipped with a built-in development environment for creating plug-ins. Here, Eclipse applies its Standard Widget Toolkit (SWT), an applications programming interface that's tightly integrated with the platform's underlying windowing system and user interface.

Small shops and individual programmers have taken the bait and are already packing the pipeline with plug-ins; for example, the homebrew site <http://eclipse-plugins.2y.net> lists nearly 400 new offerings.

As the IDE and numerous plug-ins continue to propel Eclipse in the software community, the project leadership has additional plans to broaden the platform's reach. Leading that list is an effort to create a full-blown applications platform. Wiegand characterizes the goal as a "rich client platform." Wiegand explains: "We want to take a subset of the Eclipse platform and make it available to build apps."

Asked if this would be akin to a visual programming environment, Wiegand says, no, "we're talking about an actual platform." Since Eclipse itself runs on top of Windows, Unix, and other basic operating systems, what

Wiegand envisions can be considered to be a platform shell. Nevertheless, it would be very useful—especially in the enterprise space—for mid-level developers trying to build end-user applications.

While Eclipse hasn't come anywhere near hitting the peak of its popularity yet, there have been some criticisms. Chief among them has been the charge of "featuritis." As a recent posting on eclipse.technology,⁵ one of more than a dozen newsgroups devoted to the platform, put it: "Why is it

that Eclipse and so many other IDEs try so hard to make sure that the developer can not have simplicity in their development environment?"

And it's sometimes vexing, especially given the growing popularity of the effort, that the public faces of the Eclipse organization seem so intent on keeping its public pronouncements low-key and rather bland. On the other hand, perhaps such reticence, which harkens back to the days of the old-line Unix heads, is preferable to the hyperbole and vaporware promises we've become used to.

There's been less restraint on TheServerSide.com, an influential blog that tracks the Java developer community. "Keep a close eye on Eclipse, it's going to change the way we write code,"⁶ wrote software developer Cedric Beust in a posting earlier this year.

Perhaps Beust put it best when he tried to explain why Eclipse has gained so much software mindshare, yet can sometimes seem dull when discussed in strictly objective terms. "It's hard to communicate why Eclipse feels so good to use, because a lot is due to little details that manifest themselves to you as you use the tool," he wrote. ☐

REFERENCES

1. Eclipse: see <http://www.eclipse.org>.
2. Wiegand, J. Personal Interview (Aug. 20, 2003).
3. Eclipse. Eclipse.org delivers C/C++ development tools (Jan. 29, 2002); <http://www.eclipse.org/org/press-release/nr.redhat.linux.htm>.
4. The Open Source Initiative. CPL Version 1.0; <http://www.opensource.org/licenses/cpl.php>.

5. Eclipse Newsgroup (Sept. 2003 archive); <http://news.eclipse.org/newsgroups/>.
6. Beust, C. Serverside at JavaOne. The ServerSide Blog (sponsored by Sun Microsystems and the Middleware Co.); <http://www.theserverside.com/>.

ALEXANDER WOLFE received his electrical engineering degree from Cooper Union in New York City. A science writer based in Forest Hills, New York, he has contributed to *IEEE Spectrum*, *EE Times*, *Embedded Systems Programming*, and *Byte.com*.

©2003 ACM 1542-7730/03/

RESOURCES

INTERNET

Eclipse Web Site

<http://www.eclipse.org>

Eclipse Newsgroups

<http://www.eclipse.org/newsgroups/index.html>

(password required; available via free signup.)

Java Software Developer's Blog

<http://www.theserverside.com/>

Sponsored by Sun Microsystems and the Middleware Co.

Eclipse Plug-Ins

<http://www.eclipse.org/community/>

[plugins.html](#)

<http://eclipse-plugins.2y.net>

BOOKS

The Java Developer's Guide to Eclipse

Sherry Shavor, Jim D'Anjou, Scott Fairbrother, Dan Kehn, John Kellerman, and Pat McCarthy (Addison-Wesley, 2003)

Eclipse in Action: A Guide for Java Developers

David Gallardo, Ed Burnette, and Robert McGovern (Manning, 2003)

Eclipse: Step by Step

Joe Pluta (MC Press, 2003)

acm
queue
tomorrow's computing today

Surviving a Distributed Development Environment

- Adding Collaboration Tools into IDEs
- Are Cultural Differences Slowing Your Team?

Also Next Month:

- Gordon Bell on Dysfunctional Start-ups
- Debugging Black Box Binaries

Coming in December/January