

## Beware the March of This IDE:

### Eclipse Is Overshadowing Other Tool Technologies

**Greg Goth**

**T**he open source Eclipse Project is rapidly obtaining majority market share among development platforms, according to analysts' surveys as well as adoption rates by major vendors.

The project, originally conceived in the late 1990s by a small group of IBM engineers as a way to standardize development platforms across the company, now includes 97 members. It has expanded well beyond its original core technology of a Java integrated development environment (IDE) to include approved projects or proposals for data tool development, embedded-device tool development, and rich-client application development, among others. (For more on Eclipse and related projects, see the "Related URLs" sidebar.)

#### **Solid growth**

According to industry surveys, well over half the developers contacted prefer Eclipse as their primary Java IDE. Two Evans Data reports from 2004 showed rapid adoption of Eclipse. From February 2003 to February 2004, Eclipse use among Linux developers grew by 80 percent, and Eclipse became the operating system's most-used IDE. A follow-up Evans survey of over 400 Java developers in May 2004 concluded that Eclipse was the only one of the top three Java IDEs gaining market share.

The trend has continued in 2005. A survey of 515 attendees at the March 2005 Java Symposium revealed that 53 percent of Java developers preferred Eclipse as their primary IDE. The next

most popular IDE, JetBrains' IntelliJ IDEA, received 19 percent of the vote. The Eclipse Foundation, which oversees the Eclipse Project, estimates that Eclipse has gained a huge share of the tools platform market—at least 75 percent—for the most advanced aspects of its technology, such as corporate Java development.

The software industry has recognized Eclipse's grassroots popularity by lending significant corporate resources to it. In late February and early March 2005, industry heavyweights BEA, Borland, Computer Associates, Wind River, and Sybase agreed to participate as strategic developers. Vendors participating at that level must shoulder annual dues of US\$250,000 and commit at least eight developers to leading an Eclipse project.

Mike Milinkovich, the year-old Eclipse Foundation's executive director, says these recent statistics and announcements should make it clear that Eclipse has completely shed its IBM cocoon and is fully independent.

"It's kind of like riding a tiger," Milinkovich says. "The growth curve has been very much like a hockey stick at the strategic level."

Lee Nackman, vice president of product development at IBM's Rational Software, is one of the pioneers behind Eclipse. He notes that while 2005 might mark the unofficial coronation on the corporate level, developers were swarming to it long before.

"The thing that really drove it home for me was when we released version 2.0," Nackman says. "It was released on a Friday late in the

day, and the servers were overwhelmed, which told me there were tons of people who were eagerly watching and waiting.”

**Extensibility + practicality = popularity**

The Eclipse platform’s core of functionality is the ability to accommodate tools developed as plug-in modules. Although Eclipse is written in Java, it supports any language. Typically, a developer can write a given tool as a separate plug-in that operates on files in the platform’s workspace.

The platform’s workbench displays the plug-in’s tool-specific user interface (see the figure). Additionally, each plug-in may be extended by other plug-ins. These additional plug-ins may be either open source or proprietary tools for which developers pay a licensing fee.

Vendors see the two-sided “value add” capability Eclipse brings to the industry. They don’t have to rewrite fundamental development platform code, yet Eclipse is compatible with proprietary plug-ins with which they try to distinguish themselves from competitors. While unable to reveal specific amounts, IBM’s Nackman claims the company has saved “many millions of dollars” by adopting Eclipse.

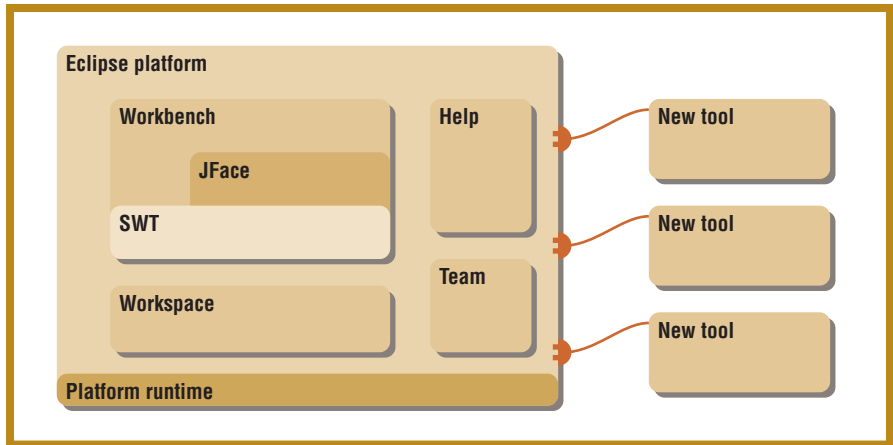
Tim Wagner, senior manager of open source compiler and IDE tools at BEA, says, “Probably half of the investment we were making in Workshop [BEA’s own extensible Java development environment] was what I would call infrastructure. It was redundant with things we could simply take for granted with Eclipse. When we look at the ubiquity of the platform, if we get even a 5 to 10 percent boost in penetration, it will pay for our strategic developer investment in Eclipse.”

One of the newest top-level Eclipse projects is the device software development project, proposed by Wind River. Rob McCammon, director of product management for the Wind River Development Suite, says the need for Eclipse in the embedded and device software market might be even more pronounced than that for enterprise tool design.

“In the device software space, histor-

**Related URLs**

- Device Software Development Platform:** [www.eclipse.org/proposals/eclipse-dsdp/index.html](http://www.eclipse.org/proposals/eclipse-dsdp/index.html)
- EclipseCon 2005:** [www.eclipsecon.org/2005/index.php](http://www.eclipsecon.org/2005/index.php)
- Eclipse Foundation:** [www.eclipse.org/projects/index.html](http://www.eclipse.org/projects/index.html)
- Eclipse Projects and Proposals:** [www.eclipse.org/projects/index.html](http://www.eclipse.org/projects/index.html)
- Eclipse Platform Technical Overview:** [www.eclipse.org/whitepapers/eclipse-overview.pdf](http://www.eclipse.org/whitepapers/eclipse-overview.pdf)
- JetBrains Open Source License:** [www.jetbrains.com/idea/opensource/application.htm](http://www.jetbrains.com/idea/opensource/application.htm)
- An Open Letter to Eclipse Membership from Sun:** [www.eclipseplugincentral.com/displayarticle8.html](http://www.eclipseplugincentral.com/displayarticle8.html)
- Philippe Mouglin looks at Eclipse:** [www.macdevcenter.com/pub/a/mac/2003/11/14/eclipse.html](http://www.macdevcenter.com/pub/a/mac/2003/11/14/eclipse.html)
- The VSTSEclipse project:** [www.vstseclipse.org](http://www.vstseclipse.org)



**The Eclipse platform. JFace and SWT (Standard Widget Toolkit) are the Eclipse GUI components.**

ically there have been many companies providing development tool capabilities in a proprietary way,” McCammon says. The reason, he says, is that the device market is fragmented across many technologies. While the desktop and server environments are fairly standard, the device landscape includes everything from digital cameras to antilock brake systems to giant racks of telecommunications equipment. A successful Eclipse device tool platform, according to McCammon, will let vendors concentrate on their own contributions. The Eclipse-supplied building blocks will furnish the components that don’t need to differ from one project to the next.

The device project is just one of 15

proposals that Eclipse is investigating; six top-level projects, including the Eclipse parent project, are under way. In addition to tools projects, Eclipse features a Rich Client Platform for application development, available since June 2004. IBM’s Workplace Client Technology, which lets enterprise end users access editing tools via a client-server architecture, is written on the RCP.

Both Nackman and Milinkovich say one of the organization’s greatest challenges will be to reconcile the need to nurture the rapid growth of various Eclipse projects with the need to release new technology at reasonable intervals.

“On one hand you want to let all the flowers bloom,” Nackman says. “On

## What Developers Say

As Eclipse has grown into ever more complex applications, developers have been concerned about its scalability. Philippe Mouglin, creator of the F-Script scripting language for Cocoa, wrote in a 2003 O'Reilly report that "for Eclipse to scale well on big projects, some special attention and fine-tuning is required from the user. Things like customization of the graphical environment, definition of subprojects, organization of source code and resources, choice and customization of plug-ins have to be well thought. Without such fine-tuning, the Eclipse environment can slow down significantly when the code base grows."

Such concerns persist. Australia-based developer Joe Sango, who has adopted Eclipse, says, "The relative ease of plug-in development is a definite plus, but saying that, depending on the complexity of your solution, developing an enterprise plug-in for Eclipse can be quite daunting."

Sango says such concerns haven't derailed Eclipse's popularity among developers he knows. "Most Java developers that I have come across, including myself, either use Eclipse or are in the process of switching. In my opinion, it is one of the easiest and most intuitive open source IDE's currently out there and the framework developed makes it easily extensible."

Martial Dore, chief architect of Paris-based Orchestra Networks, who collaborated with Mouglin on his report, still uses the platform regularly and is impressed with its progress. Dore says he still has some "little disappointments" with Eclipse. For example, search functionality hasn't improved, and some configuration files are still not fully XML-ized. However, his global impression is that the latest version, Eclipse 3.0, "is the cathedral of the 21st century! Foundations are stable, it is able to elevate very high but is still widely open, [and] many areas are (and will be) progressively enriched."

Yet Luke Hutteman, developer of the RSS aggregator SharpReader, says that Eclipse's rapid growth shouldn't lead you to count other IDEs out. (RSS stands for Really Simple Syndication, a Web content syndication format.) Hutteman prefers IntelliJ IDEA, which he classifies as more intuitive and feature-rich than Eclipse.

"In theory, I do like Eclipse for being open source and free, and the fact that it's therefore well supported by a growing number of companies and developers," Hutteman says. "In practice, though, I still keep coming back to IDEA. On the other hand, Eclipse is a pretty good product that has borrowed a lot of its functionality from IDEA and that has been gaining on IDEA in recent years. If at some point they eclipse IDEA (forgive the pun), I will have no problem switching and saving myself the upgrade fees. I don't see this happening any time soon, though."

"We have to balance between allowing the technology to evolve at its natural pace without causing breakage and instability. That will be a problem," says Nackman.

(For more opinions on Eclipse's growth and scalability, see the "What Developers Say" sidebar.)

## Well-defined process, unique incubation model

To help coordinate development, the Eclipse Foundation recently hired veteran developer Bjorn Freeman-Benson as technical director to coordinate each project's progress. In addition, Milinkovich says, just because Eclipse is open source, it shouldn't be construed as an anything-goes environment. In fact, a Project Management Committee manages Eclipse's development. In addition, councils for requirements, planning, and architecture oversee their respective areas, make progress reports to the PMC, and ensure timely release of new technology.

BEA's Wagner, a member of the planning and architecture councils, says the organization must remain cognizant of its external image and internal reality. "Facing out to customers, I think the challenge is to make Eclipse the easy, integrated, 'feels like it came from a single vendor' solution we all want it to be," he says. "Inside Eclipse, it's a slightly different question. It's how can we manage what is becoming a very large, diverse software artifact built by people all over the globe and keep it orderly and under control?"

The consensus holds that the key element behind Eclipse's recent rapid growth has been the acceptance of the Eclipse Foundation as an entity entirely separate from IBM control. Although IBM had decided to make Eclipse open source and ran it as a consortium from late 2001 until the foundation was announced, Nackman says even the loose corporate ties were enough to keep other vendors on the sidelines.

Michael Azoff, an industry analyst with the UK-based Butler Group, says the new Eclipse organizational structure reminds him of the collaborative underpinnings common in basic academic

the other hand, you could have overlap, duplication, and conflict."

According to Nackman, such duplication has already happened. At EclipseCon 2005, the organization's annual conference, a new Sybase-led data tools project was announced. Nackman says some of the technology proposed for this project already existed in the Eclipse Web tools project. Those two projects are now coordinating, and some Web tools technology will be

moved to the data tools project so that it needs to be developed only once, says Nackman. He concedes that rapid growth might lead to other such redundant code, which must be watched for constantly.

A similar challenge, Nackman says, will be balancing overall platform stability and growth. As more and more people become dependent on Eclipse, pressure will increase for slowing down the rate of change to the platform.

and scientific research. Basic research in those communities, he says, is pooled, and intellectual property is culled from that common base. Likewise, with the Eclipse Project and Eclipse Foundation, vendors can mutually benefit from having basic tools to build other tools. They can then build on that to provide premium products that have a license fee.

Yet, while the industry needed to wait for IBM to fully release Eclipse into the community for it to reach its full potential, that potential might have been delayed for years or perhaps never been realized, if IBM had employed the traditional open source development model. Instead, IBM released a platform that was already capable of turning heads because it had been coded by a small group with a clear mission from Day One.

“We took technology that we started to develop in a conventional way and used that to seed an open source project that has become a traditional open source project, if there is such a thing,” Nackman says. “There was enough there to attract interest and build excitement. This mixture of the open source way of doing things and commercial vendors is really quite neat.”

**So what’s next?**

With Eclipse’s popularity booming, strategists, developers, and analysts are exploring and debating its place in the larger development ecosystem. Among major software vendors, just two, Sun Microsystems and Microsoft, have remained aloof from Eclipse—but Microsoft has sent representatives to both EclipseCon annual conferences.

“They sent six people to EclipseCon this year and sent at least that many people to the first conference in 2004,” Milinkovich says. “We haven’t had any conversations about them joining Eclipse, but I think there are some areas where it would be good for them, as well as for us, to cooperate.” Two areas ripe for collaboration, he says, include the whole area of Web Services interoperability, as well as an Eclipse project dealing with C# and .NET tools.

Unofficial Microsoft/Eclipse proj-

ects are already under way. Melbourne, Australia-based developer Joe Sango and several of his colleagues are beginning the VSTSEclipse project. Its goal is to create a suite of Eclipse plug-ins with which developers can leverage the source control and work item features in Microsoft’s new Visual Studio 2005 Team System, which was released in beta in April 2005. The Team System lets multiple programmers working on a single project coordinate their source code contributions and find bugs. Sango hopes their project will allow such coordination in projects containing code written in both .NET and alternative formats.

It looks far less likely that Sun will meld its NetBeans Java development technology with Eclipse (Sun didn’t respond to several requests for comments). Sun sent Eclipse an open letter upon the establishment of the Eclipse Foundation, pledging mutual support but not offering to join the organization. However, Bill Weinberg, open source architecture specialist for the Open Source Development Labs, says cross-pollination of Sun and Eclipse technology is a given whether Sun joins the open source project or not. Because Solaris and Sun’s Java tools are so ubiquitous, Weinberg contends that Eclipse will eventually show up in their environment.

Despite industry gossip, Nackman says the name Eclipse has nothing to

do with one-upping Sun on a Java platform. Instead, he says IBM’s Eclipse team was searching for a name that began with a long “e” sound to go with IBM’s “e-business” ad campaigns. If the team wanted to eclipse any technology, he says, it was Microsoft’s Visual Studio .NET.

Among dedicated Java IDE vendors, JetBrains executives say they have no plans to join Eclipse. “In coding productivity features and as a professional code-oriented developer tool, IDEA is the only product that stands in competition with Eclipse,” the company said in a written statement. “All the other companies threw their weight behind Eclipse to try and improve their offerings to effectively compete with Eclipse and IDEA.”

JetBrains recently began its own open source initiative, giving free licenses for IDEA to any qualified non-commercial open source project. By late May 2005, developers in 150 projects, including 60 percent of Apache Software Foundation Java development projects, had signed up for the program.

JetBrains is also confident the features that attract developers to the company’s technology will continue to provide a loyal customer base.

Even Eclipse executive director Milinkovich admits IntelliJ is a formidable piece of technology.

“With IntelliJ, they clearly have a great product. I wish there was a way for them to come work at Eclipse. Everything I’ve ever read or seen from those guys tells me they’re one smart bunch of people.”

However, Milinkovich also says the original Eclipse platform’s popularity places pressure on the organization to look past its success as an IDE platform and its competition in that space.

“When I think of how would I measure my success a couple years from now, I’d like to see some additional projects that are as successful in their space as the Eclipse Platform Project has been in the Java IDE space. How you go about redoing that magic is the real challenge, I think.”

**“In coding productivity features and as a professional code-oriented developer tool, IDEA is the only product that stands in competition with Eclipse.”**

# Software Boot Camps Are In, and Focused on Security

**Bernard Cole**

**D**emand for software developers and IT professionals with up-to-date training is increasing. And so are certification schools and training courses—the more intense, focused, and security oriented, the better.

A Google search bears this out. The keywords “certification boot camp” yield 240,000 hits. A nested search on those results for “security” turns up 106,000 hits. According to Barry Kaufman, CTO and cofounder of Intense School ([www.intenseschool.com](http://www.intenseschool.com)), such results reflect accurately the demand from the individuals and corporations that come to his institution for training.

## That secure feeling

“With the computerization of every aspect of our lives, there has always been a demand for the continuing training of computer professionals in the fundamentals and for various certifications,” Kaufman says. “But with virtually every type of computer system now networked, there is almost as much, if not more, emphasis on training in all aspects of computer and network security.”

Even when security issues aren’t being taught directly, they permeate general software training, according to Ralph Echemendia, lead instructor at the Intense School. “One of the messages I try to leave with my students in even the most general of certification courses is that if you understand every aspect of the system for which you are responsible, the more likely you will be able to sense when there will be problems and anticipate them,” he says.

Echemendia believes the students coming to the classes share this attitude. “Before 9/11, the attitude of many schools and their students was short term, focused mainly on certification and the short-term cramming needed to pass the tests. Now the attitude is ‘Hey, this is serious business, and I’d better learn and remember it for the rest of my life.’”

## The power of concentrated training

Students, and the corporations that sponsor them, says Kaufman, like the week-long “boot camp” type of training because they think it reinforces the learning process and leads to better retention.

“A lot depends on student motivation. If they come to a course intent on learning and not just cramming, then that is what will happen,” says Echemendia. “But there is a lot a boot camp environment can do to guarantee that the student is learning the right things and doing so in a way that facilitates long-term retention.”

## Clément Dupuis’ Software Boot Camp Checklist

- How many years has the school been doing this type of training?
- How many years of experience do the instructors have?
- Are the instructors recognized in their specific teaching areas?
- How many times has the class been delivered?
- What are the courseware developers’ credentials?
- Is the offer all-inclusive (hotel, meals, and preclass and postclass assistance), or does it include just the course?

Sheer immersion in the subject matter, 10 to 14 hours a day for the seven days of a typical boot camp, ensures that a lot of what is studied sticks, Kaufman says. “But beyond well-structured course material, there has to be a well thought-out testing methodology, many times a day,” he adds. “Also, the testing needs to balance between certification needs and ensuring understanding of the general topic.”

In the view of Clément Dupuis, president of CCCure Enterprise Security & Training ([www.cccure.org](http://www.cccure.org)), the key to the success of boot camps such as Intense’s is the right mix of live instructor delivery, graphic support, video, reinforcement of key points, quizzes to evaluate how much students have retained, and further review of the material after the quizzes to clearly explain any unclear concepts. “These steps guarantee that the student has been exposed to the subject matter at least four to five times by the end of the class,” he says. “If you still cannot retain it then, you might need to think about another career.” (For advice from Dupuis on choosing a boot camp, see the sidebar.)

The best way to reinforce the formalized coursework, according to Kaufman, is hands-on lab work using equipment and software similar to the students’ work environment and dealing with simulated problems mirroring the real world. “Companies and professional societies need course structures that are quantifiable so that they can be sure that the students who are given certification are qualified,” says Kaufman. “Although it’s hard to apply metrics to them, the scenarios and hands-on lab work are extremely effective. It’s a shame more of this can’t be worked into the certification process.” 