Interview: The Eclipse code donation

IBM's Marc Erickson gives the details on Eclipse

Steve Brody (sbrody@us.ibm.com), developerWorks Open source projects zone editor, IBM

Summary: OTI marketing lead Marc Erickson gives developerWorks some of the details regarding the recent release of $40 million worth of tools to the Eclipse project by IBM as the open source code that will form the basis of IBM's next generation of WebSphere Studio products.

Date: 01 Nov 2001
Level: Introductory
Comments: 0 (Add comments)

Average rating (based on 51 votes)

On November 5, 2001, IBM announced its donation of $40 million worth of tools to the Eclipse project. Eclipse, a fully functional software development environment that is written in Java, and that runs on both Linux and Windows, is intended to solve many of the problems of tool interoperability faced by developers of conventional tools. IBM is handing the tools over to the Eclipse consortium, an independent group that will handle the further development of the code based on contributions by board members and individual developers. The consortium includes roughly 40 members who will manage the code intended to become the basis of IBM's next generation of WebSphere products.

The move is designed to capitalize on the same kind of momentum that has propelled projects like Apache forward, namely the open source movement and development model. It also represents IBM's competition for Microsoft's .NET program, which is extendable but still remains a proprietary platform. Also competing in the field is Sun's NetBeans technology.

As analysts from the Hurwitz Group concluded, the move is consistent with IBM's commitment to Linux and growing tradition of incorporating open source code into its product lines. "With its experience with the open source application server Apache, and the Linux operating system, it makes sense that IBM would now move to provide the developer community with an open source development platform. The challenge for IBM and the Eclipse organization will be to draw strong and broad tool-vendor support to advance the platform, and to demonstrate that it is truly an open platform that enables straightforward tool integration to make it worthwhile for organizations to adopt. In addition, Eclipse needs to capture and enlist the efforts of the developer community at large, to test and refine the platform and add their innovation.

In the following interview, Marc Erickson, marketing lead for IBM's OTI subsidiary (the developers of Eclipse) gives you more details on the history and future of Eclipse.

developerWorks: How did Eclipse come to be?

Marc Erickson: Well, look at the history for a minute. When IBM purchased the OTI (Object Technology International) subsidiary in 1996, OTI had done a lot of work on tool development and embedded systems, and had a technology written in Smalltalk called Envy that became the essence of the development model in VisualAge for Java 3.5. That was the initial code stream developed for IBM's advanced tools.

The next generation was developed for VisualAge Micro Edition. It featured some very advanced approaches to code development, including, for example, the possibility of collaboration without locking out the code repository. VAME also perfected cross-platform development so that a PC-based development tool could interoperate with an embedded device via a network.

We set out to bring these two streams together inside IBM so our development teams could focus on a single basic technology for all target environments.
developerWorks: Why was Eclipse developed?

Marc: The lack of interoperability among tools has been a major problem for developers -- getting things to work together properly can drain resources that could be better spent on project development. That primary problem and a desire to make the market more open for developers are the two needs that motivated the creation of Eclipse, which is also now the foundation of the next generation of IBM's WebSphere Studio products.

developerWorks: What makes Eclipse more than a collection of open source APIs and an IDE?

Marc: The Eclipse Platform is shipped as a completely functional development environment. It consists of a framework that is extendable by plug-ins (and it comes with sample plug-ins). The Platform includes a fully functional IDE written in Java. This makes Eclipse a lot more than a collection of APIs. And it's much easier to work with than .NET, for example, which you cannot extend unless you are one of the commercial partners.

The basic framework of Eclipse has many extension points that a plug-in can interact with. They allow plug-ins to interact with each other or with the Platform itself. Even though the Eclipse Platform is written in Java, and comes with a Java IDE, it's in no way limited to creating Java applications. It's already being used for creating many other types of content.

developerWorks: What is the basis of the technology?

Marc: The Eclipse Platform is based on the concept of a plug-in. Tools that "plug in" to the Eclipse Platform are provided with a set of common services that make tool-to-tool integration possible.

The first common services focus on the project model. This is rendered as a graphical view of all the resources (U/I, UML, business logic, HTML, Java, C, COBOL) that make up a project. As developers find, create, and modify resources using different tools, the resource management service shares the resource status (creation, deletion, and modification) with other tools.

Workplace synchronization keeps the views inside each of the tools consistent to the latest modification of the resource. And because developers work within teams, a Team programming model, exposed as an interface to the open standards-based CVS version control system, provides version control management.

You also have a remote debug interface so that tools can be built to test the application as it runs live on the target machine (such as a server, device, or rules engine).

The Eclipse Platform also exposes the extension points needed for constructing new services. Developers can get a total picture of all the details by reading the white paper at Eclipse.org (see the Resources later in this interview).

developerWorks: What will be the key benefit to Linux developers (tool integration, etc.)?

Marc: There are two primary benefits to the Linux developer. For one, we now have a fully mature IDE that can be deployed on Linux workstations. The technology is interoperable to the point where you can have someone running the IDE on Linux and someone else on Windows, and it won't pose any barriers to collaboration. The interoperability means an increase in efficiency. If a developer can do a build with the push of a button and incrementally enhance the code (instead of focusing on the mechanics of make files and command lines), development is a lot more efficient.

In addition, we are adopting the open source philosophy that is already at the core of the Linux community. There's a great deal of interest in this from Linux product support organizations involved in everything from mainframe environments to embedded systems. A lot of very innovative people out there have chosen to work on Linux, and now we have an opportunity for them to make use of these tools. That means we're reaching the best developers in academia and industry, as well as anywhere people cannot afford an expensive development environment.

Remember that the Eclipse Platform doesn't limit the developer to deploying applications on any particular target. In fact, a developer using a Linux workstation could easily build and deploy Windows-hosted applications.

developerWorks: What does IBM hope to get out of turning this software into an OS project?

Marc: IBM believes it benefits when there is an open marketplace for interoperable tools. We all benefit because the products deployed become more useful and more easily integrated into developers' projects no matter what approach they are taking. A development environment that is really open creates a much more vibrant marketplace where people aren't locked into a proprietary system and where we've leveled the playing field that IBM joins with others to compete on.
Since IBM is confident that its products can compete feature for feature, this is good news for us. When companies have already made a choice that locks them into a given set of tools, it's harder for any company to jump that barrier and get their foot in the door. In an open marketplace, products can compete based on merit and that's where IBM wants to be.

**developerWorks: How is Eclipse licensed?**

**Marc:** These issues are under review right now by the board members. The final decisions on licensing will continue to be up to the Eclipse.org board. Right now if you go to download the code you'll be asked to accept the CPL. The member companies have all reviewed the CPL and appear comfortable with it. Each board member is preparing a commercial product based on Eclipse, so they are looking at it from that perspective.

**developerWorks: The architecture of Eclipse has been described as "controversial"; why is this?**

**Marc:** It's controversial in that not everyone wants to take an open approach. Some existing open approaches have not been commercially accepted.

In terms of its technology, the Eclipse Platform is a very mature and proven object oriented project developed by a group of people who really understand the needs of code developers. In fact, they've been using the Eclipse Platform itself to develop Eclipse! The Platform is the culmination of more than a decade of work, and as people open the code they will get a chance to see everything. So far, the feedback we've been getting is very positive.

### Resources

- You can join the Eclipse community and download this new open source IDE at the [Eclipse.org](http://www.eclipse.org) site. There you'll also find the Eclipse overview white paper, a glossary of terms, and description of Eclipse projects, along with technical articles and newsgroups.

- Read the November 5, 2001, interview with IBM's Scott Handy on why Eclipse means more Linux applications. Scott Handy is director of Linux solutions marketing for the IBM Software Group.

- Read the November 5, 2001, press releases announcing IBM's donation of tools to the Eclipse community and IBM's new WebSphere Studio tools.


- Read Daniel Steinberg's article on Refactoring with Eclipse.

- Learn more about Working the Eclipse Platform.

- Browse more Open source resources on developerWorks.

### About the author

Steve Brody is a contract writer and editor for developerWorks. He has contributed to a wide variety of online news
resources including Wired News, CNet's News.com, CNN Interactive, LinuxWorld, JavaWorld, and SunWorld. Prior to developerWorks, Steve was a contract editor for IDG's SunWorld. He can be reached at sbrody@us.ibm.com.

Trademarks