PACKET

Simson Garfinkel TECH

Redmond's Java Jolt

Microsoft knew Java could threaten its monopoly ... so there was only one thing to do

Many developers seem to be questioning Microsoft's commitment to Sun's Java programming language. But having played around with the company's J++ development environment for more than a week, I know that Redmond is throwing its full support behind Java and Java developers. And in doing so, the company hopes to kill off that language's great promise, and great threat: Freedom from the Microsoft monopoly.

It's no surprise, then, that Redmond programmers are excited about Java. Compared with C++, Microsoft's current language of choice, Java is far friendlier. The errors that most commonly plague C++ are gone before they even have a chance to appear - tha nks to runtime garbage collection and the absence of pointers. Indeed, it's hard to find a computer programmer who, having coded for a few days according to the Gospel of Java, would willingly go back to C++.

Is it any wonder that Microsoft has promised to make Java run just as fast as C++? Microsoft's programmers surely want it to be their language of choice. It's a matter of efficiency: Programmers don't want to spend all their time chasing down memory le aks and bad pointer assignments. That's one of the reasons why Corel, the company that bought WordPerfect from Novell, is in the process of developing a new word processor, spreadsheet, and drawing program in a Java environment. Gates's geeks are certainl y no different.

It's hard to find a computer programmer who, having coded for a few days according to the Gospel of Java, would willingly go back to C++.

But to Sun, Java is much more than a spiffy new programming language. It's a religion, complete with evangelists. One tenet of the faith is that Java is a secure programming language - a rogue Java program can't wipe out all the files on your hard disk without your permission. Another tenet: Java programs can run on PCs, Macs, and Unix boxes; they can run inside Web browsers or as stand-alone applications, all without recompiling. That's an important credo if your company makes SPARC microprocessors in a world dominated by Intel-based PCs.

The problem with this doctrine? It demands that applets and applications be written purely in the Java programming language. But for many, such a wholesale conversion is both expensive and scary.

Consider Jim Flynn, general manager for @Work Technologies, a consulting firm with a Wall Street clientele. One of Flynn's largest clients has "a legacy system that's a couple of years old, bu ilt with Microsoft Visual C++ and making RPC [remote procedure calls] back to their mainframe," says Flynn.

Would you use J++?

Jump into the fray in Threads.

The latest post to Tech is "Bookmark file is better" by Chris Andersen (stranger)

Flynn's client wants the application to work on the company's intranet, so they want to write the whole thing in Java - why pay consultants to bang their heads against pointer and memory allocation errors? The only snag is that the client has already s pent a pile of money developing its current C++ infrastructure.

That's the beauty of Microsoft's J++ environment. Seamlessly built atop Microsoft's Developer Studio, J++ allows programmers to freely intermix Java with C++ and Visual Basic. J++ comes with a "resource wizard" which lets a programmer draw dia log boxes and graphical user interfaces and then have them compiled into Java code. And the whole thing hooks up with Developer Studio's class browser and help system, which lets you click on a function name and automatically jump to its source code or do cumentation.

PacketChat: Chat here.

Microsoft's Java compiler is the fastest I've ever seen, spinning its way through source code at a million lines a minute. And the debugger hooks into Microsoft's Internet Explorer, allowing you to debug an applet as it's running inside a Web browser.

Subscribe to PacketFlash, for Packet news.

If you are new to Java programming, the J++ application wizard will create your starter project, complete with a starter class file, an HTML file, and the basic code for animation and event handling. To experienced programmers, the wizard will feel like a gimmick, but it will save hours for beginners who are trying to figure their way around the system.

In other words, Microsoft is doing what it does best: Using its developer tools to leverage its operating system, using its operating system to leverage its developer tools, and all the while adding hundreds of niche features - each one designed to app eal to a slightly different part of the market.

In the process, Microsoft will diffuse the threat of Java, Sun, and Netscape. Applets that use Visual Basic simply won't run on Netscape Navigator. Java applications that make calls to C++ libraries stored in DLLs won't run on Macs or Unix workstations unless those DLLs are specifically ported - an expensive, time-consuming, and bug-producing process.

Sure, it's possible to write portable Java programs with J++. The genius of adding Java to Developer Studio is that it is now possible to write Java programs that are not portable. Instead of being a tool that frees people from the Microsoft monopoly, Java becomes just one more golden link in those Microsoft handcuffs.

Given the choice between J++ and any other Java environment that I've

seen, I would chose J++. Barring some moral reason for avoiding Microsoft software, you probably would too.

Send mail to Simson Garfinkel at simsong@hotwired.com

Illustration by Dave Plunkert

webmonkey SURF CENTRAL

Previously in Garfinkel ...