

***For Immediate Release***

For More PR Information Contact:

Jeff Elliott, Power PR

Phone (310) 787-1940

Fax (310) 787-1970

E-mail: [Press@PowerPR.com](mailto:Press@PowerPR.com)

**Breaking the Language Barrier:**

*Programming platforms J2EE and Visual Studio .NET are fighting it out for market dominance. But EiffelStudio threatens to make the choice irrelevant.*

A great battle is raging in the software development world over the relative merits of the two leading platforms: J2EE, which utilizes the Java programming language; and Microsoft Visual Studio .NET which focuses on C++, VB.NET and C#. Each offers advantages as well as drawbacks. In addition, EiffelStudio by Eiffel Software is being advocated as a possible alternative. So which should you chose as your development environment?

“When you are looking at making a purchase decision, the compelling differentiators between Java, C++ and EiffelStudio are going to revolve less around apparent benefits, and more around true cost,” said Roger Osmond, a consultant with Amalsoft Corporation, a firm specializing in systems and software design.

**J2EE versus Visual Studio .NET**

Web services allow applications to be assembled out of components running on different machines and operating systems connected using Internet standards. That’s the intention behind Java 2 Platform, Enterprise Edition (J2EE) and Microsoft Visual Studio .NET. So which is better? The simple answer is – it depends who you ask.

“From a development platform perspective, J2EE is farther along and is widely supported, making it a much stronger solution at the moment,” say Gary Hein, Senior Analyst for The Burton Group (Midvale, UT). Nevertheless, he points out, “Visual Studio.NET, which is relatively easy to use, will bring the power of web services programming model to the masses.”

### **Yabba-Java Doo!**

J2EE certainly is more established in large portions of the market. Some would say it is already the de facto standard programming language for Internet-based applications. Further, Java works on most major platforms whereas C# is Windows only, at least for the moment. And in terms of market perception, J2EE is regarded as having greater maturity.

“Microsoft is overly proprietary,” says Stefan Van Overtveldt, Director of Technical Marketing for IBM’s Websphere product line, which harnesses J2EE. “We focus on enabling any type of application, not just .NET or COM components, on any type of platform to participate in the Web services environment.”

But Java doesn’t have it all its own way. While J2EE talks to other Java-based environments easily, some regard it as it as cumbersome, especially when trying to get it to talk to other platforms or web services. Unlike Visual Studio and its associated languages, web services are more of a bolt on in J2EE. Further criticisms concern Web services. Java tends to have too narrow a focus on application-to-application integration rather than integration across organizational boundaries as laid out in the concept of Web services. And Sun has angered some within the programming community by continuing to license the Java language.

### **Nothing but .NET**

Microsoft responds to the many criticisms leveled by the J2EE camp by highlighting the advantages of .NET and Visual Studio. To begin with, all the components come from a single vendor. This should guarantee greater compatibility, and eliminate vendor finger-pointing when problems arise. Also, since Microsoft is incorporating web services support

right into operating systems and server applications, this eliminates the need for an additional layer of middleware to convert these into Web services.

“Visual Studio .NET and the .NET Framework are baked into the plumbing of all the software we produce so that the traditional complexity associated with middleware is greatly reduced,” said Barry Goffe, Group Manager of Enterprise Solutions at Microsoft. “Anyone skilled in earlier languages and technology from Microsoft can quickly pick up .NET.”

To give more force to this argument, Microsoft has adapted Visual Studio to make it even more broadly encompassing. The company now offers four programming languages and associated development environments, each designed to appeal to a particular school of programmer. These include Visual Basic .NET, Visual C++ .NET, Visual C# .NET, and now Visual J# .NET as a Java-language tool for Microsoft .NET.

The downside, of course, is a familiar refrain. Despite Microsoft’s assurances that its Visual Studio tools are multiplatform, its products do tend to work better with each other than with competing systems. For exclusively Microsoft shops, this can be an advantage. But for those that want more options, J2EE advocates say that their own platform lets users select from a wider variety of operating systems and hardware platforms, as well as providing greater scalability.

### **Wrong Debate**

In some quarters, however, the debate of J2EE versus .NET misses the point. While Sun and Microsoft currently have market share advantage, a third alternative is receiving kudos due to ease of use, low cost and platform independence – EiffelStudio, based on the Eiffel programming language.

Eiffel was originally designed in 1985 by object-oriented software guru Bertrand Meyer. It quickly gained popularity, and eventually evolved into EiffelStudio, an Integrated Development Environment (IDE) designed for the Eiffel language.

EiffelStudio has the edge over the other developer languages, platforms and tools as it is multiplatform and multi-language. This means that programmers can use it to write code and not have to even worry about compatibility with Java, C++, or any other language. Eiffel runs on Windows, Unix, Linux, embedded, and also VMS systems. Programmers can create an application in Eiffel (for example, by importing existing applications as a starting point), and migrate it to other platforms. This allows developers the flexibility to maintain their legacy code while developing new code on another operating system. Thus there is no complexity involved when dealing with multiple platform consolidation, reconciliation or communication,

“The code you write for one platform will compile and run identically on another,” said Osmond. “You’re not incurring any of the nasty overhead that you do with something like Java.”

The EiffelStudio development environment runs on Windows (NT, 2000, XP), Unix (Solaris), SunOS, HP 9000, IBM AIX, Unixware, Silicon Graphics, Data General, Fujitsu, DEC, VMS (Alpha and VAX), and Linux. A Mac OS X version is presently in beta testing. Result: EiffelStudio looks and acts the same on all platforms. Linux and PC software development environments, for example, are identical, right down to the graphical interfaces.

Centered on a Design by Contract methodology, EiffelStudio is designed to isolate bugs, minimizing the cost of maintenance that systems designed with other languages have to endure. Design by Contract is a key tool for producing reliable software, for documentation, and for debugging. In addition, the plain English syntax of the Eiffel language lies at the heart of the matter. Eiffel is perhaps the easiest language to learn, write and read and this has a major impact on programming productivity.

“Since going with Eiffel 4 years ago, revenue growth has been steady and dramatic every year,” said Rex Fowler, CEO of Denver, CO-based independent software development firm Fowler Software Design LLC. “We estimate that we take a 30%+ cost advantage (as well as a significant time-to-delivery advantage) into any competitive bidding situation where Eiffel software is matched toe-to-toe with traditional development platforms like C++ software.”

According to the inventor of the Python language Guido van Rossum, a good programmer can reasonably maintain about 20,000 lines of code. However, Eiffel experts contend that its inherent readability and structure mean that the average Eiffel developer can maintain 100-200,000 lines of code, and the most proficient can maintain up to one million lines—about 50 times the industry standard. This level of productivity is hard to approach in any other language.

“Java and C++ work for smaller scale problems, but soon run out of gas,” says Amalsoft’s Osmond, an IEEE and ACM member who holds several patents. With a proven track record of heading teams that consistently deliver software projects on time and within budget, he boasts a computing-industry “Product of the Year” credit amongst his accomplishments. “Due to the Eiffel syntax being so small, I can take one or two people and do the job of 20 to 30 others who use a different programming environment.”

Eiffel runs on virtually any OS platform that runs C, making it highly portable. Further, there is no need for “reverse engineering”, as everything is done concurrently. Gathered together into the EiffelStudio development environment, it contains all of the tools in one easy to find place - there is no need to purchase expensive add-ons such as those offered by Rational and Borland for testing or roundtrip engineering.

According to Osmond, Eiffel is the only “true” object-oriented programming language, one that permits the programmer to focus on overall goals and the solving of key architectural challenges rather than having to spend endless hours in laborious data entry. While C++

and C# are hybrids as opposed to genuine object oriented languages, Java has received criticism as being a C extension rather than a full-fledged object-oriented language.

Using Eiffel, Osmond claims to have solved many engineering challenges, and witnessed productivity gains of 5 to 10 times relative to similar non-Eiffel teams. He believes that the Eiffel camp is growing steadily due to failures experienced with other languages.

“Some developers sell senior management on how ‘cool’ or ‘hot’ their favorite language and tool-set are but eventually this brings them to a very uncomfortable place: the project is spinning out of control and there’s no way to save it,” said Osmond. “The lucky ones discover Eiffel. The rest go on to repeat the same tragedy on a different stage.”

A recent assignment, for example, called for developing a Windows-based graphical management application, an XML management protocol, and a server-side agent for a rather complex system. Two teams were assigned to build this system. One group at a competing company, using mainly Java, had over thirty developers while Osmond's team consisted of himself and two software engineers, each with less than two years of industry experience. Only one had any legitimate Eiffel experience and neither of them was an experienced Windows programmer.

"We delivered our first release in six months and won the contract," concludes Osmond. "Without Eiffel and EiffelStudio, there is no question that the project would have been a total failure."

*For more information about Eiffel Studio, contact Eiffel Software at 356 Storke Road; Goleta, California 93117; (805) 685-1006; sales@eiffel.com; or visit [www.eiffel.com](http://www.eiffel.com).*

###