

# JAVA™ TECHNOLOGY TODAY

Driving the new wave of open, intelligent Web services.

## **THE NAME OF THE GAME**

This much is clear: The name of the game is Web services – sophisticated network software that will bring us what we need, when we need it, no matter where we are, through any device we choose – wireless phone, pager, PDA, or one of the new hybrid handsets.

It's a logical extension of Sun's founding vision of open network computing, but we're hardly alone anymore. Every major technology company – Sun, Microsoft, IBM, Oracle, Hewlett Packard – is talking up the benefits of Web services.

Several new technology standards are emerging to help make this common vision a reality. One is already well established: the Java™ platform – an important element of the Sun Open Net Environment (Sun ONE).

Invented by Sun and developed in cooperation with a global community, the Java platform has matured rapidly since it was introduced in 1995. In fact, the Java programming language has proved to be the most portable ever developed, and is now a vital element in millions of smart cards, cell phones, application servers, and other programs that form the foundation for a new generation of Web services.

Even in such a short time, the impact of Java technology can rightly be compared with milestones such as the invention of Ethernet and the introduction of the microprocessor.

## **DYNAMIC DUO**

With the Java platform, programmers can readily define the operations they want to perform and place that logic on virtually any computer system, regardless of the underlying hardware architecture or operating system – a distinct advantage in the diverse world of Web services.

What Java technology does for logic, its perfect complement – XML – does for data.

Short for extensible markup language, XML is an emerging standard for classifying, structuring, and encoding data. Jon Bosak, the distinguished engineer from Sun who led the team that developed XML, sees the new standard playing a key role in four areas: subscription services that send relatively short bits of information such as stock quotes and weather reports; platform services such as e-mail; portal services for

aggregating information; and trading services for business-to-business transactions. The beauty of XML lies in how it enables companies and services to exchange information more easily and efficiently. Take purchase orders. Every company has them, and there are many common elements, but there is no standard way of expressing those elements. XML makes it possible to encode purchase orders – and all sorts of business information – in a way that is recognizable by all parties.

In short, XML frees data from the bounds of closed or proprietary systems.

Already, the Java platform provides native support for XML processing, and the community is working toward ever tighter integration between the two for use in Web services and other applications. It could be said that XML provides the nouns of e-commerce; Java technology provides the verbs. Together, they form the industry's only universal platform for exchanging logic and data – the basis of Web services.

### **IN THE HERE AND NOW**

Today, the high-tech industry is starting to develop the next generation of Web services – smart service that recognize time, place, and other elements of context – and the Java platform is integrating the key technologies necessary to build those services.

As a result, developers don't have to start from scratch. They can use the Java 2 Platform, Enterprise, Standard, and Micro Editions, as a base for their services, ensuring that those services can be delivered to the broadest possible market – everything from PCs to PDAs, mobile phones to automobiles, even devices that have yet to be invented.

Additional technologies, including the JavaServer Pages™/servlet engine, specific XML technologies (including Java APIs for XML processing, binding, and messaging), and tools for building graphical user interfaces are available from Sun and will be bundled together by leading tool vendors.

In addition to using these building blocks, developers will be able to leverage their existing skills with the Java programming language – popular for both its ease of use and the clean code it engenders – to bring new services to market quickly.

### **DIFFERENT APPROACHES**

It seems like every major technology company and consulting firm has an architecture for Web services. While there is some common ground around standards such as XML, UDDI, and SOAP, there are also important differences – the most notable being between the Java platform and Microsoft .NET. On the one hand, we have an open, industry-wide community that works together to continually extend the potential of the network with compatible, vendor-neutral technologies; on the other, a set of products (more accurately promised products) controlled by a single company with a

record for locking customers into proprietary solutions, selectively sharing its source code, and not documenting (read: hiding) its programming interfaces.

The application programming interfaces of the Java platform – as well as the source code – are available to anyone via download. This stands in stark contrast to Microsoft's "Shared Source" program, which allows only 1,500 of its customers and partners to view *some* of the source code.

To date, there have been almost 5 million downloads of all the APIs and source code of the Java platform by interested companies and developers. This kind of openness allows developers to learn the Java platform from the inside and become more productive more quickly. It also fosters innovation because developers can actually see the source code that determines how the software works.

The key here is interoperability – it's hard to imagine Web services without that. So the answers to the following questions will tell you all you really need to know: Are .NET users free to choose any directory other than Microsoft's Active Directory? Can they replace Microsoft's Kerberos with any other Kerberos server? Can they replace Windows 2000 with any other operating system? The answer, in every case, is no.

With Java – indeed, with all of Sun ONE – users are free to choose the products that best suit their needs.

## **INCREASING MOMENTUM**

In the past year, the Java 2 Platform, Enterprise Edition (J2EE™) has firmly established itself as the platform of choice for developing and deploying enterprise applications. Hundreds of software products from hundreds of vendors are based on the J2EE platform, including many production-class Internet services. Indeed, more large-scale Web-based services are built on J2EE platform than on any other architecture.

Application server suppliers, in particular, have adopted the J2EE technology because of its cross-platform capabilities, productivity, and flexibility. (The list includes leading suppliers such as iPlanet, BEA, IBM, Oracle, and SilverStream, among others.) This important form of middleware connects Net users with applications running in the enterprise – giving new life to legacy systems, while giving employees the information they need to make better decisions and be more productive.

An interesting phenomenon has developed around the J2EE platform. System software suppliers such as IBM and Oracle are building their products from the J2EE reference implementation. The reference implementation is a working, compatible implementation of the J2EE specification. Every Java specification is teamed with a reference implementation, so developers have a "model" application they can analyze to better understand how the technology works. Many build on top of the reference implementation, in much the same way that a personal computer user takes an existing document and modifies it to create a new document.

This model is rapidly expanding the use of J2EE technology, as software companies are now building implementations, based on implementations, based on the J2EE reference implementations.

J2EE technology is also helping service providers to better manage the network in general and run back-office operations such as billing with Java Technology for Service Providers (Java TSP). The J2EE platform has also been used as the basis for new softswitch technology, which promises to revolutionize the switching business. In short, the Java platform is giving service providers and carriers an integrated, easy-to-manage delivery platform.

## **WIRELESS WEB SERVICES**

In the wireless market, Java technology-enabled handsets from Fujitsu, Matsushita, Motorola and Sony among others, are being deployed by key carriers such as Nextel, NTT DoCoMo and LG Telecom. More deployments and handset models will be unveiled shortly.

Mobile phones, pagers, personal digital assistants, automotive systems – all are joining the traditional client, the personal computer, on the Net. The number of these new client types is expected to meet and exceed the total number of installed PCs by 2003. They represent new platforms – always on, always with you – that are opening up exciting new opportunities for networked applications and services.

With a wide range of options beyond the PC, individuals will use the device most convenient for them at the time – which means developers will need to design their services for delivery in a variety of forms. Java technology facilitates this new usage model because it provides the only open, cross-platform and cross-device application and service platform.

End-to-end – from server to switch to desktop to handset to smart card – Java technology continues to revolutionize networked computing. At last, real convergence is starting to take place and users are starting to reap the benefits of being able to access digital content anywhere, anytime, from any device.

JavaServer Pages, J2EE, and iPlanet are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc., in the United States and other countries.