FOCUS ON BEA eWORLD 2003

BEA WebLogic DEVELOPER'S JOURNAL

MARCH 2003 - Volume:2 Issue:3

UPDATES TO WEBLOGIC

RETAILERS PLEASE DISPLAY UNTIL JUNE 30, 2003

weblogicdevelopersjournal.com

$8.99US $9.99CAN

03>

0 09281 03424 7

SYS-CON MEDIA

ReportingEngines

www.reportingengines.com/download/jdj2.jsp

BEA WebLogic DEVELOPER'S JOURNAL

EDITORIAL ADVISORY BOARD
LEWIS CIRNE, SHAUN CONNOLLY, STUART HALLOWAY, KEVIN JONES, TYLER JEWELL, WAYNE LESLEY LUND, SEAN RHODY, BLAKE STONE

FOUNDING EDITOR
PETER ZADROZNY

EDITOR-IN-CHIEF
JASON WESTRA

EDITORIAL DIRECTOR
JEREMY GEELAN

EXECUTIVE EDITOR
GAIL SCHULTZ

EDITOR
NANCY VALENTINE

ASSOCIATE EDITORS
JAMIE MATUSOW, JEAN CASSIDY

ASSISTANT EDITOR
JENNIFER STILLEY

WRITERS IN THIS ISSUE
STEVE BOOM, SCOTT DIETZEN, MIKE FISTER, YARON GOLAND, EVELYN HOBSON, WILL LYONS, SHANE PEARSON, JIM RIVERA, CARL SJOGREEN, VITTORIO VIARENGO, JASON WESTRA, DAVE WILBY

SUBSCRIPTIONS
For subscriptions and requests for bulk orders, please send your letters to Subscription Department.
SUBSCRIPTION HOTLINE:
888-303-5282
Cover Price: $8.99/Issue
Domestic: $149/YR (12 Issues)
Canada/Mexico: $169/YR
Overseas: $179/YR
(U.S. Banks or Money Orders)

PRESIDENT AND CEO
FUAT A. KIRCAALI

COO/CFO
MARK HARABEDIAN

VP, BUSINESS DEVELOPMENT
GRISHA DAVIDA

SENIOR VP, SALES & MARKETING
CARMEN GONZALEZ

PRODUCTION CONSULTANT
JIM MORGAN

ART DIRECTOR
ALEX BOTERO

ASSOCIATE ART DIRECTORS
LOUIS F. CUFFARI • RICHARD SILVERBERG

ASSISTANT ART DIRECTOR
TAMI BEATTY

VP, SALES & MARKETING
MILES SILVERMAN

ADVERTISING SALES DIRECTOR
ROBYN FORMA

ADVERTISING ACCOUNT MANAGER
MEGAN RING-MUSSA

ASSOCIATE SALES MANAGERS
CARRIE GEBERT • ALISA CATALANO
KRISTIN KUHNLE • LEAH HITTMAN

PRESIDENT, SYS-CON EVENTS
GRISHA DAVIDA

CONFERENCE MANAGER
MICHAEL LYNCH

FINANCIAL ANALYST
JOAN LAROSE

ACCOUNTS RECEIVABLE
KERRI VON ACHEN

ACCOUNTS PAYABLE
BETTY WHITE

VP, INFORMATION SYSTEMS
ROBERT DIAMOND

WEB DESIGNERS
STEPHEN KILMURRAY • CHRISTOPHER CROCE

ONLINE EDITOR
LIN GOETZ

CUSTOMER SERVICE REPRESENTATIVE
MARGIE DOWNS

JDJ STORE MANAGER
RACHEL MCGOURAN

EDITORIAL OFFICES
SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645
Telephone: 201 802-3000 Fax: 201 782-9637
SUBSCRIBE@SYS-CON.COM
BEA WebLogic Developer's Journal (ISSN# 1535-9581) is published monthly (12 times a year)
Postmaster: Send Address Changes to
BEA WEBLOGIC DEVELOPER'S JOURNAL,
SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645

SYS-CON MEDIA

Convergence – BEA eWorld 2003

BY JASON WESTRA
EDITOR-IN-CHIEF

This year's BEA eWorld 2002 show is the center of attention for BEA's product announcements and vision for the upcoming year, exciting stuff indeed. The theme for this year's conference is "convergence." You'll notice that this theme is likewise ingrained in the articles in this issue of *WLDJ* to provide continuity with the show.

This issue contains articles from visionaries in the Internet infrastructure landscape, including BEA, HP, Intel, and Yahoo!. Take a look at the BEA Platform from a pervasive computing perspective… applications built in the future will have to be designed to reach an audience over numerous computing platforms or distribution channels. As data and distribution channels converge, application infrastructure platforms like BEA WebLogic and computing platforms from the likes of HP and Intel are necessary to get the information into the right hands at the right time, anytime to be exact.

I look forward to seeing how BEA will once again revolutionize development of enterprise applications this year with its release of BEA WebLogic Platform 8.1. BEA took advantage of the past year, fine-tuning the integration of its complete product stack into a single, installable, manageable platform. This month, Will Lyons, a senior product manager for BEA, explains the new enhancements in WebLogic Platform 8.1. BEA has made an even deeper convergence of product lines with an emphasis on a unified development environment under WebLogic Workshop. The initial release of WebLogic Workshop, announced at last year's eWorld, focused on easing the design of Web services. The new Workshop now supports the BEA Platform's entire product stack, including Portal and WebLogic Integration. With WebLogic Workshop Platform 8.1, the development of Web applications, portals, business processes, and Web services all benefit from Workshop's innovative approach toward visual design, development, and runtime components.

This month, Carl Sjogreen of BEA emphasizes the importance of BEA's vision of converging design time and runtime components within WebLogic Workshop. He elaborates on its visual design environment, which now allows the design of workflows and drag-and-drop design of Web pages. Javadoc annotations are then used to describe the properties of these components and enable Workshop's runtime to build, compile, and deploy the J2EE components necessary to implement the plumbing for you.

"Workshop is exciting," you say, "but isn't it proprietary?" A convergence of open-source and proprietary, yet standards-based, software is taking place in the application infrastructure market. Even Workshop's Web application runtime is based on Apache Jakarta's Struts, the leading JSP framework on the market (or not on the market; it's open source and free). This issue includes another column from the office of the CTO, this month by Scott Dietzen and Yaron Goland, and an interview with Olivier Helleboid, BEA's president of products. Scott mentions the importance of both WebLogic and open source in today's application infrastructure, and Olivier confirms the benefits of building on standards and open source.

The question remains, "Is Workshop proprietary?" While BEA is integrating key pieces of open source and enriching the development experience with Java Web Services (JWS) and J2EE controls, it is also giving back to the development community. BEA plans to open up some of its product stack in the near future and it is currently involved in the Java Community Process (JSR-181) to standardize its innovations around JWS.

"Convergence" is a theme that transcends BEA eWorld 2003. *WLDJ* will continue to cover it and BEA's new platform in the months to come.

AUTHOR BIO...

Jason Westra is the editor-in-chief of ***WLDJ*** and an application architect for a global consulting firm. Jason has vast experience with the BEA WebLogic Server Platform and was a columnist for ***Java Developer's Journal*** for two years, where he shared his WebLogic experiences with readers.

CONTACT: jason@sys-con.com

A New Platform

Enhanced Functionality to

BEA has begun the rollout of BEA WebLogic Platform 8.1, a new release of BEA's application infrastructure platform product, with powerful new functionality for building and integrating enterprise applications.

When BEA introduced WebLogic Platform in 2002, we articulated our vision of an application infrastructure platform that enables developers to build and integrate enterprise applications easily and rapidly, that simplifies the deployment and administration of these applications, and that is inherently extensible, facilitating widespread adoption and market acceptance. WebLogic Platform 8.1, which was available in beta form in March 2003, implements and extends this vision by converging Platform development tools into a unified visual development environment, and by enhancing the ability to integrate WebLogic applications with other enterprise applications. This article provides an overview of WebLogic platform 8.1, including the major themes of this release and what they mean for developers.

WebLogic Platform 8.1 and Component Products

BEA WebLogic Platform 8.1 is an integrated platform that enables the combined development of Web applications, Web services, Enterprise JavaBeans, workflows, messaging applications, enterprise portals, trading partner applications, and more. Applications can be integrated with other enterprise applications and managed in a fully integrated, high-performance, and highly reliable runtime environment. It provides this in a single, integrated product package.

- ***WebLogic Workshop 8.1 Platform Edition:*** A unified visual development environment and runtime framework for building enterprise-class applications for WebLogic Server, WebLogic Integration, and WebLogic Portal.
- ***WebLogic Integration 8.1:*** An integration solution, built on WebLogic Server, for connecting enterprise applications, databases, business processes, and business partners. Includes business process/workflow management, data transformation, message brokering, adapter integration, and B2B support.
- ***WebLogic Portal 8.1:*** A complete portal framework, built on WebLogic Server, for the integrated enterprise. Includes support for portal application development, application aggregation, portal rendering and personalization, user and entitlement management, content management, and delegated portal administration.
- ***WebLogic Server 8.1:*** The leading Web applica-

BY **WILL LYONS**

AUTHOR BIO...

Will Lyons is the product manager for the BEA WebLogic Platform product. He has been in product management with BEA since 1999, and has more than 15 years of experience in the information technology industry.

CONTACT...

wlyons@bea.com

tion server and foundation for the WebLogic Platform. Includes full J2EE runtime, development tool, and administration support for highly available, scalable, and secure applications.
- ***WebLogic JRockit 8.1:*** A high-performance JVM optimized for server-side performance and scalability. Supported by all WebLogic Platform components and now fully integrated into the WebLogic Platform package, in addition to being available as a stand-alone JVM.

BEA continues to make these products available separately, enabling customers to select the components that are most appropriate to a specific project with the assurance that additional WebLogic Platform functionality can be added and integrated later, as enterprise application requirements dictate.

The new features of WebLogic Platform 8.1 and the component products above are the focus of this article. In addition, BEA offers complementary products that help to define the BEA WebLogic Enterprise Platform, including:
- BEA WebLogic Adapters for application integration
- BEA Liquid Data for WebLogic
- JBuilder, WebLogic Edition
- BEA Tuxedo

BEA will provide new releases and/or certification of all of the products for use with WebLogic Platform 8.1.

Convergence and Integration

In defining the new features of the WebLogic Platform 8.1 release, we focused on major themes that would provide the greatest benefit to customers and developers. One theme was integration. Most WebLogic applications are required to integrate with third-party application packages, custom applications, messaging systems, Web services, content management systems, trading partners, and so on. To address these requirements, WebLogic Platform 8.1 contains major enhancements to integration capabilities, including simplified development, new runtime functionality, and improved administration to better leverage existing enterprise environments.

A closely related theme is convergence. More and more customer solutions require multiple WebLogic Platform runtime components – combinations of WebLogic Server, Integration, and Portal. In response, we have converged and unified the development tools for these products into WebLogic Workshop Platform Edition, which now supports the visual development of WebLogic Portal and WebLogic Integration applications, as well as Web services and Web applications for WebLogic Server. For J2EE developers, we have enhanced the J2EE compilers, tools, and utilities provided with WebLogic Server, and continue to support tools such as JBuilder, WebLogic Edition for building J2EE applications. However, for a broad base of developers, WebLogic Workshop provides an easy-to-use visual development environment that enables development of WebLogic Server, Integration, and Portal applications without requiring J2EE expertise.

Enhancements in the areas of convergence and integration have resulted in a platform that provides extraordinary new capabilities for custom application development and ease of integration, while continuing to leverage the proven reliability and scalability of WebLogic Server and WebLogic Server J2EE support. For developers, this means reduced training time, higher productivity, higher software and skill reuse, reduced time-to-market/deployment, and increased ROI for enterprise application development projects. It also means broader acceptance of WebLogic Platform in your enterprises, and broader adoption by ISVs who will deliver complementary products and layered applications.

WebLogic Workshop 8.1

WebLogic Workshop is both a development tool and a runtime framework that abstracts underlying J2EE constructs from the Java application developer. It presents a visual development environment for application development, and generates the underlying J2EE infrastructure for running these applications on WebLogic Server. This release includes new "designers" that enable development of new types of applications. This section describes WebLogic Workshop 8.1 Application Edition, and new features for development of WebLogic Server applications. Later we'll look at WebLogic Workshop 8.1Platform Edition, including support for WebLogic Integration and Portal.

The first release of Workshop provided prepackaged controls that enabled access to external resources from Web services applications. In this release, Workshop supports the development of custom controls that incorporate business logic. For example, you can create a control that performs a database query (via a database control), receives a JMS message (via a JMS control), executes business logic, and returns a result to the Workshop application. Once developed, such controls, including nested controls, can be leveraged across all applications developed with Workshop. Custom control development and reuse is a powerful capability that will promote software reuse and developer productivity, and simplify application integration.

Workshop also provides new designers for building Web applications within the Workshop IDE. JSP pages can be developed using the same development environment familiar to Web services developers. A different visual metaphor is used, but the structure of the overall Workshop environment (two-way source/visual editing, use of annotated Java code, integrated testing and debugging, access to external resources through controls, etc.) is the same. Workshop leverages Struts technology to enable the creation of Pageflows that combine multiple JSPs, and that specify the flow of applications and data across them.

Workshop contains a variety of additional enhancements to existing capabilities. New Web services features include more reliable and secure Web services, leveraging Web services support provided by the WebLogic Server runtime. There are core IDE enhancements as well, such as improved debugging support and software source control integration support. Finally, Workshop has added designer support for "XML Beans," providing developers with easy access to data stored in XML format by automatically generating interfaces based on XML Schema. With the ability to build custom controls and Web applications, Workshop now provides a rich environment for developing a broad range of enterprise class applications for WebLogic Server.

WebLogic Integration 8.1

As stated earlier, most WebLogic applications have at least some integration requirements, and integration projects typically involve use of workflow and data transformation. To enable the easy development of such applications, WebLogic

Workshop now supports new designers that enable the creation of workflows and data transformations from within WebLogic Workshop Platform Edition.

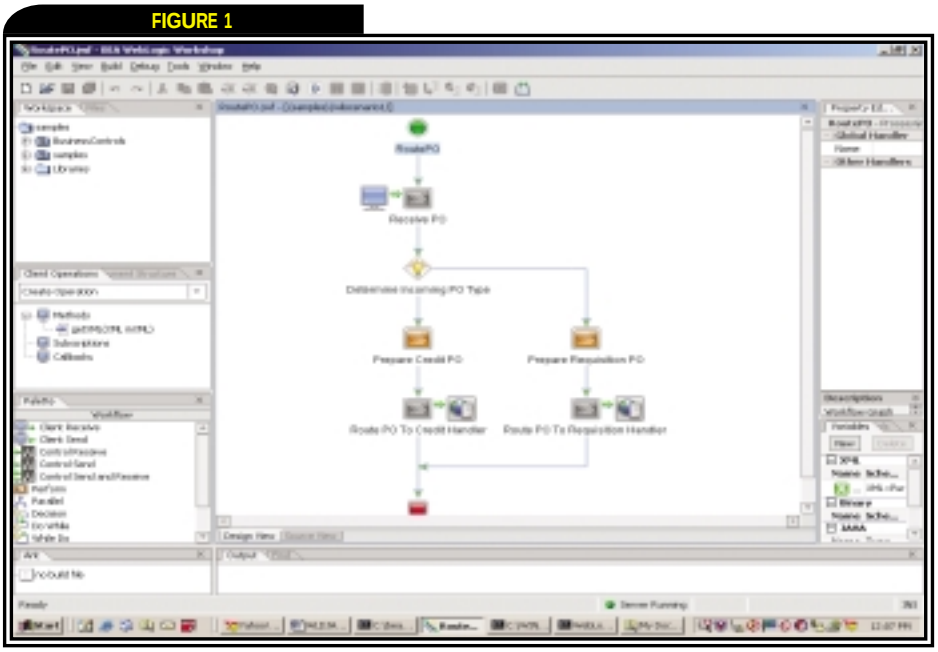
The visual metaphor for building workflow applications is shown in Figure 1 with a sample workflow. The green and red icons represent the start and end nodes of the workflow. The developer inserts steps into the workflow to represent the business process by dragging and dropping additional workflow nodes from the Workflow Palette. Having created an overall structure, the developer can add methods and callbacks to nodes, and add business controls to create a workflow process. Alternatively, developers can begin workflow development using business process templates, and customize them for their specific business requirements. The resulting workflow designs control execution of workflow process decisions, access to external resources and exchange of business documents across applications, and execution of business logic within workflow nodes.

Integration applications typically involve data transformations – business documents retrieved from one data source or application must be transformed to the format expected by another data source. WebLogic Integration 7.0 supported XML<->XML data transformations using XSLT, and continues to do so in this release. However, Workshop 8.1 offers new capabilities for the development of transformation controls that can be incorporated into workflows. In the case of XML<->XML transformations, Workshop provides a designer that enables mapping from one XML Schema to another, and generation of a transformation control that includes an X-Query expression defining the transformation. To perform XML<->Binary transformations, Format Builder is used to define the transformations, which are then incorporated into Workshop transformation controls.

Once again, the new workflow and transformation designers leverage the same core Workshop development features, such as the use of annotated Java code, two-way visual/source editing, ability to set workflow properties, integrated debugging, and use of controls for access to external resources. Having developed integration applications in the Workshop environment, integrating these with other Workshop applications is easily accomplished.

These enhancements to the development environment are enabled by enhancements to the WebLogic Integration runtime. For example, the Integration runtime supports the execution of workflows developed in Workshop, and a new X-Query engine supports high-performance runtime execution of X-Query-based transformations. In addition to providing support for long-running stateful business processes, WebLogic Integration provides a new message broker runtime for execution of stateless workflows with high-performance message routing and transformation. Other enhancements include a message repository for high-performance message processing, native support for Web services, and a new Web-based Administration Console that leverages the JMX infrastructure provided by WebLogic Server. The Integration runtime also contains enhancements for B2B application support, and continues to provide support for application views to enable control-based access to application adapters. BEA has provided a wide range of application and technology adapters for WebLogic Integration 7.0, and will continue to provide such complementary adapters for WebLogic Integration 8.1.

FIGURE 1



Building workflow applications

WebLogic Portal 8.1

The Workshop environment has been further extended with new designers that support development of WebLogic Portal applications, such as the construction of portal pages, describing navigation among these pages, and including portlets within the portal pages being developed. This is all accomplished with drag-and-drop creation, two-way visual/source editing, modification of portal and portlet properties, integrated test/debugging, and so on. Workshop facilitates the inclusion of Workshop-based Web applications, controls, Web services, and business integration workflows (and externally developed Web applications as well) into your portal application. This application leverages all the existing and new features provided by the WebLogic Portal runtime environment. In this way Workshop facilitates the aggregation and integration of applications into enterprise portals to support the targeted delivery of application and content to portal users, and to facilitate the management of portals by business and IT owners.

In addition to providing the new Workshop development environment, and the ability to easily aggregate and integrate Web applications into enterprise portals, we've provided architectural and administrative enhancements to the WebLogic Portal runtime that address requirements for highly flexible deployment and management of portal applications. Architectural enhancements such as support for multiple portals per Web application, and support for multiple "desktops" (end-user entry points) per portal provides developers with greater flexibility to build portals with complex hierarchies. Multiple administration enhancements enable flexible administration of portal environments, application of entitlements, delegation of administration tasks, and delegation of functions such as page and portlet creation to end users. New content management capabilities provide federated access and

management of content repositories. Portlet enhancements enable creation of high-performance, standards-based portlets that are easy to manage and maintain. The new features reinforce WebLogic Portal's ability to address requirements scaling from simple Web applications to a complex Portal networks.

WebLogic Server 8.1

WebLogic Server 8.1 reinforces WebLogic Server's position as the underlying foundation for the WebLogic Platform. In addition to enabling the runtime framework for Workshop applications, the latest release of WebLogic Server includes usability enhancements to development, configuration, and management tools; performance and reliability enhancements; and integration enhancements that are available to developers and also leveraged by other WebLogic Platform components.

For developers, WebLogic Server continues to support tools and utilities, in addition to WebLogic Workshop, for the J2EE developer building EJBs, or who prefers to develop in a command-line environment. Usability enhancements for these developers include a single J2EEC – EAR compiler, replacing the separate compilers in previous releases, enhancements to rapid iterative development and deployment, improved error message handling for debugging, simplified integration with foreign JMS providers, and Web services task enhancements.

For developers and administrators, WebLogic Server 8.1 provides enhancements to configuration and management tools that will assist both developers and administrators, including simplification of installation and domain configuration using the Configuration Wizard, and usability enhancements to the Admin Console that simplify commonly performed administration tasks. BEA continues to work closely with third parties who provide value-added monitoring and management capabilities.

WebLogic Server's leadership performance and availability has been raised to a new level through enhancements that include EJB, JDBC, security subsystem and load balancing optimizations, and address requirements of customers demanding the highest levels of performance and scalability.

Integration enhancements include JMS reliability improvements and enhancements to Tuxedo integration support. Web services now support reliable, exactly once delivery over HTTP and JMS, and support for the WS-Security Web services security standard, which enables encryption and signing of SOAP messages, independent of the underlying transport being used.

WebLogic JRockit 8.1

WebLogic Platform now includes the WebLogic JRockit JVM on Windows and Linux systems to provide a high-performance, scalable, and reliable JVM optimized for server-side applications. Historically, JVMs have been evolved to support a diverse set of requirements, including client application requirements. Our analysis has shown that in a wide range of server applications for which JVMs are not generally optimized, the JVM has been a key component that limited server application performance and scalability. WebLogic JRockit is the first commercial JVM built from the ground up for server-side Java applications. WebLogic JRockit design and development has focused on three major goals:

- Combine the best adaptive optimization techniques for code generation, memory management, thread management, and native methods.
- Make the system as independent as possible from the underlying hardware and operating system.
- Provide a robust management framework for JVM profiling and tuning.

With optimization for the types of applications supported by WebLogic Platform, proven and enhanced over multiple releases during the past year, JRockit provides an ideal JVM environment for deploying WebLogic Platform server applications on Windows and Linux systems. The latest release, WebLogic JRockit 8.1, provides a range of new features including:

- J2SE 1.4.1, JVMPI, and JVMDI support
- Workshop IDE support
- Integration with third-party profiling, development, and management tools
- Performance enhancements
- 32- and 64-bit Intel architecture support on Windows and Linux systems

Integration of Components into a Single Platform

The first release of the WebLogic Platform set the standard for the integration of key application infrastructure technologies into a single, proven, standards-based application infrastructure platform. All of the platform integration features already delivered and supported have been brought forward to WebLogic Platform 8.1: common packaging and licensing, integrated installation and ease of configuration, common WebLogic Server foundation, coexistence of platform product components within the same server process, interoperation across product components, unified documentation, common support model, and consistent hardware and operating system support.

In addition to providing extraordinary richness of functionality in each of the WebLogic Platform component products, in this release BEA has extended the seamless integration of the WebLogic Platform as a whole through component features, such as the unified Workshop development environment, and enhanced and simplified the administration discussed earlier. BEA has also provided additional integration enhancements to provide a seamless experience for developers and administrators.

For example, enhancements to the Configuration Wizard and configuration templates enable not only the initial configuration of WebLogic Platform environments, but also the ability to flexibly add WebLogic Platform components to existing configurations. This facilitates the addition of the WebLogic Portal, for example, to configurations where WebLogic Integration applications are already deployed. Further, we're providing developers with the ability to create their own configuration templates to enable the "handoff" of WebLogic Platform application configurations to other developers, administrators, or customers. Another area of enhanced cross-platform integration is in the area of security. All WebLogic Platform components now fully leverage the security infrastructure enhancements first provided in WebLogic Server 7.0. This means greater consistency in user management, integration with third-party security providers, support of security policies and entitlements across platform applications, and use of new security features such as WS-Security.

Conclusion

We're extremely proud of this release because we believe it will help developers and solve real customer problems. We've provided a preintegrated infrastructure that enables you to focus on solving problems of enterprise application integration, rather than worry about integrating products from your vendors.

Visit our Web site (www.bea.com) to download the beta software – we hope you'll be as excited as we are.

BEA WebLogic INTEGRATION

UNIFYING THE INTEGRATION ENVIRONMENT

BY **VITTORIO VIARENGO**

AUTHOR BIO...

Vittorio Viarengo is senior director of product management with BEA Systems. He is responsible for the direction of BEA WebLogic Integration, business process management, and the Web services development framework.

CONTACT...

vivi@bea.com

Enterprise application integration, now the centerpiece of corporate IT strategy, remains unacceptably high in cost and effort for one reason. Today's typical IT organization has separate environments for application development and integration.

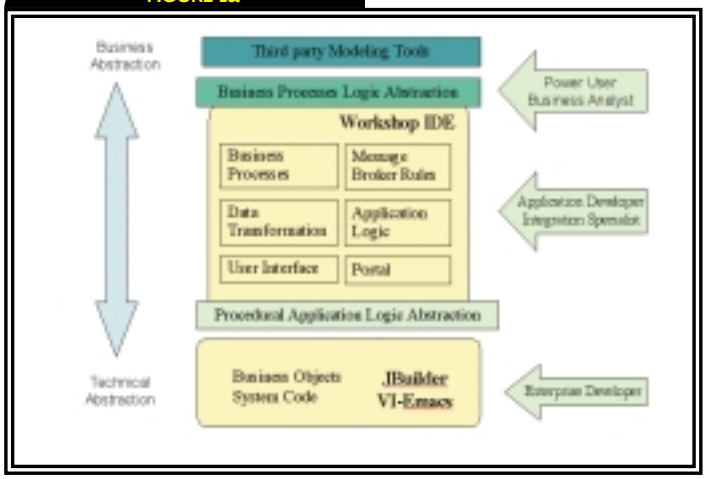
BEA WebLogic Integration 8.1 bridges this gap – converging application development and integration technologies into a unified platform. It leverages the BEA WebLogic Server as the underlying deployment environment, uses Web services to integrate distributed systems inside and outside the organization, and utilizes BEA WebLogic Workshop to simplify services-oriented development of applications (SODA).

SODA requires IT staffs to design for application integration while the applications are being built, minimizing the burden of after-the fact integration. According to Gartner, SODA has the potential to reduce integration costs by 30% or more. Using service-oriented architecture (SOA) and SODA together, IT staffs can build a foundation that catalyzes a movement from monolithic to more flexible systems. BEA believes that "building to integrate" represents a significant innovation that will help to reduce the cost, and increase the effectiveness, of integration projects in the Global 2000.

Based on a services-oriented design, WebLogic Integration 8.1 will bridge the "gaps" in today's integration picture (see Figures 1a and 1b). A close look at the integration and application gap clearly indicates that these environments need to be unified for two reasons. First, all integration projects require IT departments to write code to fill the inevitable gaps in the proprietary integration solution, or to implement business logic that is unique to a core business. Second, most applications require integration, even if only with a database. If people don't perceive accessing a database as an integration challenge, it's because over the last 10 years SQL and JDBC-type standards and off-the-shelf technology have made database integration a natural part of the development process.

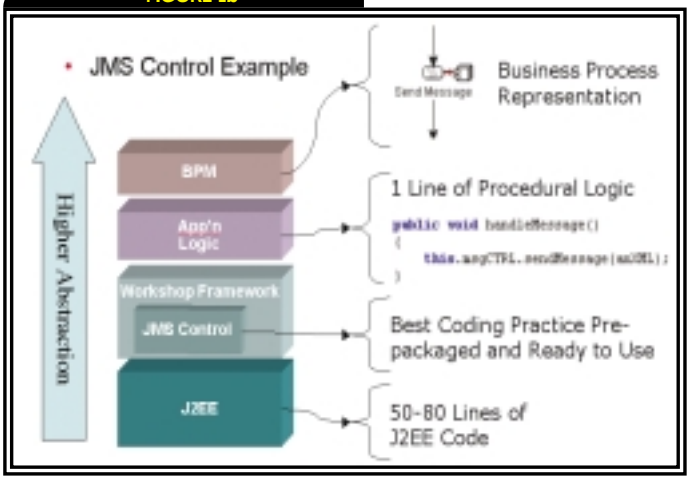
The second gap, between integration modeling tools and the integration deployment environment, calls for a communication solution. Traditional business process management (BPM) tools target business analysts, helping them paint

FIGURE 1a



Development continuum

FIGURE 1b



JMS control

a high-level picture of integration. But implementing that picture requires staff with expensive, specialized knowledge of proprietary integration and deployment environments. To optimize the use of these resources, our research has shown that an integration environment must:

- Let the business analyst speak to an IT group's application developers and integration specialists communicate in terms of business process model requirements
- Equip IT staff with the tools to implement and bind business processes without requiring low-level coding and details

The third gap consists of the spaces between varied layers of integration technologies that have accumulated over IT's fast-evolving history. Early EAI projects focused on connectivity and data transformation, adapters, message brokering, and binary transformations. The proprietary nature of back-end systems propagated throughout the enterprise resulted in solutions that were hard to maintain and evolve. More recently, EAI products have layered BPM on top of proprietary solutions to capture and automate the semantic of the transactions being executed via the exchange and transformation binary messages. With B2B came a shift toward standards-based integration, using XML as business process modeling. Automation became the focus, with the public processes being exposed and shared with business partners. Again, additional layers drove the integration chasm wider.

To optimize enterprise integration, BEA has concluded that loosely coupled integrations are easier to maintain than traditional tight and rigid integrations; asynchronous communication is critical to conduct and protect business operations; coarse-grained communications maximize the efficiency of typically high-cost communication between loosely coupled systems; and the enterprise's need to know "who can access what" makes security services, and user and trading partner management, essential requirements for any integration project.

A Unified Integration Framework

The new version of BEA WebLogic Integration is designed to bridge all the existing chasms. BEA WebLogic Integration's unified framework unites common activities across the whole spectrum of application development and integration based on three development and integration scenarios:

- At the bottom, object-oriented J2EE enterprise developers create business objects and write the necessary system-level code.
- At the next level, business application developers – closer to the line of business – take objects and components from the J2EE developers and assemble them into business applications, adding application logic and user interfaces.
- At the top, business analysts model business processes where the primary activity is the routing and transformation of XML business documents across loosely coupled systems.

In order to create a development continuum across these layers, BEA believes that each user needs the right level of abstraction. An abstraction is needed on top of J2EE to allow application developers and integration specialists -- WebLogic Integration's target users -- to implement business processes without requiring object-oriented and J2EE knowledge. Another layer of abstraction, based on business process logic, is required at the higher level to enable business analysts to communicate the requirements, validate the IT implementation, and monitor the behavior of the overall systems at run time in business terms.

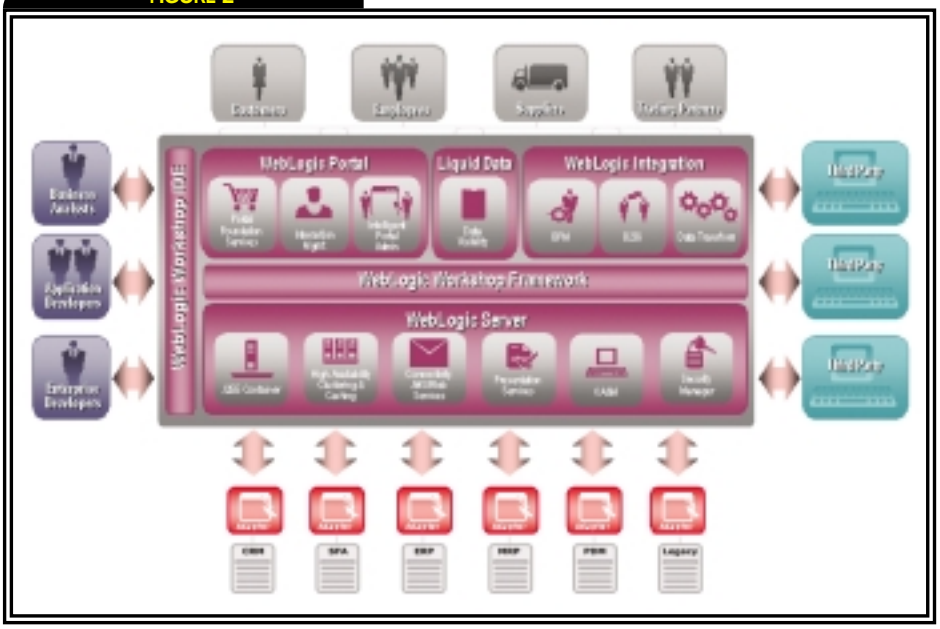
The Components of the Unified Integration Platform

BEA WebLogic Integration is built on WebLogic Server and utilizes a single architecture that leverages the WebLogic Workshop framework, as well to supply a high level of abstraction to simplify application development and integration on the J2EE platform. The WebLogic Workshop framework supplies all the components needed to build and integrate applications – from business processes to application integration – within a unified development and integration environment.

WebLogic Workshop Controls

WebLogic Integration leverages the WebLogic Workshop runtime framework to provide access to resources via high-level controls to enhance developer productivity. This architecture also supplies a unified programming model based on procedural logic development and event-driven programming. The platform abstracts the low-level technical details of the J2EE APIs and back-end resources for application developers and integration specialists. For example, consider sending a JMS message. At the J2EE coding level, sending a JMS message requires 50 to 80 lines of code because the

FIGURE 2



Platform for Enterprise Intergration

J2EE API provides the user with access to all the JMS parameters and knobs.

Within the WebLogic Workshop framework, you can use a JMS control that prepackages the 80 lines of code into one high-level component. This component lets you send a JMS message by writing a simple line of Java within some application logic, or by entering a single step in a business process model. While this approach trades flexibility for ease of use, BEA believes that this approach is suitable for 80% of applications. And even when an out-of-the-box implementation doesn't fit, the enterprise developer can still access the J2EE APIs (see Figure 2).

BEA WebLogic Integration will provide a large number of out-of-the-box controls, starting with an extensive portfolio of resources ranging from back-end packaged applications and J2EE resources to B2B networks and users.

WebLogic Workshop IDE

The new WebLogic Workshop IDE breaks new ground by targeting application developers rather than J2EE experts. BEA realizes that application developers need a high-level tool that allows them to be productive without requiring J2EE expertise. This sets the WebLogic Workshop IDE apart from other attempts to implement a unified IDE, which typically boast IDE components but still require J2EE expert support behind the scenes.

With a high-level interface to J2EE and the WebLogic Integration components in place, you have an integrated development and integration environment within one tool. This lets you model business processes, define transformations, build Web services, write application logic, define Web-based interfaces and Web Pageflows, and build process portals.

The WebLogic Workshop IDE eases the building of custom controls, enabling application developers, integration specialists, and ISVs to build their own controls and extend the platform. In fact, any artifact built within the WebLogic Workshop IDE automatically becomes a control that can be reused across the board and included in business processes.

Application Integration Architecture

Now that we've defined the architecture, let's look at how this release supports the traditional components of integration: connectivity, message brokering, BPM, data transformation.

Connectivity

BEA's connectivity strategy is based on open standards wherever possible. Even where access to proprietary interfaces is required, the architecture is designed so that the low-level proprietary characteristics of the legacy systems are not propagated throughout the architecture. Rather, they are confined to back-end systems and wrapped with a standard interface like J2EE CA or Web services. In this way, all resources look like business services to the consumer.

WebLogic Integration can connect to three types of main resources:

- ***Legacy Systems and Packaged Applications:*** A set of J2EE CA–based adapters is provided to integrate with back-end systems, including all the major packaged business applications. The platform supports vertical networks and formats such as FIX, SWIFT, HIPAA, and HL7. In addition, it supports the development of J2EE CA–based custom adapters using an Adapter Development Kit. These adapters are exposed to the WebLogic Workshop IDE via the application view control. To configure an application view control, the application expert uses the WebLogic configuration tool to configure the adapter and define relevant high-level business operations and events.
- ***J2EE Resources:*** Includes a set of controls to access any J2EE resources such as JMS, EJB, databases (JDBC), and J2EE CA.
- ***B2B and EDI:*** Uses controls to provide access to resources outside the firewall via Web services, RosettaNet, and ebXML as well as some of the more basic protocols like e-mail, HTTP, and FTP.

Business Process Management (BPM)

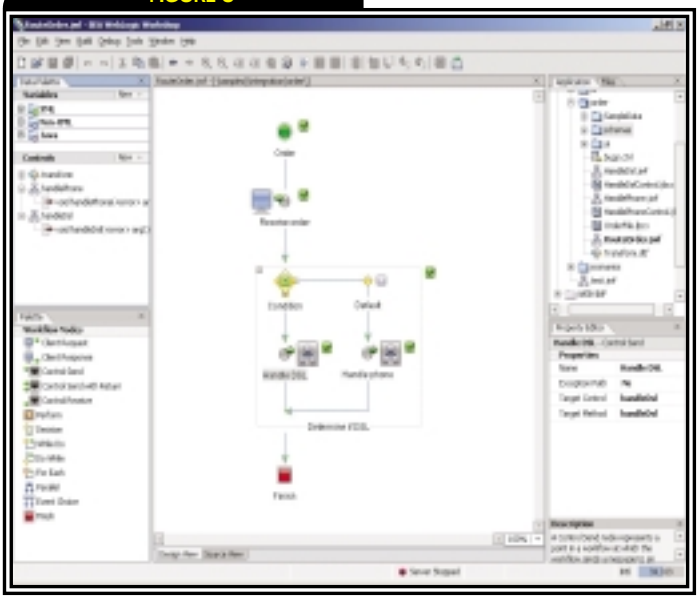
WebLogic Integration BPM allows users to model and execute business processes that span multiple internal systems, external resources, and users. From the BPM perspective, the enterprise is a set of business services that are accessed via controls and can be orchestrated to model a business process. WebLogic Integration supports synchronous and asynchronous communications, and stateless and stateful processes.

The WebLogic business process engine offers flexibility by keeping the ability to write Java code constantly available (see Figure 3). Writing a few lines of procedural Java logic is often the most expedient way to accomplish a task, such as determining the lowest price across a set of XML documents that contain pricing information. WebLogic's solution is to make one coarse-grained step in the business process called 'find lowest price', and let the application developer develop the best implementation.

Message Brokering

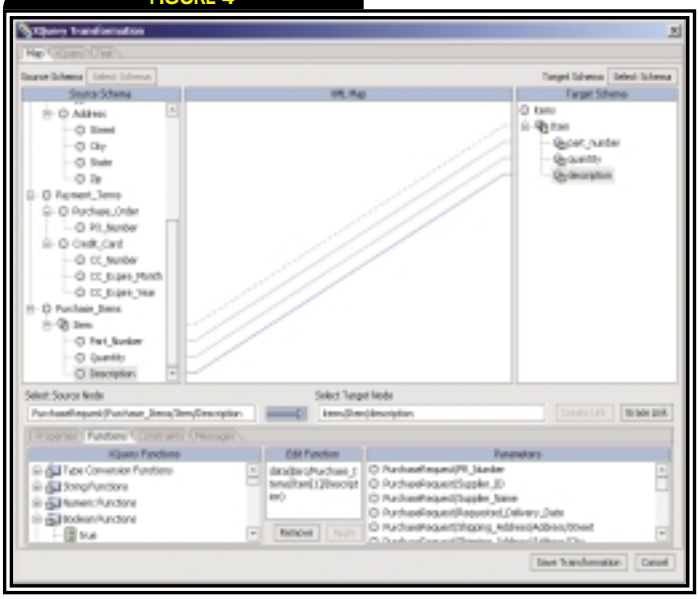
The WebLogic Integration message broker provides business processes with a channels-based "publish & subscribe" communication mechanism. This enables busi-

FIGURE 3



BPM UI

FIGURE 4



Xquery Mapper

ness processes to communicate in a loosely coupled, anonymous manner using a business-naming paradigm. For example, a purchase order routing process that subscribes to the "New Order Entered" channel is activated each time a new order message is published to that channel.

Using the IDE, developers can define the channel naming hierarchy and specify the channels that it will publish and subscribe to for each business process. Filters – defined using XQuery – allow the subscription to be refined based on message content. Using the previous example, a business process can use a filter to subscribe only to purchase orders over $10,000 that are published to the "New Order Entered" channel.

The Message Broker supports event generators that publish events from external sources to channels. WebLogic Integration will provide event generators for JMS, file, e-mail, and timed events. Adapters publish events from packaged applications to channels.

Data Transformation

WebLogic Integration provides components and tools to support the three main categories of transformation: binary to XML, XML to XML, and XML to binary.

The platform packages transformation components as controls, treating them as resources that can be reused across multiple processes and integration solutions, such as a set of transformations from RosettaNet to OAG (Open Application Group) formats. The new data transformation feature uses XQuery for XML-to-XML transformation. XQuery is an XML standard in the final stage of W3C standard process.

Traditionally, XML-to-XML transformation has used XSLT, a task XSLT was never designed to do. XSLT was intended to handle semi-structured data and document-based transformation: taking XML, applying XSLT, and converting the XML into content that can be presented to an end user or device. XSLT is recursive, pattern-matching based, and declarative – three of the most difficult concepts to learn. Moreover, an XSLT transformation defines how to perform a transformation, not what to perform, limiting the XSLT processor's optimization because it is bound to user code.

While WebLogic Integration fully supports XSLT, the new release introduces a new XQuery processor and tools to make transformation easier and faster. XQuery looks much like SQL for XML, making it familiar to application developers who already understand SQL. Further, BEA preliminary performance tests show that because XQuery defines *what* to do, not *how* to do it, an XQuery transformation can be optimized much more than XSLT transformations (see Figure 4).

System Management

WebLogic integration provides a unified, Web-based administration console that addresses:

- Workflow management and monitoring
- Task or work list administration
- Trading partner administration
- Message routing management
- Event and adapter configuration
- Security and archiving administration

While the system assumes that the integration environment will be managed primarily from the WebLogic Integration administration console, it also provides JMX interfaces to support third-party administration tools.

WebLogic Integration separates runtime administration from offline analytics by maintaining two logical database stores. The online administration database contains runtime data about the integration engine, business process states, and message history. This repository is designed for performance. To scale and to retrieve information as quickly as possible, its data is maintained in an optimized format. According to configurable archiving policies, this online repository is periodically archived to an offline data store.

Conclusion

BEA WebLogic Integration closes the gaps between development and integration, it staff and business user, and EAI and B2B by providing a unified architecture. It reduces the learning curve by unifying the development skill sets needed by all users. In bridging integration environments that were previously isolated technology islands, WebLogic Integration is itself a bridge to reduced development costs, better use of organizational talent and knowledge, and increased productivity.

The New BEA WebLogic Workshop 8.1

SETTING THE STANDARD FOR A DEVELOPMENT FRAMEWORK

BY **CARL SJOGREEN**

AUTHOR BIO...

Carl Sjogreen is senior product manager for BEA WebLogic Workshop, an integrated development framework that makes it easy for all developers – not just J2EE experts – to build enterprise class applications on the WebLogic platform. Carl has been involved with XML, Web services, and developer tools since 1998, when he founded Transformis, developers of the award-winning Stylus Studio IDE.

CONTACT...

carls@bea.com

The initial focus of Weblogic Workshop was on Web services applications, but the core mission of the Workshop team has always been to deliver unprecedented productivity building enterprise-class applications. Many of the innovations introduced in the first version, such as visual designers, controls to simplify access to resources, and declarative annotations in Java code, apply to many applications, not just Web services.

In this new release of Workshop, BEA has dramatically expanded the kinds of applications you can build within the Workshop environment, and significantly enhanced the basic capabilities of the Workshop IDE.

Workshop 8.1 will be available in two editions: WebLogic Workshop Application Developer Edition and WebLogic Workshop Platform Edition. WLW Application Developer Edition includes the basic features targeted at application developers and supports building Web services, Web applications, and custom controls. WLW Platform Edition includes additional extensions to the IDE and runtime framework that let you build portal applications and workflows in conjunction with our Integration and Portal Server products. This article introduces the new features in WebLogic Workshop Application Developer Edition, and the core features of Workshop that are shared across the other BEA WebLogic Platform products.

Based on customer feedback, we focused our efforts around five major initiatives:

- Provide a unified development experience for the entire BEA platform
- Do for Web applications what Workshop 7 did for Web services
- Enable the development of custom controls
- Continue and expand leadership in Web services
- Dramatically improve basic IDE functionality

Unified Development Experience for the BEA Platform

One of the major themes over the past year at BEA has been the convergence of our product lines. Our customers have told us time and time again that they no longer want to buy separate products to build portals or to integrate their systems or run their Web applications. With the introduction of WebLogic Platform 7, we took the first steps toward bringing the entire BEA product set closer together. With this release, Workshop now provides a completely unified developer experience for the entire platform. This includes a common tool and runtime framework for the platform, as well as a shared programming model across application types.

Unified Architecture

Although the Workshop IDE is perhaps the most prominent portion of the product, it's important to realize that Workshop is both a tool and runtime framework, both of which are now utilized by BEA Portal and Integration server.

The Workshop IDE is a Swing-based Java development tool that provides many of the capabilities developers have come to expect from a development environment – great debugging,

source editing, project management features, etc. Moreover, the Workshop IDE provides visual editors and designers to enable a drag-and-drop development experience for most applications. The IDE's principal mission in life is to create files containing application logic that can then be executed by the runtime framework.

The Workshop runtime framework is a standard J2EE application that runs on top of WebLogic server. The Workshop runtime is responsible for handling the details of compilation, deployment, test harness generation, etc. The framework automates all the plumbing details and complex programming of building a J2EE application. Developers work with simple Java classes, and the framework automatically generates standard EJB components, message queues, databases, etc. This notion of using annotations in Java code to specify additional capabilities is being standardized via the Java Community Process in JSR 175.

The platform development experience is converging around both the design time and runtime components of the architecture. In the IDE, the tools to build portals and business processes are now fully integrated within the Workshop development environment. Common windows in the IDE that manage your project's files and folders, palettes that list available controls, and common gestures are shared across designers for each of the products. All designers share the familiar "design" and "source" views of the application, and feature full two-way editing so that changes made in one view are immediately reflected in the other. The Workshop runtime supports the business logic developed for these applications, and generates a common set of J2EE components for applications and controls that are automatically deployed to WebLogic server. Most important, however, it provides a unified programming model across the application types.

Programming Model

What does it really mean to have a unified programming model across the BEA platform?

A programming model is the "programmer's UI" – the way a developer builds an application and how he or she interacts with the development environment. What APIs are available? What is the architecture and the process for constructing an application? What are the basic components, and how do I interact with them?

In Workshop, the programming model is based on the notion of controls – simple components that have methods, events, and properties with a visual representation in the design view – and annotated Java code that lets you declaratively specify behavior and focus on handling events and calling methods, instead of writing complex object-oriented infrastructure code. This lets developers focus on writing the application logic that is important to them – the code that really builds the application – and lets the framework handle the details of the plumbing.

The Workshop programming model was introduced in Workshop 7 with Java Web Service (JWS) files and controls. Controls enabled you to easily connect to databases, message queues, and EJB components without having to learn the details of the J2EE APIs, and configure settings by setting properties (instead of making API calls). Similarly, when building an asynchronous Web service, instead of writing the code to handle message correlation and state management manually each time, you could simply set a property on a service and the Workshop framework would automatically take care of these details.

With Workshop 8.1, applications – whether they are Web services, Web applications, portals, or workflows – share a common process of assembling controls that encapsulate business logic or resources with additional code, workflow logic, personalization information, etc. All platform products share the notion of simple Java classes with declarative properties as the primary way application logic is built.

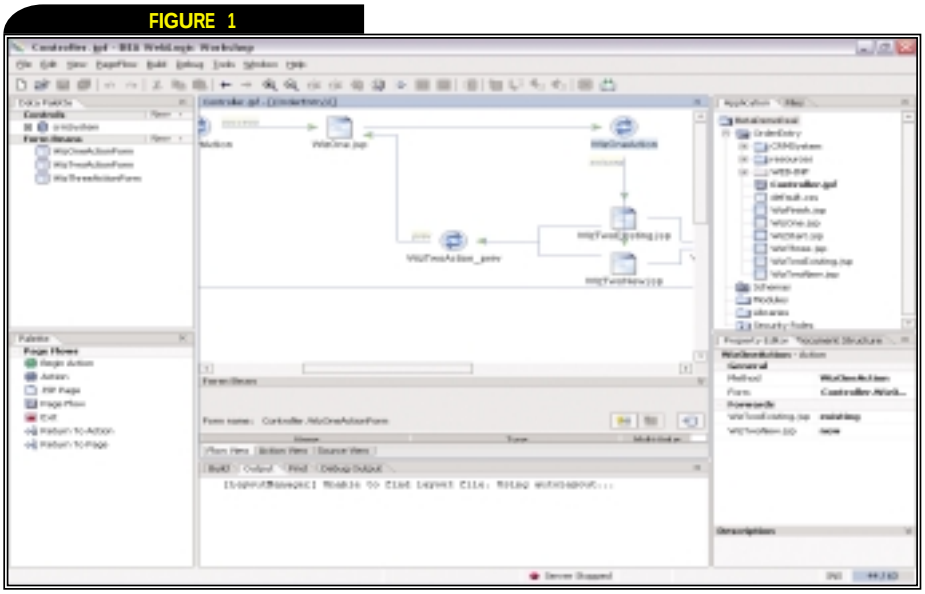
This level of integration delivers tremendous value – much more than just having all the products use the same tool. With a common application model shared across all products, controls can be leveraged across application types; common paradigms are used for accessing databases, Web services, etc. Moreover, skills learned in building one type of application can be leveraged to learn new products. This shared programming model, supported by the runtime framework and exposed via the IDE, is a major enabling factor for the 10x productivity gain that Workshop delivers.

Do for Web Applications What Workshop 7 Did for Web Services

Probably the most exciting new area of functionality in WebLogic Workshop Application Developer Edition is the addition of new features for building Web applications. Workshop 8.1 extends the programming model into the realm of Web applications. Leveraging the Model, View, Controller (MVC) paradigm for building Web applications, constructing a Web application involves constructing a set of standard JavaServer Pages (JSPs) for each of the pages (the views) in an application, writing the business logic, data, and navigation flow in a Pageflow file (the controller), and modeling the data that flows between pages and forms in an application in the form of JavaBeans (the model).

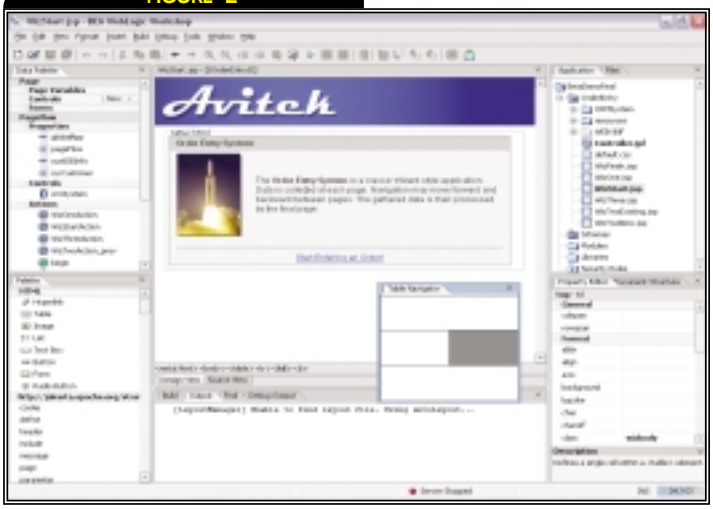
For those of you familiar with the Struts framework, you'll find this approach very familiar – in fact, Pageflow files are actually compiled into a set of Struts classes and the Struts runtime is responsible for the execution of the completed application. This enables developers to take advantage of a simplified programming model and rich IDE support, but still build cross-plat-

FIGURE 1



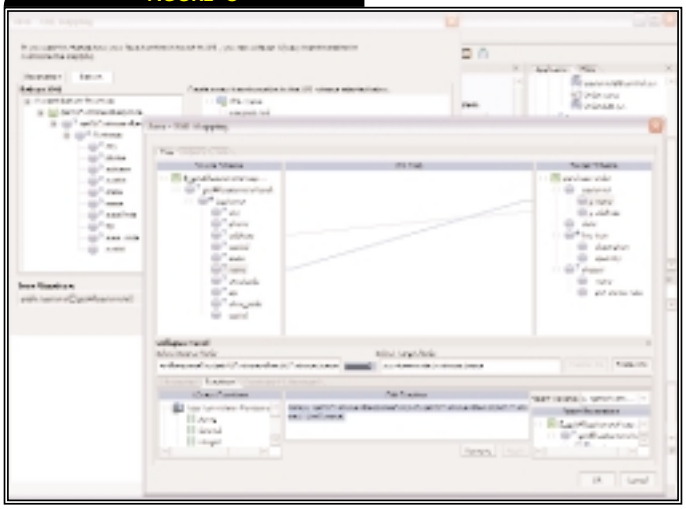
New IDE & Pagegroup editor

FIGURE 2



JSP Design View

FIGURE 3



XQuery Mapper

form applications that run on the open-source Struts framework.

PAGEFLOW

The Pageflow file is the central dashboard for a Workshop Web application. In design mode, a Pageflow provides a visual overview of an application, showing the pages and the "actions" or business logic that they invoke. The Pageflow editor also helps create and manage the data that flows between pages and actions. The design surface gives a quick visual overview of the application and navigation flow (see Figure 1). The source of a Pageflow file is a standard Java class that contains methods that correspond to actions that can be performed in the Web application. An example of a simple Helloworld Pageflow application is included in the source code for this article (HelloForm.jsp, HelloFlow.pfl, HelloResult.jsp; the source code for this article is available on the Web at www.sys-con.com/weblogic/sourcec.cfm). You'll see from this example that it's very easy to construct data-bound forms that are tied to simple Java code. What you can't see just by looking at the code is how easy an application like this is to construct within the IDE. Workshop now provides a set of wizards that makes it incredibly easy to construct forms and result pages via simple drag and drop operations.

Pageflows can also leverage controls to access prepackaged business logic, read data from a database, invoke a Web service, or call any other available control. In fact, IDE wizards can automatically construct a Pageflow and set of JSP pages on top of any control to immediately create an application that performs simple CRUD (Create, Read, Update, Delete)-style operations on a database or invokes a Web service. These pages and the navigation flow can then be further customized and extended into a complete application.

JSP EDITING AND DATABINDING

Easily building dynamic Web applications involves more than just stringing pages together with business logic, however. It's important to be able to easily construct the JSP pages themselves, and bind those pages to dynamic data. Workshop 8.1 features a complete, two-way, WYSIWYG JSP and HTML editor that is fully integrated within the IDE. Using this editor you can construct JSP pages by dragging HTML and JSP tags from the shared palette of resources into the design surface and visually setting properties. The data palette provides full access to all of the data in the Pageflow, so you can create new forms to take data as input or pages to display processing results with one drag and drop. This is accomplished via a rich set of custom JSP tags that can display tabular information, lists, and trees, with built-in support for filtering and querying database information (see Figure 2).

Enable the Development of Custom Controls

When Workshop introduced the concept of Java controls as a simplified way to access enterprise resources, we were immediately asked to make this model extensible so that customers and ISVs could build their own controls to plug into the Workshop framework. With Workshop 8.1, using familiar visual designers you can easily build tightly coupled business logic components that support methods and events as well as asynchronous invocation. Simply by setting a property, Workshop will automatically generate and deploy a JMS queue to support reliable processing of asynchronous messaging. You build controls by defining their interface (what methods and events they support) on the left side of the designer, setting properties to indicate asynchronous messaging, security restrictions, and the like, and then write business logic to implement these methods.

Controls can use other built-in or custom controls (shown on the right) in infinite levels of nesting so it becomes very easy to repackage and reuse these components. Control authors can also specify custom properties for their control via a simple XML syntax that are immediately supported within the IDE property editor. For example, a control that accesses an enterprise application might have a username and password property, or properties to configure the interaction with that system. Once controls have been built, they can be packaged as redistributable .jar files and easily added to other projects within your organization, or distributed externally. Controls will automatically appear on the palette of available controls as soon as the .jar file is installed.

Advanced control authors can even build custom property editors and wizards that appear within the IDE to automate control setup. More information on these advanced control features can be found in the Control Developer Kit distributed with WebLogic Workshop.

Continue and Expand Leadership in Web Services

Don't worry, we haven't forgotten about Web services! Workshop 8.1 builds upon a solid foundation for enterprise-class Web services that provides native support for loosely coupled interaction, asynchronous

"conversational" messaging between parties, and rich business documents. It adds a variety of new enhancements around these three design principles, and extends the Web services stack to support key additional enterprise requirements.

Addressing the security and reliability concerns about Web services is crucial to enterprise adoption. To meet this need, Workshop now has full support for the WS-Security specification for message-level security (digital signatures and encryption) and has added a set of properties to the JWS file format to easily support declarative role-based security that leverages the underlying WebLogic Server. In addition, Workshop now supports "exactly once" messaging, which ensures that even over protocols like HTTP messages are reliably delivered once and only once to the recipient. Reliable messaging is exposed as a property that can be set on a message.

One of the major design goals of Workshop 7 was to enable the development of loosely coupled Web services. To BEA, just having XML on the wire isn't loosely coupled enough – you need a mechanism that enables easy mapping between specific XML Schemas and Java objects so that both the XML and Java code can change independently without breaking users of your Web service or requiring major coding changes. We solved this problem with a simple, declarative mapping language. In Workshop 8.1, we've moved to XQuery as the standard language for binding XML to Java code and now provide a complete visual editor for performing these mappings. This makes it drop-dead simple to build a Web service from a set of messages (defined either as XML Schemas or example documents) and then map fields into individual Java parameters (see Figure 3).

In other cases, however, fully decoupling XML from Java code using a transformation might be overkill. Sometimes it's easier just to access information from the XML directly, or save it away and potentially perform a transformation later. To enable this scenario, Workshop has introduced XML Beans – a unique new technology that dramatically increases the productivity of accessing XML from Java.

Traditionally, developers have had to choose between low-level APIs such as SAX or DOM that enable direct access to XML but are tedious to work with, and Java binding solutions such as JAXB that provide a convenient set of Java interfaces but lose structural information from the original XML. XML Beans provide the best of both worlds. Using an efficient representation of the original XML document, XML Beans provide a simple cursor-based API for direct navigation through an XML document, an XQuery interface for retrieving information, and a set of Java class "views" on the underlying XML data. Given an XML Schema description, the XML Beans engine automatically generates a set of Java types that enable read-and-write access to the XML. What's different here is that XML Beans are based on XML Schema from the ground up, so there is no schema that cannot be bound to Java types (other solutions support about 50% of schema) and the fact that these Java classes are simply views on the underlying XML. No data is ever lost (even things like comments!) because the original XML is always retained. To use XML Beans, simply add a schema file to your Workshop project, and all the Java types will automatically be generated.

Dramatically Improve Basic IDE Functionality

Although the Workshop IDE continues to be focused on developers interested in leveraging the Workshop framework, and not for J2EE development in general (where we recommend Borland JBuilder, WebLogic Edition), many of the basic editing, debugging, and management features of leading IDEs are important to all developers, and we've made great strides in increasing the sophistication of the Workshop IDE itself. The new features here are too many to list, but I've included a few of the highlights.

BASIC IDE WINDOWING

The IDE look and feel has been updated significantly and now fully supports customizable window layouts. It is organized as a set of supporting windows that can be docked and undocked, and a central document editor with design and source views.

DEBUGGING

We've significantly enhanced the debugging engine in Workshop to support JSP debugging and cross-component debugging. This means you can step from JSP code into Pageflow code into code inside custom controls that you have written. Moreover, the debugging framework is leveraged by the other platform products, so you can even debug from a business process workflow into custom control code or JSPs. The JSP debugger lets you step through the execution of an individual JSP page and view the output stream as it is being constructed. Debugging performance has also increased significantly, and immediate mode and multithreaded debugging support have been added.

New Project Model

Workshop applications now correspond directly to J2EE applications and are packaged into EAR files for deployment. A single project model that can contain Web applications, libraries, custom controls, portals, and workflows enables you to build applications that cross product boundaries and easily manage and deploy them as one unit. The project model is also fully integrated with leading source code control systems such as CVS and Perforce so that you can check in and check out files directly from within the IDE.

SOURCE EDITING FEATURES

We've also added support within the source editor for many popular productivity features. For example, when you use a class that has not yet been imported into your code, Workshop will automatically prompt to import that library. Holding down the CTRL key and hovering over functions and variables provides additional information and a link to jump to the definition or declaration. Basic source editing features like auto completion have also been enhanced and expanded beyond Java functions and variables. For example, any "href" tag in a JSP or HTML page will autocomplete to any file or image in the project.

Standards

BEA has always been committed to developing standards-based technologies and these innovations in the Workshop environment are no exception. The Web services capabilities introduced with Workshop 7 are well on their way toward standardization via the Java Community Process in the form of JSR 181. New features for building Web applications fully leverage Struts, an open-source and cross-platform runtime, and we will be announcing new JSR and community initiatives around technologies like XML Beans, Controls, and Pageflow annotations.

Conclusion

As you'll see, this release sets the standard for an integrated, easy-to-use development framework for a J2EE platform. You'll find it easy to get started if you haven't worked with J2EE in the past, and your productivity will improve dramatically even if you already know J2EE. Workshop doesn't just make it easy, though – it has been architected from the ground up to fully leverage the power of WebLogic Server and support application architectures that are secure, fast, and scaleable.

Continued on page 25

WEBLOGIC SER

ENHANCEMENTS THAT CAN MOVE YOU
INCREASINGLY COMPETITIVE WORLD

BY **JIM RIVERA**

AUTHOR BIO...

Jim Rivera is a product manager for WebLogic Server. He has been with BEA Systems since 1999, and has over 10 years of experience in information technology and electrical engineering.

CONTACT...

jim.rivera@bea.com

In today's competitive business environment, success depends on the quality of information available and the sophistication with which it is used. Many companies successfully differentiate themselves with the quality of their information-handling systems. As technologists, our job is to build systems that fully leverage the information available within the enterprise. It's critical that these systems adapt seamlessly and efficiently to ever-changing business requirements.

In spite of the relative importance of information systems, the current economic conditions have had an enormous impact on Information Technology (IT). While the need for improved information systems has not gone away, the resources allotted for such projects have been reduced, in some cases dramatically. We're all faced with the considerable challenge of doing more with less, finding ways to work with greater efficiency. Building and maintaining custom enterprise application systems is a difficult proposition. Even with the many tools available on the market, the complexity associated with developing and managing custom applications can be a drain on our resources. This wastes time, money, and perhaps most importantly, opportunity. I've heard some customers say, "We have stopped all new development projects because we can hardly afford to maintain the applications that are already deployed." More than ever, reducing the total cost of ownership of information systems has become a necessity.

At the same time, application server vendors are faced with an increasingly competitive market. In many cases, vendors have responded by adding features, the latest bells and whistles intended to differentiate themselves from their competition. While additional features are intended to improve a product, they often make a product more difficult to understand and use. Elegance and simplicity give way to feature "bloat" and complexity. The vast majority of commercially available software products have grown in complexity with each subsequent release. For example, think of some of the desktop applications that we love to hate – word processors, spreadsheet applications, and the like, that add new features we rarely use, obscuring the few features that we do use on a daily basis. This is not to say that new features are bad. They are often necessary to address new or changing requirements. However, it is critical that new features reduce, not add to, overall complexity. With WebLogic Server 8.1, we have focused on improvements that simplify interaction with WebLogic Server, with the end goal of raising productivity and reducing costs.

Maximizing Productivity

BEA WebLogic Platform 8.1 reduces complexity and maximizes productivity by providing a *unified* framework for all core application infrastructure components. Common development tools, install, and configuration for all WebLogic Platform products *simplify* user interaction. When necessary, WebLogic Platform can also be *extended* to seamlessly incorporate custom com-

ponents or third-party, best-of-breed products.

WebLogic Server is the foundation for WebLogic Platform and provides the kernel of services for the other components. For WebLogic Server 8.1, we have paid particular attention to how various user roles interact with WebLogic Server with the goal of maximizing productivity and reducing costs in the following scenarios:

- ***Developing applications:*** Maximize productivity with simplified programming models optimized for iterative development cycles.
- ***Administering applications***: Reduce the ongoing costs of operations, and maximize uptime, reliability, performance, and scalability.
- ***Leverage existing IT investments:*** Get the most out of existing systems by integrating applications using standards-based integration technology such as Web services and messaging.

In the following sections we'll take a closer look at how WebLogic Server 8.1 can reduce your costs and help you do more with less.

Developing Applications

Different development tasks require different skills sets. Because of this, a certain amount of specialization is expected in every IT organization. You wouldn't ask a Web jockey to develop a highly optimized data access layer based on Entity EJBs, nor would you ask an integration expert to design a Web-based user interface. As developers specialize around different tasks and skills sets, they also develop different practices and use different programming models, patterns, and tools. It is therefore critical to choose a platform that maximizes developer productivity regardless of sophistication level or specialization. WebLogic Server 8.1 provides an environment that allows developers to use their preferred programming paradigms and work within a level of complexity that they are most comfortable.

Historically, the complexity of developing advanced J2EE applications has limited adoption to only the most sophisticated developers with a good understanding of distributed computing and object-oriented programming. It can be very difficult and expensive to build a large staff of developers with this level of sophistication. WebLogic Workshop provides a unique IDE and framework that greatly simplifies development of WebLogic Server applications. Developers work within a higher level of programming abstraction that shields them from the complexities of J2EE and object-oriented programming. With WebLogic Platform 8.1, WebLogic Workshop can be used to create Web services, Web-based user interfaces, business process workflows, and even EJB applications.

Experienced enterprise-level J2EE developers may also choose to use a J2EE development environment. Borland JBuilder, BEA WebLogic Edition, is a tightly integrated development solution for building enterprise applications on WebLogic Platform. It is optimized for seamless development and deployment to the WebLogic Platform, enhancing developer productivity with support for UML code visualization, refactoring, and unit testing.

Of course, we haven't forgotten about those developers who prefer to stick with the flexibility and low overhead of Emacs or vi and custom build environments. While these environments can be fast and efficient to work with, they often require significant effort to create. In WebLogic Server 8.1, we continue to embrace the Ant build tool and have provided custom Ant tasks for handling the repetitive tasks of building, configuring, and deploying WebLogic Server applications. The provided utilities are extremely efficient, avoid all unnecessary copying and archiving of files during the build process, and maintain a clear distinction between source files and output files for simple integration with source-control systems.

Regardless of the development tools you choose, developer productivity relies on the ability to quickly apply changes to an application and verify the intended effect of those changes. In WebLogic Server 8.1, we paid particular attention to optimizing for iterative development cycles. We made significant improvements to minimize the time between when a change is applied to source code and when that change can be verified in a deployed application. These changes include minimizing the need to restart the server for configuration changes, optimizing application redeployment so that only the delta is redeployed, significantly improving server startup time for those cases where a re-start is necessary, and consolidating component compiler tools to a single utility that can handle all application component types. The end result is an extremely productive development experience for all developers.

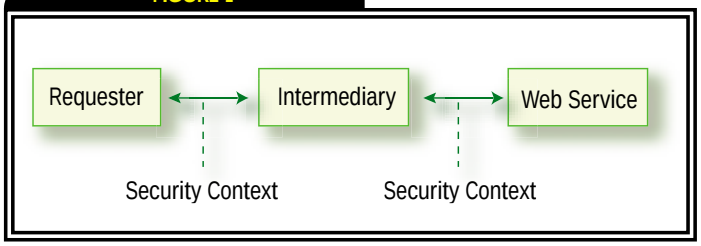
Administering Applications

With the success of J2EE and WebLogic Server, the number and scale of deployed applications within the enterprise have increased dramatically, leading to increased technical and organizational complexity in application management. With the increased focus on the bottom line, the cost of administering existing applications has crippled many IT organizations. WebLogic Server 8.1 increases administrator productivity and reduces costs, with significant enhancements to application monitoring, analytics, and configuration as well as overall server performance and scalability.

In WebLogic Server 8.1, BEA introduces significant new functionality in the area of application monitoring and analytics for fault diagnosis and troubleshooting. Intended for use in development, test, and production environments, this functionality provides monitoring visibility of all J2EE components and WebLogic Server services to help developers and operators understand the runtime characteristics of their code, and quickly isolate performance and logic problems. And because all data visualization, monitoring configuration, and rules creation are presented to users in the context of the WebLogic Server Administration Console, this functionality is accessible to any authorized user without the need for separate installation or third-party tools.

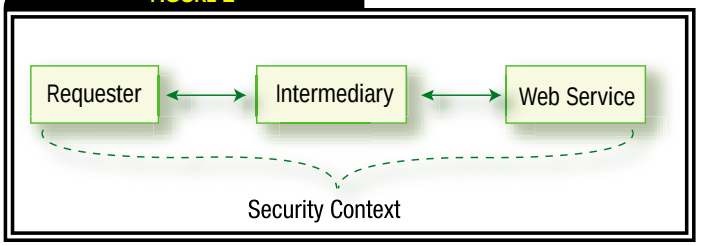
WebLogic Server monitoring auto-discovers and instruments user applications to provide the depth of visibility needed to

FIGURE 1



Point-to-point security using SSL

FIGURE 2



End-to-end security using WS-Security

support troubleshooters as they drill from service, to object, all the way down to individual methods executed at runtime. If developers have exposed custom metrics using JMX, WebLogic Server monitoring will recognize and expose these as well.

Users can define thresholds and rules that throw alerts when violations occur, and, because all WebLogic Server monitoring data is exposed to third-party tools, it is possible to automate alerts and other operations-level logic in third-party enterprise-management tools from HP, BMC, and Mercury Interactive (to name a few). This provides the flexibility to integrate WebLogic Server into any support infrastructure, helping to lower costs of support, and improve the availability of online products.

Application configuration has also gotten easier. Improvements to the Administration Console greatly simplify some of the more complex configuration tasks such as SSL, JDBC Pool, and cluster replication group configuration. For repetitive tasks, such as migrating applications from development environments to testing/staging, administrative commands can be scripted using batched mode support with the administrative utilities as well as new Ant tasks that provide simplified abstractions to all WebLogic Server administrative commands.

Of course, server reliability, availability, scalability, and performance provide obvious benefits in terms of application uptime, delivering the best possible end-user experiences, and meeting the most demanding application requirements in terms of scope and scale. WebLogic Server 8.1 continues its tradition of excellence in all of these areas. We have paid particular attention to improving scalability with new load balancing options, and performance to maximize the computing power of each server instance, allowing for the most efficient use of software license and hardware resources. Couple this with dramatic improvements in administrator productivity resulting from the enhancements in application monitoring, analytics, and configuration described above and WebLogic Server 8.1 can help to significantly drive down the cost of managing your applications.

Leveraging Existing IT Investments

With the massive investments in enterprise software applications over the last few decades, today's typical enterprise is made up of a vast number of disparate systems purchased and implemented throughout various stages of software evolution. These systems are implemented on a variety of software and hardware platforms without a standard protocol or data model for communication between applications. It is no secret that the company that can best leverage its existing IT investment to create unified systems that share data and services will have an advantage over its competitors. In the past, the lack of standards-based integration technology required any integration effort to be based on proprietary software solutions. The resources required to develop and maintain proprietary solutions come at a far greater expense than standards-based solutions that can leverage existing skill sets and prebuilt components and adaptors. With the emergence of Web services, J2EE Connector Architecture, and JMS-based messaging products, standards-based integration is becoming a reality. As these technologies mature, greater cost savings can be realized as the technology's applicability to various integration scenarios increases. WebLogic Server 8.1 provides two critical components that are lacking in many Web services platforms on the market today: end-to-end security and reliable delivery.

End-to-End Security

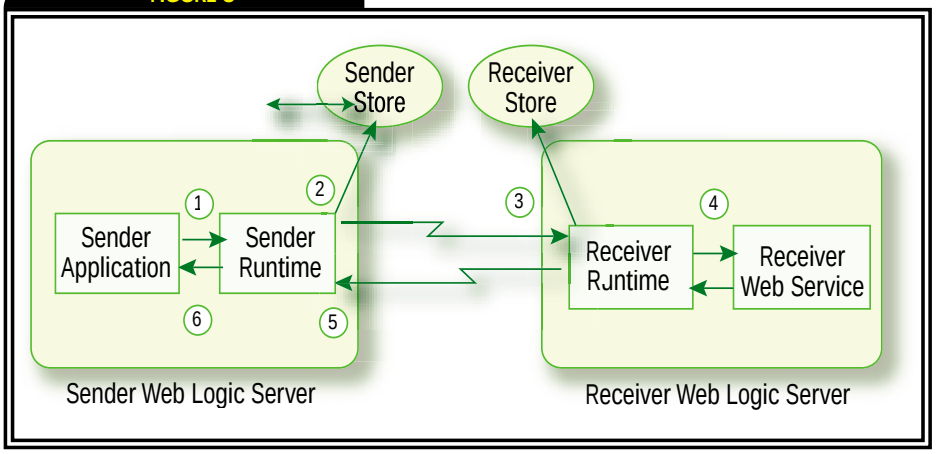
In many integration scenarios, it is critical to secure the integrity and confidentiality of the messages and to ensure that the service acts only on messages that express the claims required by policies. Since Web services typically leverage HTTP as the transport protocol for sending SOAP messages to service endpoints, it's possible to provide point-to-point, or transport level, security using the Secure Socket Layer (SSL; see Figure 1).

However, SSL alone is insufficient for many common topologies that require message-processing intermediaries. When messages are received and processed by an intermediary, they must be pulled off the transport layer. When the message is then forwarded, both the integrity of data and any security information that flows with it may be lost. This forces any subsequent message processors to rely on the security evaluations made by previous intermediaries and to trust their handling of the message content. WebLogic Server 8.1 is the first application server to provide an implementation of WS-Security, an OASIS specification (currently in draft). WS-Security solves these issues by providing facilities to ensure data integrity and confidentiality at the message level. XML Encryption and XML Signature can be applied directly to the SOAP message. Therefore, even in multi-hop scenarios, where a message must pass through multiple intermediaries and be pulled off the transport layer, the integrity and confidentiality of the message can be maintained (see Figure 2).

Reliable Delivery

Asynchronous, reliable delivery is another crucial component of many integration projects. This can be a challenge for Web services architectures that extend beyond the firewall and need to use HTTP as the transport protocol because HTTP is inherently synchronous and unreliable. Reliable messaging is a new framework for WebLogic Server 8.1, whereby an application running in one WebLogic Server instance can asynchronously and reliably invoke a Web service running on another WebLogic Server instance. The reliable messaging feature works in a way similar to ebXML Reliable Messaging protocol, only the concepts have been applied to Web services. The architecture is described below and in Figure 3.

FIGURE 3



Reliable Messaging Architecture

1. The sender application invokes a reliable operation running on the receiver WebLogic Server.
2. The sender runtime saves the message in its persistent JMS store. The store can be either a JMS File or a JDBC store.
3. The sender runtime sends the SOAP message to the receiver WebLogic Server.
4. The receiver runtime receives the message, checks for duplicates in its persistent JMS store and, if none are found, saves the message ID in store. If it finds a duplicate, the receiver ignores the message.
5. The receiver runtime immediately sends notification back to the sender that the message was received.
6. The receiver runtime invokes the reliable operation.
7. Because only void operations can be invoked reliably, the receiver does not return any values or exceptions to the sender.
8. The sender runtime removes the message from its persistent store so that the message is not sent again.
9. The sender is configured to retry sending the message if it does not receive notification of receipt. You configure the number of retries, and amount of time between retries, of the sender using the Administration Console. Once sender runtime has re-sent the message the maximum number of retries, it removes the message from its store.
10. The sender runtime sends notification to the sender application (either via callbacks or polling) that either the message was received or that it was never successfully delivered.

For messages within the firewall, it is often acceptable to use the Java Message Service (JMS) or a message broker as a way to reliably deliver messages from one node to another. In WebLogic Server 8.1, WebLogic Server JMS can be selected as the transport protocol (as an alternative to HTTP/S) for SOAP messages sent to a given Web service. The generated WSDL of the Web service will contain a port definition with a JMS binding.

Additional enhancements to WebLogic Server JMS in WebLogic Server 8.1 include:

- A thin JMS client that provides full client functionality at greatly reduced client-side footprint (400K)
- Simplified configuration for integrating with foreign JMS providers
- New "wrappers" that make it easier to use JMS from inside J2EE components such as EJBs and servlets.
- Improved handling of expired messages to ensure that they are cleaned up immediately. Moreover, message auditing gives you the option of tracking expired messages, either by logging when a message expires or by redirecting expired messages to a special destination.
- A "Blocking Send" feature to help you avoid receiving message quota errors by temporarily blocking message producers from sending messages to a destination when the destination has exceeded its specified maximum message quota.

By addressing the crucial elements of end-to-end security and reliable delivery, WebLogic Server 8.1 greatly increases the applicability of Web services. This allows you to leverage your existing IT investment in a cost-effective manner without relying on costly one-off solutions that are difficult to develop and maintain. For integration projects that require higher-level functionality, such as Business Process Management (BPM), B2B, and complex transformations, WebLogic Integration 8.1 is a natural extension on top of the base functionality offered in WebLogic Server 8.1.

Conclusion

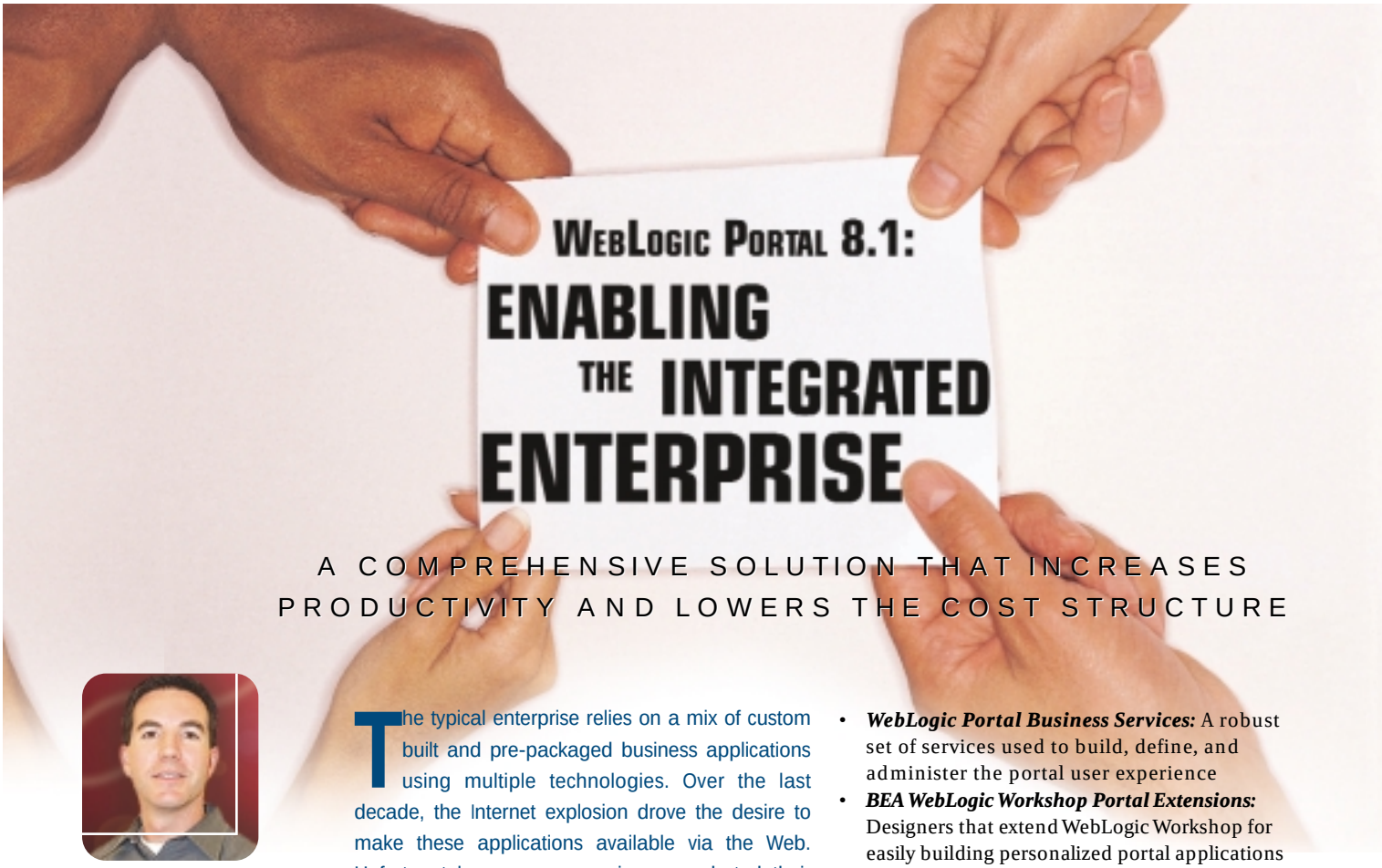
In the face of economic uncertainty, IT organizations are forced to reduce costs and make the most out of available resources. It has become necessary to maximize developer and administrator productivity and fully leverage existing IT investments. WebLogic Server 8.1 provides the development environments, administrative tools, and integration technology that can help you get ahead in an increasingly competitive world without damaging your bottom line.

continued from page 21

The combination of ease of use, tight integration, and a powerful platform is unbeatable.

Get it Today!

To best experience what Weblogic Workshop 8 is, try it yourself. A beta release of Workshop is available as part of the new developer subscription program. This gives you free access to all BEA developer products for one year and provides a low-cost, "pay for support" program with quarterly updates. You can get more information on downloading Workshop and the subscription program at www.bea.com. Also be sure to check out http://dev2dev.bea.com for additional developer resources, technical articles, code examples, and how-to tips. I'd also love to hear your feedback and suggestions for the product.



WebLogic Portal 8.1: ENABLING THE INTEGRATED ENTERPRISE

A COMPREHENSIVE SOLUTION THAT INCREASES PRODUCTIVITY AND LOWERS THE COST STRUCTURE

BY **SHANE PEARSON**

AUTHOR BIO...

Shane Pearson is the director of Product Management for BEA's WebLogic Portal and manages the team responsible for its product strategy and design. This includes coordinating the definition of product requirements based on input from customers and partners to ensure the successful implementation of a competitive and innovative product. During his tenure, several releases of WebLogic Portal and earlier products have been recognized as innovative products and received industry awards.

CONTACT...

shane.pearson@bea.com

The typical enterprise relies on a mix of custom built and pre-packaged business applications using multiple technologies. Over the last decade, the Internet explosion drove the desire to make these applications available via the Web. Unfortunately, many companies exacerbated their existing IT fragmentation problem by developing 10s or 100s of Web sites to surface their silos of data and applications to users both internal and external to the enterprise.

With this proliferation of Web sites, portal solutions initially provided a way for companies to provide visibility to content. These early solutions aggregated content and applications, but did not provide integration of applications, support for custom application development, or the ability to interact with and expose business processes.

Today, evolving business requirements are pushing next-generation portal solutions to meet the ever-increasing needs of the enterprise to provide integrated and composite views of applications and data while providing the ability to enforce business policies and processes in order to increase user productivity and utilization of corporate assets.

This article provides an overview of the functionality and design of the latest release of BEA WebLogic Portal – an ideal solution for your current and future portal needs.

- ***WebLogic Portal Framework Services:*** A flexible framework for defining the way applications, content, and business processes are presented to various portal audiences
- ***WebLogic Portal Business Services:*** A robust set of services used to build, define, and administer the portal user experience
- ***BEA WebLogic Workshop Portal Extensions:*** Designers that extend WebLogic Workshop for easily building personalized portal applications
- ***BEA WebLogic Portal Administration Tools:*** An extensible administration environment for managing multiple portal Web applications, portals, and portal resources for an individual department or a portal network that spans an enterprise

Portal Development Life Cycle

Common portal projects include sales force portals, intranet portals, customer self-service portals, or business partner portals. In contrast to other portal products, WebLogic Portal provides several entry points that make it an ideal solution whether the project initially requires a single application, delivers content-centric portals, or requires making transaction-based applications available to support business-to-consumer or business-to-business portals.

WebLogic Portal offers functionality and tools tailored to specific users, including application developers, JSP developers, HTML and graphic designers, system administrators, portal administrators, and business analysts. The following functions assist the creation of portals and portal resources and in many cases do not require J2EE programming.

Portal Resources and Application Development

Application developers using standards-based technologies can create portal

resources. HTML and graphics designers can easily define new look-and-feel specifications and portal page layouts with traditional page design tools. JSP developers can customize the default portal presentation and layout templates to accommodate sophisticated user interface requirements and to add fine-grained personalization.

JSP developers can also create new applications and expose them as portlets. In addition to JSP- or HTML-based portlets, developers can create more sophisticated applications that leverage Web services, Pageflow, or available J2EE resources, using WebLogic Workshop.

Portal Development

Portals can be created in the same WebLogic Workshop developer environment that supports application development by installing the WebLogic Workshop Portal Extensions.

- ***Portal Designer:*** Allows easy definition and configuration of the portal layout, available pages, booklets and portlets, and the portal look and feel
- ***Portlet Designer:*** Allows developers to build several types of portlets (JSP/HTML, Java API, Pageflows, or Web services; see Figure 1)
- ***Portal Resources Designer:*** Allows developers to define content selectors, user profiles, events, user segments, placeholders, and other portal resources for use when building, testing, and deploying applications

Portal Administration

After the portal or portal resources are created and deployed, the portal characteristics can be managed using the browser-based portal administration tools. The following are a sample of the functions available in the portal administration console.

- ***User and group management:*** Manage user and group access to available portals, define delegated administration roles, and define visitor entitlement roles.
- ***Portal management:*** Manage portal resources, including portal desktops, pages, portlets, and layouts.
- ***Interaction management:*** Manage campaigns, placeholders, content selectors, and user segments that can be used to personalize the portal experience.
- ***Content management:*** Manage and define content schemas and repositories. Create and manage the content available to respective portals.
- ***Server management:*** Manage lower-level portal server settings such as cache management.
- ***Third-party plug-ins:*** The portal administration tools' extensibility allow for additional tools to be made available within the portal administration environment.

Portal End User Customization

Finally, as is customary for portals, portal visitors can customize their portal within the constraints imposed by administrators. They can add and remove portlets and pages and arrange portlets on a page, select the default start page, and select the look and feel and page layout.

Portal Framework Services

The portal framework is composed of several components that support the development of a flexible user interface to provide secure access to content, applications, and business processes. Several new components, including desktops and booklets, are introduced in this release of WebLogic Portal to add increased flexibility when defining and deploying portals and portal networks that require hierarchy and multiple levels of delegated administration.

Portal Anatomy

Several enhancements have been made to the technical architecture of WebLogic Portal to provide greater flexibility for building portals. In WebLogic Portal 8.1 the multiportal architecture is augmented to support multiple portals per Portal Web Application, as compared to the one-to-one relationship in WLP 7.0. This change provides more flexibility to developers when building portal applications with more complex hierarchies.

Portal Rendering

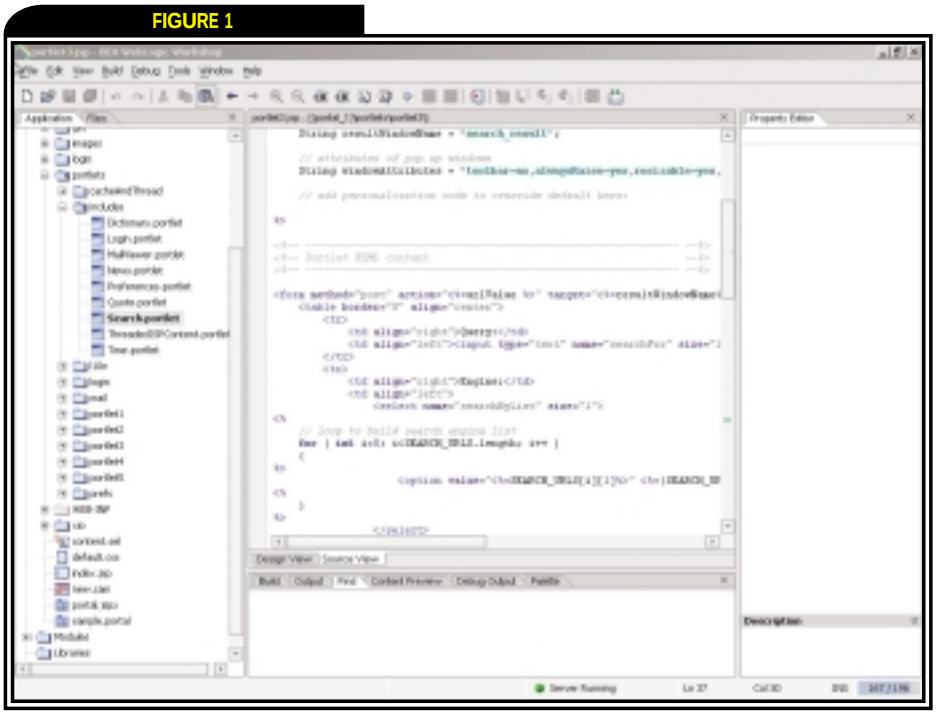
The main portal template is represented by an XML file. The portal file can be static (accessed directly via a browser – great for development) or dynamic (assembled from a relational database), based on business or customization needs for the portal. Each element in the portal file represents a portal control. These controls provide great flexibility in portal design by supporting the development of loosely coupled portal applications. In addition, this model provides a rich life cycle and supports portlet-to-portlet communication via events.

A developer may also override the functionality in the life-cycle methods via the control's *backing file*. A backing file is a simple Java class that implements the appropriate life-cycle methods. All life-cycle methods on the backing file are run before the control is rendered, allowing the developer to modify certain attributes or fire events before the portal is rendered.

Desktop

Portal Desktops represent a portal instance that is accessible by end users. Access to a desktop can be controlled by

FIGURE 1



WebLogic Workshop source view for portlet development

"Today, evolving business requirements are pushing next-generation portal solutions to meet the ever-increasing needs of the enterprise"

entitlements, which grant authorization to desktops based upon user roles. A desktop is the physical entity that is accessed by end users and is the successor to group portals that were used in earlier releases of WebLogic Portal.

The relationship between desktops and portals is illustrated in Figure 2, expanding upon the earlier example.

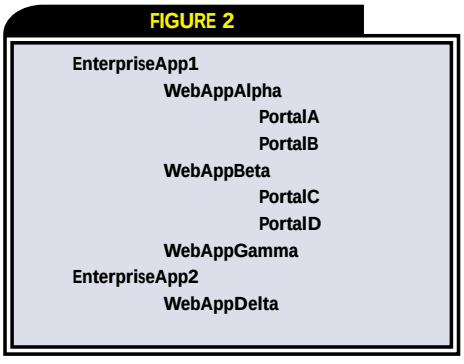
Booklet

A booklet is a container for one or more pages and/or booklets, and are typically associated with a navigational control for switching the currently displayed page(s). Booklets can be used to more easily create *n*-tiered navigation and are used to provide navigation options for the portal.

Page

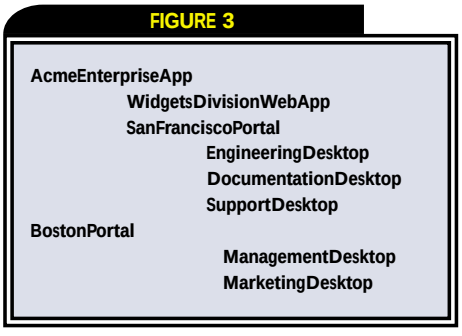
The main portal content area may contain one or many pages. Pages can be stacked in order to create a multitiered navigation model. By default, a tabbed navigation bar is used to quickly switch between pages by bringing a page to the front. Each page consists of one or more layouts, and contains portlets within a layout.

FIGURE 2



Relationship between portals built using WebLogic Portal and J2EE

FIGURE 3



Relationship between desktops and portals

Portlet

Web applications can be presented as one or several rectangular windows on a page or booklet. Portlets may have several views, such as the maximized or edit view. Portlet views are implemented using JSP and may range in complexity from simply including static HTML content, employing JSP tag libraries or WebLogic Workshop Java controls to access EJB components, or other application functionality within the Workshop Framework or WebLogic application server.

Several new features are supported in this release:

- ***Multithreaded portlets support:*** Allows for a more flexible and higher performing portal rendering model. A portlet property can be set in order to make the portlet multithreaded. This means that on rendering the portlet spawns an additional thread in order to render the portlet. This feature would most likely be used on portlets that contain applications, which for business reasons are not cached or require large amounts of processing.
- ***JSR 168 Portlet API will be fully supported:*** Allows WebLogic Portal developers to use the Java portlet as an implementation choice. There is no difference in how Java portlets appear and function to portal administrators or end users.

Portal Development Framework

BEA WebLogic Portal is implemented using the Java 2 Enterprise Edition (J2EE) architecture; it's a J2EE application that runs in the WebLogic Server environment. In J2EE terms, WebLogic Portal is an *enterprise application* that consists of a collection of Enterprise JavaBean (EJB) components and a set of *Web applications*, which in turn are collections of servlets, JavaServer Pages (JSPs), JSP tag libraries, and supporting Java classes.

WebLogic Portal 8.1 provides developers with a rich set of portal functionality exposed via JSP tag libraries and APIs, such as user and group profile, rules-based personalization, and content management. In this release, many of these features are also exposed as WebLogic Workshop Java controls that further increase developer productivity and make these features available to a larger developer audience.

Controls were introduced in the last release of WebLogic Workshop for use during application development. This release of WebLogic Workshop adds additional business controls that support application development in WebLogic Portal and WebLogic Integration.

A *control* is a component that you can incorporate into your Web service, JSP, Pageflow, or other application so that it can communicate with other kinds of applications and components. For example, a database control enables your application to request data from a database. A service control makes it easy to call another Web service.

Portal controls allow for specific portal functions to be more easily incorporated during application development by exposing the appropriate methods to the application developer. Control methods allow a developer to visually or programmatically invoke the functionality of the underlying resource without having to write the wire up to the resource.

The User Profile control is one example of a portal control. This control provides the business logic to retrieve and update user profile information. It allows a developer to determine whether a user exists, determine what user profiles exist, and retrieve a list of all users with the option to specify search parameters or obtain a list of all user profiles.

Application developers who require user information for custom applications can then use this control without having to first learn the lower-level WebLogic Portal APIs and JSP tags. While they could write the same application without leveraging this control, it would require them to write many more lines of code to make the appropriate calls to the underlying portal services.

Enterprise Integration

BEA WebLogic Portal supports several methods of enterprise integration in order to communicate with other applications and systems.

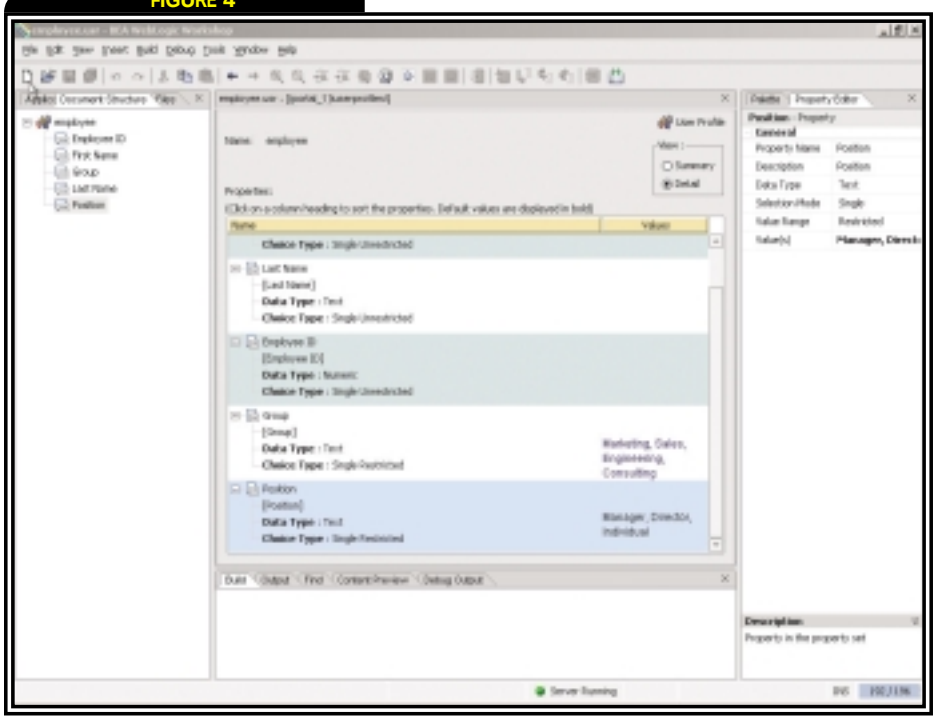
- ***Portlets:*** Portlets are applications, or views of applications, that exist within a portal. Different types of portlets are supported (JSP/HTML, Web service, Pageflow, or Java) and can be used by developers to aggregate or integrate data and applications from enterprise systems.

- ***Unified User Profile (UUP):*** The Unified User Profile in WebLogic Portal can be used to integrate with any external system to aggregate user profile information for use within WebLogic Portal applications. The UUP API can also be used by external systems to change, read, update, and delete data contained in the UUP (see Figure 4).
- ***Web Services:*** WebLogic Portal empowers business application developers to create, test, and deploy enterprise-class Web service applications within the portal environment. The simple visual development tools, available in WebLogic Workshop, make it easy to access enterprise resources and other Web services to support Web service and portlet development. Because WebLogic Portal tools are made available within WebLogic Workshop, your Web services are ready to deploy on a BEA WebLogic Portal within your organization.
- ***Pageflow:*** Pageflow represents the next generation of application flow technology from BEA and is the logical evolution of what was provided in previous releases as Webflow. Pageflow manages the flow of business logic and application presentation while allowing developers to use discrete units of business logic called Actions to execute focused processes within a session. The Pageflow Editor in WebLogic Workshop can be used to build applications that integrate data or business processes from external sources, while the Portlet Designer allows the developer to generate an appropriate portlet to make the application available to various portal audiences.
- ***Business Process Management and Application Integration:*** WebLogic Portal's flexible architecture supports the ability for portal-based applications to interact with J2EE Connector Architecture adapters and with business process management applications. In addition, WebLogic Portal is pre-integrated with WebLogic Integration to make it easy to expose WebLogic Integration application views and business processes within the portal environment, or to have portal events trigger long-running enterprise business processes.

Portal Business Services

BEA WebLogic Portal has at its foundation a core set of modular business services that support enterprise portal and application development. These services improve the developers' ability to build custom portals and for administrators to tailor the portal user experience. These flexible and extensible services are provided by BEA, but are also in many cases available to be extended by customers, independent software vendors, and solution implementation providers in order to provide best-of-breed solutions.

FIGURE 4

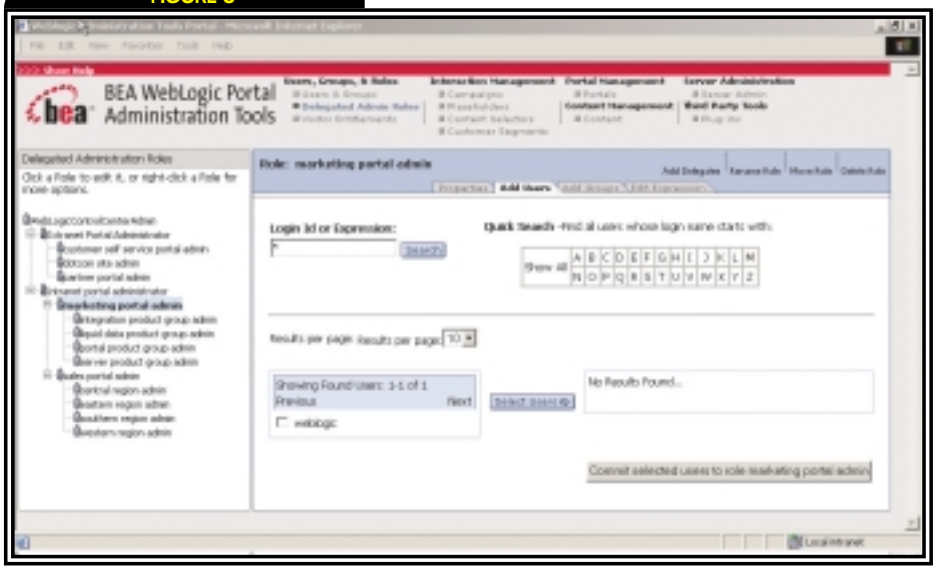


User profile definition

- ***Collaboration:*** Collaboration is a key component of many portal implementations as increased user productivity is usually a component of the return on investment. WebLogic Portal allows for relevant information to be made available where and when users require the information, thereby increasing group and user productivity by improving communication and reducing information search time and project cycle time. Services included in WebLogic Portal are provided through prebuilt portlets and the flexible portal architecture, which supports multiple ways to define communities that support sharing of information and group collaboration.
- ***Commerce:*** Commerce services can be used to build partner, employee, and customer portals that require support for transactions. These features support custom application development and back-end system integration. Developer samples covering catalog, shopping cart, order management, and order discounts, as well as documented programming interfaces and JSP tag libraries, enable you to extend the commerce features or to add your own custom components.
- ***Content management:*** WebLogic Portal provides support for portal content management with a set of content services and an extensible architecture that supports the ability to integrate multiple content repositories as required by business requirements. Developers can use a common set of content and personalization JSP tags and application programming interfaces that support targeting portal users with dynamic content based on user, user segment, or group attributes.
- ***Search:*** WebLogic Portal's enterprise search capabilities use sophisticated pattern-matching techniques to provide secure access to users searching for information. Search capabilities support keyword and metadata to identify concepts within the text itself to determine which are the most important in order to display search results. In addition, a default search interface is provided that can be customized as needed in order to use relevance ranking, document similarity, and proximity search features.

The search engine includes several connectors that support options for integrating different content sources, including ODBC, JDBC, and HTTP sources. This flexibility allows for making unstructured, semistructured, and structured information from

FIGURE 5



Delegated administration Role Mapping Process

multiple repositories based on the business needs of the portal audience. In addition, any search engine that provides a Java API can be integrated into WebLogic Portal.

Interaction Management

WebLogic Portal provides personalization capabilities that support profile, segment, and event-based personalization. These core services can be used to tailor the portal user experience by providing implicit and explicit personalization and a framework for defining and measuring user interaction to better target content and services to users.

- ***Campaign management:*** A full set of campaign services can be used to create an interactive, personalized user experience. Scenarios can be defined to determine how a portal responds to user events, taking into account profile data. This functionality leverages the core elements of WebLogic Portal, such as the rules engine, the UUP, and Content Manager.
- ***Event and behavior tracking:*** Provides the ability to respond in real time to user actions and persist user actions for subsequent processing and reporting. Event and behavior tracking support both a set of pre-defined events and the capabilities for creating custom events.

Security

BEA WebLogic Portal relies on the J2EE security functionality provided by WebLogic Server. This Java Authentication and Authorization Service (JAAS)-based implementation provides a common and unified security structure for developing applications and portals.

- ***Delegated administration:*** Sometimes called "decentralized administration," it is a practical necessity in today 's portal applications. It is common to delegate authority not only to IT personnel, but also to nontechnical users to enable the performance of appropriate tasks. WebLogic Portal supports delegated administration of portals and portal tasks by providing the infrastructure and tools for defining the delegated administration structure and specifying the specific tasks enabled for given users.
- ***Entitlements:*** Provide a means for defining end-user access control to portal resources. Roles can be created and mapped to portal resources in order to restrict access to content, applications, and business processes.

Summary

BEA WebLogic Portal was built to solve the needs of the integrated enterprise. In addition, it continues to implement the latest advances in Web services, XML, and J2EE-based standards that increase the flexibility for creating and integrating enterprise applications. Today, WebLogic Portal provides the most complete portal solution for the integrated enterprise and is a cornerstone of the BEA WebLogic Platform, a comprehensive solution that increases productivity and lowers the cost structure for enterprise IT organizations by providing a unified, simplified, and extensible architecture.

BEA believes that both open-source projects and commercial Java platform products like WebLogic are crucial to the health of the Java ecosystem. That's why WebLogic runs on top of, incorporates, contributes to, and creates open-source technologies.

Open Source, Java, and WebLogic

BENEFICIAL COEXISTENCE RATHER THAN COMPETITION

BY **SCOTT DIETZEN & YARON GOLAND**

AUTHOR BIO

Scott Dietzen is chief technology officer of BEA Systems, Inc.

Yaron Y. Goland is a principal technologist in the office of the CTO at BEA Systems, Inc.

CONTACT...

sdietzen@bea.com, ygoland@bea.com

Even open-source projects that provide functionality similar to WebLogic tend to be best used in a different part of the Java ecosystem than the one WebLogic occupies.

The Roles of Open Source

To understand open source's contribution to the Java ecosystem, let's look at the many roles it plays. Open source allows the Java community to amortize the cost of commodity technologies as widely as possible. For example, let's look at general-purpose XML parsers. The principles behind designing one are well understood. Therefore, outside of application-specific optimizations, most projects won't be able to differentiate themselves based on the quality of their XML parsers. As such, everyone is better off if the community comes together in an open-source project, produces an XML parser that everyone is free to use and modify, and therefore spreads the cost of developing and maintaining the code across the community.

For this reason, BEA WebLogic incorporates Xerces (a generalized XML parser) and Xalan (a generalized style sheet processor), both from the Apache Software Foundation. By using these open-source implementations, BEA can focus its resources on adding value that customers are willing to pay for, rather than reinventing the (XML parser) wheel.

Open source is also an effective vehicle for lowering the barriers to adoption of new technologies.

BEA is a strong proponent of open sourcing Java Specification Request (JSR) Reference Implementations (RIs) and Java Test Compliance Kits (TCKs). Open source gives the Java community immediate and unrestricted access to the new technologies embodied in JSRs, speeding adoption and ensuring compatibility.

This is why, with the JSRs that BEA is leading – such as JSR 181 for Java Web Services (JWS) files – we will generally make the results available through an open-source license. It is also why BEA and its partners have submitted a reference implementation of the XML Script technology now being standardized in ECMA to Mozilla's Rhino project. XML Script is a new technology that makes XML a native data type of JavaScript/ECMAScript. Rhino is an open-source Java implementation of JavaScript/ECMAScript.

In addition to point technologies, open source has proven itself as a means to develop broader software platforms. Two of the most successful open-source projects, Linux and Apache, are of that type. BEA WebLogic was one of the first application servers to treat Linux as a tier-one platform, and the most popular Web server for "front ending" WebLogic applications has long been Apache. Both Linux and Apache help BEA's customers lower the total cost of ownership (TCO) of WebLogic deployments, while giving BEA engineers unfettered access to the underlying implementations.

BEA WebLogic and Tomcat/JBoss

Open-source application platforms raise the greatest number of questions with respect to WebLogic. Two projects in particular are worth mentioning – Tomcat and JBoss.

Tomcat, from the Apache Software Foundation, provides the official reference implementation for Java servlets – the Web containers for J2EE. JBoss provides an implementation of Enterprise JavaBeans (EJB) 2.0, the predominant server-side component model for Java.

Both projects are important members of the Java ecosystem. Tomcat has helped ensure the ubiquity of servlets and JavaServer Pages (JSPs) as well as compatibility between implementations. JBoss has made EJBs, formerly the domain of high-end application servers, available to a wider audience, thereby enabling broader adoption of EJBs than would otherwise have been possible.

In comparing these open-source projects to WebLogic, you may find yourself thinking that even if they don't provide all of J2EE or WebLogic's features, at least they're free. However, the cost of the application platform is usually just a tiny proportion of the TCO for an enterprise software project. The majority of the

TCO comes from managing and maintaining the application once it has been deployed, followed by the cost of developing the application, and then more distantly by the cost of the application platform software itself. Many of the critical features needed to reduce TCO are not available in open-source projects, so free often turns out to be more expensive than it seems.

In deciding between BEA WebLogic and open-source alternatives for your Java enterprise project, we recommend considering the following areas of competency.

Comprehensive and Tightly Integrated J2EE Programming

The J2EE standard combines the essential elements for building a Java enterprise application – a Web container (servlets/JSP), server-side transactional components (EJBs), remote method invocation (RMI), messaging (JMS), XML processing, Web services, database access (JDBC), connector/adapter architecture (J2EE CA), naming/directory (JNDI), and transactions (JTA). Most Java enterprise projects use a combination of these features. Using a combination of point solutions means having to do additional expensive integration work. WebLogic, however, is a complete and certified J2EE-compliant platform that provides all of these features in one integrated solution.

Development and Deployment Tools

Open-source projects are generally written by experts for experts. This often means that the development and deployment environments tend to be flexible but not very easy to use. While most open-source developers feel comfortable using Emacs and shell scripts, most business application developers want a friendlier, more integrated environment. That's why BEA has spent an enormous effort on lowering the barrier to developing and deploying Java enterprise applications employing a broad range of developer skillsets and across highly distributed configurations. The latest example of this is WebLogic Workshop and EJBGen, which bring J2EE development to a wider range of application developers and substantially cut programming costs.

Reliability, Availability, and Scalability (RAS)

For high reliability projects, Web sessions, Web services sessions, EJBs, messaging services, XML/HTML content, and database connections must offer automated replication, caching, load balancing, failover, and data-dependent routing, just to name a few of the requirements. These features are provided transparently by WebLogic today.

Operations, Administration, and Management (OA&M)

For most projects, the lion's share of TCO comes not from development costs but from OA&M. Understanding this, we've worked to ensure that WebLogic offers the best OA&M experience in the industry. OA&M tools are provided, both directly and through partners, that make it easier to configure and maintain a highly reliable, highly distributed system. But clearly more work is needed. Looking forward, BEA continues to make aggressive investments in new technologies to make WebLogic more self-optimizing and self-healing – moving toward more "lights out" administration of even large WebLogic clusters.

Security Framework

Most Java enterprise projects live in a complex security world requiring Web identity, single sign-on, delegation, integration with third-party security products, and policy-/rules-based authorization of access. WebLogic addresses this by providing a security framework that can handle all of these tasks, including extensive integration with multiple third-party security offerings, while at the same time offering ease-of-use to administrators and keeping the complexity of application security out of the business logic (and the programmer's hair).

Support Costs

Business-critical deployments often require 24x7 support around the world and in the local language. Moreover, commercial vendors often are contractually obliged to ensure that their quality-assurance regimen reflects the diverse range of configurations that their paying customers run in production, including JVMs, operating systems, databases, and third-party technologies (content management tools, development tools, application management tools, security plug-ins, integration adapters, etc.). Quality assurance even extends to interoperability testing with other commercial application platforms (such as that carried out for Web services under WS-I).

"In addition to point technologies, open source has proven itself as a means to develop broader software platforms"

Quality of Service and Performance Monitoring

WebLogic provides numerous tools, both directly and in cooperation with its partners, to enable configurable qualities of service across distributed platforms.

Rich Integration and Portal Technologies

Most Java enterprise applications exist in heterogeneous environments that require them to work seamlessly with other enterprise applications and external data, for example, to automate business processes or to provide a unified view of the customer. The BEA WebLogic platform includes Java products such as WebLogic Portal, WebLogic Integration, and Liquid Data for WebLogic to enable this integration.

These criteria are a snapshot of the state of the application server market today. The market is constantly changing, however. Features that were once thought of as the exclusive preserve of high-end application servers such as BEA WebLogic may eventually find their way into open-source projects. As the market changes, the evaluation criteria will change as well.

Conclusion

Almost everyone reading this is developing commercial software. Whether you are a software end user, ISV, or system integrator, you are developing intellectual property (software) that offers a commercial advantage to your employers that they can capitalize on (that is, profit from). In this endeavor, we can and should remain happy to contribute to and leverage open-source.

Because of open source's rich contributions, BEA software runs on top of, incorporates, and contributes to open-source projects. For open-source projects that provide functionality similar to WebLogic, the choice between WebLogic and open source is rarely exclusive. Rather, WebLogic and projects like Tomcat and JBoss serve different parts of the Java community, so deciding which one to use should be done on a project by project basis.

In the end, the story of WebLogic and open source is more about beneficial coexistence than competition.

One of the most enjoyable parts of my job is traveling around the world and talking to CIOs about the many pressing challenges of managing today's heterogeneous IT infrastructure. It's clear to me that in today's difficult economy, it is not that CIOs are "not spending" money. They're just spending the money that they have more wisely.

Confronting Complexity in a Cost-Sensitive World

RETHINKING HOW IT ORGANIZATIONS DISTRIBUTE RESOURCES TO ACCOMMODATE A MORE MODULAR COMPUTING STRUCTURE

BY **MIKE FISTER**

These executives are willing to open their checkbooks – even when their budgets may be 10 to 30% tighter – for anything that has a measurable return on investment. Of course, the great price/performance ratio of the Intel Xeon and Itanium processors can be a big part of that focus on ROI and use of industry standards.

AUTHOR BIO

Mike Fister is vice president and general manager of the Enterprise Platforms Group at Intel, which designs, markets, and supports building blocks for enterprise computing. Products delivered by the group are used in server and workstation platforms and include IA-32 and Itanium architecture processors, chipsets, boards/systems, and software tools and services.

Weighing the Scales: Up and Out

A mega concern for CIOs today is how to scale their software and hardware infrastructure to improve responsiveness across the lines of business – like marketing, finance, and communications departments – without requiring a "forklift upgrade." When it comes to scalability, the choice is between scaling up or scaling out – a decision in which software developers can play a significant role.

Large-scale systems offer impressive processing power. Massive 32-way and 64-way symmetric multiprocessing (SMP) machines powered by Intel silicon will become increasingly common, and some manufacturers have plans to scale Intel-based systems to hundreds of processors in non-uniform memory architecture (NUMA) configurations.

These systems will deliver increased levels of transaction performance because of the sheer power of the hardware. A less obvious, but still important, contributor to delivering these staggering performance results is the software that partitions the workload so that each processor and server effectively performs its part of the job.

When an organization upgrades to newer servers, it requires active intelligence on the part of the IT infrastructure to share the additional resources. Adding servers or storage to accommodate more engineers requires software that monitors transactions and communications to get more out of the existing wired and wireless communications network infrastructure.

In addition to looking for help to scale up to accommodate growing databases or distributed applications, CIOs will probably ask software professionals to consider the financial impact of migrating to a new platform, including the licensing or maintenance fees.

Scale out is a different animal. Rather than an "every so often" overhaul, it's more of a step-and-repeat process. This is the more frequent occurrence where systems are clustered together, each one running an instance of an operating system that provides a compartmentalized service.

Web access and e-mail are great examples that illustrate the need for software that can increase service availability when servers are added to boost processing power.

Many of these types of high-performance clusters, with hundreds or thousands of nodes, have been interconnected in a scale-out fashion. The workload is partitioned such that it runs across all the nodes, no matter how geographically dispersed they are.

Confronting Complexity

CIOs are also concerned about finding automated methods of managing complexity and communicating with disparate platforms. Things were much easier when monolithic apps ran on big equipment and were monitored by guys who were within shouting distance if something went wrong. But now, the skateboard-riding IT guys may not even be in the same building, let alone on the same floor, so they've got to be able to diagnose and manage services remotely.

The intercommunication standards that work as part of RAS (remote access services) frequently enable this, which is great progress. Software that can provide automated communication across servers, laptops, and wireless devices drives reliability, availability, and serviceability leading not only to operating efficiencies but also to improved ROI.

Introducing Modularity

CIOs also talk about the difficulties in managing heterogeneous computing infrastructures that mix wired and wireless communications, and include a variety of hardware form factors and multiple operating environments. Employees

are now distributed across a variety of devices and geographies, and the data and communications infrastructures are likewise spread out over distances.

The hardware components of these emerging modular environments can include rackable servers and blade servers. Blades are essentially circuit cards that contain processors and memory in a small form factor. These computing elements are bound together with a high-speed interconnect, such as InfiniBand Technology or Gigabit Ethernet.

The industry needs to deliver software and hardware that can easily administer and reallocate these communications and storage resources as flexible nodes.

This will require the different abstraction layers – from application servers to operating systems to Web services – to hide the complexity of a distributed computing hierarchy.

One formidable challenge will be to develop solutions that virtualize data through logical partitioning across the entire network. This flexible capacity is required to make efficient use of available resources, which increases ROI by centralizing management.

In addition to allowing databases to be maintained by just a few individuals, every company needs software that distributes and manages the data with a high degree of intelligence. I'm talking about software that is self-optimizing, self healing, and does automatic recovery. In the near future, I expect this software to do things that haven't even been dreamed of yet.

Conquering Complexity

The collision of the diversified platforms with the increasing distribution of resources has produced unprecedented levels of complexity. In fact, "complexity" is now almost a synonym for IT infrastructure.

I'm confident that developers will seize the chance to create comprehensive software that manages incredibly intricate environments while still retaining the simplicity to pacify those demanding users who won't leave their office until their e-mail account is working. Similarly, CEOs are still seeking a reliable way to synch their handhelds with the corporate databases – another opportunity for software developers.

Intel is dedicated to providing leadership in our silicon engineering as well as the overall Intel architecture. I look forward to watching the innovative things that developers will do with Intel's Hyper-Threading Technology, which is now available on desktop systems.

Intel will also continue to extend the Itanium processor family in 2003 and beyond. In addition, Intel continues to advance the Intel Xeon processor family for dual-processing and multi-processing systems.

Intel remains focused on working with the development community to devise solutions that will bring the IA ecosystem benefits of innovation and value to the constantly evolving enterprise.

Today's data center is comprised of a variety of applications, from front-end services to very large back-end databases. Each type of application taxes its server or servers differently, requiring data center administrators and architects to implement different server solutions for optimal performance, scalability, and availability.

Intel Xeon processor MP-based servers are ideal for mid-tier and back-end solutions such as BEA WebLogic, where significant processing power is needed. See www.intel.com/ebusiness/pdf/prod/server/xeon_mp/wp022401.pdf for how the Intel Xeon processor MP satisfies the demands of these high-end applications.

This enterprise infrastructure guide www.intel.com/ebusiness/pdf/affiliates/wp024202.pdf details the business and technical benefits of deploying BEA WebLogic Server* on Intel processor-based servers. It also offers design recommendations for a flexible, scalable three-tier WebLogic-based architecture, as well as best practices for migrating to a WebLogic/Intel platform. Proof points include:

- Intel Solution Services conducted tests running WebLogic Server applications on clustered, 4-way, Intel processor-based platforms. The results show fully linear scalability from 4 to 20 processors, with balanced workloads across all nodes.
- Testing WebLogic Server's scalability on the Intel Xeon processor MP showed that Intel's Hyper-Threading Technology (www.intel.com/ebusiness/products/server/benefits/ht/index.htm) can significantly boost performance for complex transactions.
- The WebLogic JRocket Java Virtual Machine has been optimized for the Intel Xeon processor family to boost performance in multi-threaded applications. BEA has also released a version optimized for Intel Itanium architectures for the most demanding, high-end enterprise applications. (www.bea.com/press/releases/2002/0909_jrockit70_ships.shtml)

BEA WebLogic Server offers a highly flexible and scalable infrastructure solution. It also provides exceptional support for n-tier architecture, and integrates easily with legacy applications and platforms to deliver exceptional agility. For additional BEA resources on Intel.com, see www.intel.com/ids/bea.

IT's Challenges for Performance Management

A COMBINED APPROACH FOR OPTIMUM RESULTS

BY **DAVE WILBY**

The explosion of Web services has spawned significant new challenges for IT and the technologies they use. With the infrastructure requirements for WebLogic applications growing more complex, the addition of Web services suddenly expands the management focus to systems and applications residing outside of IT's control.

AUTHOR BIO...

Dave Wilby is vice president of marketing at Dirig Software. He has over 12 years of product management experience in the performance management space. At Tivoli, Dave served as director of product management for the Web Solutions team, where he was responsible for strategy and development of the product roadmap and guiding eBusiness product initiatives.

CONTACT...

dwilby@dirig.com

Evolving technologies are addressing these requirements by monitoring the various application dependencies, integrations, and layers for WebLogic environments via component-level monitoring, correlation/analysis, and transaction mapping. This article highlights the challenges IT faces in maintaining Web services performance, the solutions available, and the industry/technology requirements needed for Web services management tools.

Web Services: Current State of the Industry

While Web services products are currently the focus of IT, the adoption of Web services technologies within the enterprise is slow. Every IT analyst firm predicts Web services will experience tremendous growth over the coming years. Today, companies are taking a cautious approach to Web services.

IDC recently published a report highlighting this trend:

- Four out of five enterprises intend to undertake Web services projects over the next three years, and nearly one in four have already completed an internal solution using Web services.
- Organizations plan to utilize Web services for a variety of business solutions; however, integration involving internal and external systems currently tops the list of functional uses.
- Web services market share is still up for grabs: roughly 20% of respondents are currently undecided on which vendor to rely on.

Early implementations are solving business problems, but more importantly, they are helping to identify future requirements. Many implementations highlight the need for additional standards to ensure interoperability between the various technologies. The Web services standards portfolio addresses functionality requirements, yet the management area remains vague.

Web Services Standards Evolution

The Web services standards evolution has come in three phases. The first phase involved laying out the core standards: XML Schema, SOAP, WSDL, and UDDI represent the building blocks of Web services.

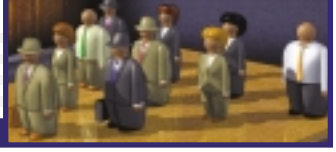
The next phase focuses on security and reliability; WS-I is working on critical Web services specifications like XML Digital Signature, XML Encryption, HTTP-R, SAML, and XACML. Additionally, groups such as OASIS, WS-I, and others are working on additional guidelines around these specifications.

The third phase will address provisioning, transactions, workflow, and systems management. Unfortunately, this phase is in the early stages, but remains a primary concern among enterprises.

Managing Web Services: What Does It Mean?

There is a multitude of interpretations around what "managing Web services" means. Several companies offer Web service management solutions for the integration and distribution, ensuring version control, change management, provisioning, etc.

Recently, OASIS announced its OASIS Management Protocol Technical Committee. The committee's goal is to facilitate distributed systems management over the Internet by sharing management information as a Web service and integrating various management tools such as network management, systems management, application and component management, etc. IT managers view this initiative as the next SNMP, offering a common language to commu-



nicate performance management information.

The success of managing Web services is reliant upon the infrastructure. After security, performance and availability are the primary concerns among companies considering Web services initiatives. Currently there are no standards in place for managing Web services performance; however, many groups, including DMTF, WS-I, W3C, OASIS, and others are investigating and drafting requirements.

Management Requirements for Delivering Web Services

Like any service-oriented IT solution, the success of a Web services transaction depends on two separate infrastructures performing optimally. The service requester first depends upon the availability of the Web services registry. The requester then relies on the availability of the provider and the performance of the provider's environment. Finally, the requester is always dependent on the overall integrity of her or his own infrastructure.

Additionally, the service provider is responsible for ensuring that its services are available on demand, i.e. when the service is down, the company is losing money. The provider must also offer acceptable performance levels to prevent requesters (customers or partners) from changing to a competitor. While availability remains a requirement, performance levels will be what differentiates providers and will act as the basis for service level agreements.

Challenges for Existing Management Tools

There are many challenges facing traditional management tools that ensure performance and availability of Web services. When integrating separate tools to address the different infrastructure elements, IT is still left without visibility into various critical components.

IT managers often default to systems management tools for monitoring of Web servers, portal servers, Web application servers, and database servers. Network management tools provide protocol monitoring for such things as SOAP, HTTP/S, FTP, e-mail, and TCP/IP, and component management tools provide metrics on the performance and availability of JSPs, servlets, EJBs, and JDBC connections. Yet, while this approach addresses some needs a number of issues remain:

- The coverage is limited, creating issues for diverse operating systems, application server platforms, middleware, and database technologies.
- Many traditional tools are not designed with or for Internet technologies, including HTML, XML, HTTP/S, and JMX.
- Enterprise management products fail to deliver a business-centric view into transactions, and a developer's view into the probable issues in the Java code.
- The use of proprietary APIs results in the incompatibility of many tools, versus using standards such as SNMP, XML, SOAP, and WSDL.
- Enterprise management tools are too invasive, whether it's a large framework solution and heavy modules on managed servers or instrumentation and profiling within Java applications. Web services tools must be lightweight with low overhead for minimal performance impact.
- Web services environments will create high volumes of events, alerts, and observations, and massive amounts of historical performance data. Web services require scalable solutions capable of growing with anticipated data volumes.

Leveraging Existing Technology – Java Management Extensions (JMX)

Existing technologies may be leveraged to help fill the void of Web services performance management standards. The JMX specification defines the architecture and API requirements that allow developers to implement and expose the distributed applications management functions. Through the use of JMX, applications can be queried and monitored during runtime to expose performance and availability information within individual components.

By monitoring a JMX-enabled SOAP runtime, a management application can consume many critical performance attributes within a Web services infrastructure, including the total number of Web services deployed and the total number of calls to all services combined. Additionally, data can be monitored from individual services, including successful invocations, the number of invocations per method, the number of failed invocations, and average response times.

For JMX-based management, there are few requirements. To monitor the network, new solutions from companies like SMARTS, Concord Communications, and Micromuse offer scalable solutions designed for Internet environments, and provide the flexibility of standard protocols and APIs to allow for easier integration.

BEA, the leading Java platform vendor, has already taken the step of adopting JMX as the mechanism to expose their performance and availability statistics. By taking these steps BEA has once again shown its commitment to the Java community and to supporting open standards for manageability. In addition, as BEA's platform strategy expands to offer greater coverage of Web services, customers can rest assured that they too will be manageable "out of the box."

Most important, companies need a management solution designed to monitor application server environments, including the operating system, Web applications, and components. Currently, some products provide comprehensive management for performance and availability of application server environments, including CPU, memory, JVM, servlets, EJBs, and methods. In addition to the provider's infrastructure, information is also ascertained from the registry's response time using SOAP, HTTPS, and TCP/IP; and the performance of individual services, such as response time and invocations.

This combined approach for proactively managing Web services ensures users that the services are performing optimally. There are only a few solutions that combine scalability and functionality with the ability to monitor, threshold, and act upon any Mbean within any JMX-enabled application, and monitor specific Java method calls. The ideal product also provides the monitoring information in business-centric views, such as the health or success/failure ratio of a particular service and views for both IT and development.

Despite the lack of standards pertaining to Web services management for WebLogic environments, leveraging JMX and the combined solutions mentioned here addresses today's needs. They prevent service failure and can immediatly identify the probable cause of issues.

References

- IDC – Web Services Awareness and Adoption Study, 2002: Ready and Willing, but Able? (IDC #27736)
- W3C Web Services Architecture Requirements, Working Draft 19; Section D-AC018 Management and Provisioning

Building on a Stable Foundation

AN INTERVIEW WITH OLIVIER HELLEBOID

The editors of *WebLogic Developer's Journal* recently sat down with Olivier Helleboid, BEA's President of Products, to talk about WebLogic's direction in 2003

WLDJ: Tell us about your role as the new president of products.

Olivier Helleboid: I am responsible for all product development activity at BEA. That means the engineering and product management teams, and leading the product strategy and the definition of the future product from development to general availability. That's true for all the WebLogic products and Tuxedo – the entire BEA product family.

WLDJ: What was your background before joining BEA and why this role?

OH: I've been in the enterprise computing industry for roughly 25 years; in that time, I spent about 18 years at Hewlett-Packard in a variety of roles in the networking space and the server area. During my last four years at HP I managed the HP OpenView software business and its business in the enterprise management space. That group generated about $1 billion for HP in my last year with the company. After that I was president and CEO of RainFinity, a start-up company that provides high availability clustering software for network devices.

What really attracted me to BEA was its tremendous position in the marketplace, its number one position in the application server space, and its strategy for leadership in the whole enterprise application infrastructure area. This is a tremendous opportunity to lead BEA to the next wave.

WLDJ: How do you find new features and products that you want to incorporate into the BEA product stack?

OH: As we spend a lot of time and resources developing new products, it is absolutely vital to make sure that those new products are tied to what our customers want. There are a number of processes. The major one is that our product managers spend a lot of time with customers and partners, ISVs, and system integrators to get their feedback. New ideas for innovations also come from the engineers themselves, and discussions with industry analysts also generate new products. A good example is BEA Liquid Data for WebLogic, which is a completely unique product for data integration; that idea came from some internal engineering teams. Overall, it's a combination of gathering customer input, really understanding where the customer pain points are, and always looking for ways to innovate from a technical perspective.

WLDJ: What announcements are you making at BEA eWorld ?

OH: The eWorld theme is around convergence. That represents what we see happening in the industry right now. Customers' requirements and enterprise computing are converging, especially in the application infrastructure space between development and integration. BEA's overall strategy is to deliver a unique WebLogic Platform that helps customers build, integrate, and extend their software applications in the enterprise. It brings together the notion of developing an application with the idea of integrating with your existing products.

We have a very strong set of new products across all elements of the WebLogic Platform to deliver on this notion of convergence, of helping customers do application development and integration in a single environment.

WLDJ: Would you say that BEA is heading more towards development products or administration and management solutions?
OH: Both. BEA's strength has been around development and deployment. This is really to help customers develop new applications, integrate with their existing environments, and then deploy these new applications in their environment. That's the core of our business today. As we look into the future, we are definitely going to move into areas around the operations side – for example, security. The last release of WebLogic Server included a new security framework that allows customers to more easily manage the security of their applications. Customers really want an integrated solution, a platform that deals with the whole life cycle – developing the application, integrating it with their existing environment, deploying it, rolling it out, and then maintaining it, managing it, doing the updates, keeping it current, etc. That's what we are offering.

WLDJ: What do you think about BEA's products regarding ease of use versus extensibility and how are you balancing them?
OH: BEA made a huge leap forward this past year when we introduced WebLogic Workshop. We provide enterprise application infrastructure, and by its very nature it's a pretty complex area. When you look at enterprise environments, they have many, many applications from many different vendors, many different technologies, different hardware operating systems. We have to make things simpler for our customers because just adding to the complexity won't help them. The goal of many of our solutions is to help customers develop new applications more easily and faster than previously possible, and WebLogic Workshop plays a big role in that. A number of studies have shown that by using WebLogic tools you can develop applications much faster in a much smaller number of steps than with alternative solutions. The traditional BEA WebLogic application server also has proven to be very fast to implement. One of our customers is Toshiba; they deployed a portal solution to 15,000 dealers in North America. From start to finish it took them 16 weeks to implement a fairly complicated, large-scale application that connects into their back-end system. Again, we are talking just weeks to deploy applications to make it easier. We sometimes talk about our value proposition as 'Enterprise Made Easy'; we recognize that we are in a complex environment; we need to invest to make it simpler and faster to deploy applications in that environment.

WLDJ: We've talked about simpler and faster. What advantages aside from those do you see BEA's Web services stack having over WebSphere?
OH: Our products are designed to be loosely coupled and work well together, and the whole WebLogic platform is built on our application server. We have a common WebLogic application server as the deployment engine for all of our solutions. WebSphere has many different application server environments that don't work together. We have an integration solution that works with WebLogic Integration, WebLogic Portal, and Liquid Data, all of which have a common deployment environment. WebLogic Workshop sits on top of that and integrates it into a single development environment. That's one of the major benefits to the customer: products that are designed and built to work together out of the box mean ease of use and ease of deployment for customers.

Another big advantage we have over WebSphere is being ahead in new technologies. Take standards as an example; we're 18 months to 2 years ahead in delivering the latest releases supporting the latest versions of J2EE. We keep on releasing new innovations, such as WebLogic Workshop and Liquid Data, ahead of WebSphere. Another thing that also works in our favor is our industry partnerships with ISVs.

WLDJ: What about Oracle's offering?
OH: In the application server area, we don't see them. From our perspective, BEA is the market leader in the app server market. It will be hard for Oracle to break into the app server market in a meaningful way. The other big thing, as you know, is that the industry is now evolving beyond the app server and is really delivering a platform. We have the WebLogic Platform with WebLogic Workshop and WebLogic Integration and everything else in that area. We really have the market share, the customer base to prove that. Oracle is in the database space and that's where we see them, in terms of interconnecting with Oracle databases.

WLDJ: A lot of vendors are developing and marketing solutions around the BEA WebLogic platform. Borland ships an IDE that integrates nicely with WebLogic. Systinet has a Web services framework, and companies like Alignment Software and Precise Software are building management and performance-monitoring solutions for WebLogic. How does BEA work with these vendors when you could just as easily release a similar solution that would affect their business?
OH: Those vendors and those partnerships are very important to us. Obviously, our strength is in putting out this application infrastructure for others to build solutions on. These partners are building applications either in vertical spaces – financial, for example, or transportation – or horizontal applications – customer relationship management or security. They allow our end users to have a much richer set of functionality that is ready for use on the WebLogic Platform. We want to do everything we can to make it easier and faster for our partners to develop on WebLogic and to release new products on it. We have a sizeable team, and a set of programs in sales, marketing, and development to help those partners develop and deploy solutions. We have a standard API and tools documentation and training support for partners, so they can build on WebLogic, whether it's a portlet or a portal solution, or a control for WebLogic Workshop. We want to make sure that's all built in, whether it's also adapters for WebLogic Integration, so they can stay up to date – we can help them do that.

WLDJ: A lot of your products are downloadable for free for a trial basis. Why is this important for the developer community?
OH: It helps to make it easier and faster. In many cases, developers want to try new

releases as early as possible and see what's changed and what they have to change in their product to tie into this new release. It gives developers an early view in terms of what they can work with. That works both for the existing developers to stay up-to-date with us and it allows us to attract new developers so that they can try our software and see the benefits of it, see what they can do around it, without having to commit to anything financially. It's a very powerful tool. We have a very strong program that I'm sure that you're familiar with, dev2dev (see *WebLogic Developer's Journal*, Vol. 2, issue 1).

We work with more than 500,000 developers within that whole program with training, not just the free trial. It's the training seminars, user groups, and a lot of resources that they have available just to make them more comfortable and more expert in working with our software.

WLDJ: Let's talk about the Java Community Process.
OH: That's another thing that BEA is very involved with. The role we have around standards is not just that of adopting and building into a product, it's really a leadership role of driving innovation and driving the definition of new standards. In a number of cases, our engineers have come up with new technical solutions for easier development and then brought that innovation to the JCP in the form of JSRs, worked with other vendors to make it broad enough and adopted, and then the whole community – vendors and developers – can leverage that standard. This work is very important for us because, as you know, BEA is built on standards. Having a proprietary interface doesn't work for us. Our products and architecture are based on an open standards-based approach, so that we can get the broadest adoption of our technology across the industry.

WLDJ: BEA is known particularly for the standards issue. What about open source?
OH: Open source is another interesting area. We have incorporated a number of open-source products and components into our products – Apache is one example. It's something that we believe in and participate in. A number of BEA employees are connected with that community and provide feedback to some of the proposals that are made there. We are currently looking at submitting some of our technology to the open-source space in the next year.

WLDJ: Is there anything else you'd like to mention?
OH: I'd like to reinforce that BEA is in a leadership position. BEA has been very strong in the application server market and in the development of standards-based applications around Java. What we are doing now is adding to that with this notion of convergence – of developing and integrating applications around a single platform based on standards. That's really the big evolution for WebLogic and for BEA – to go from being number one in the application server market to being number one in the application infrastructure market. Industry analyst firm Gartner calls it the application platform suite market. From a developer perspective, we are giving them one unique environment where they can develop and code an application, but also connect to SAP or Oracle or whatever applications are there, and build a portal to extend with third-party security components or Borland tools and so on, all in a single environment running on a single app server. That's totally unique in the industry and BEA is leading that. I think developers will find that's going to make their lives much, much easier.

Many articles have claimed that Java 2, Enterprise Edition (J2EE) and .NET Web services are interoperable. What does this mean from a developer's perspective? What issues and problems arise as you actually do the work?

Integrating J2EE and .NET Web Services

SIMPLE INTEROPERABILITY IS ALREADY HERE

BY **EVELYN HOBSON**

AUTHOR BIO...

Evelyn Hobson is a software engineer and consultant in HP's Developers Resource Organization, which explores leading-edge technologies as they relate to developers. Her areas of interest include XML, J2EE, .NET, and Web services.

CONTACT...

evelyn.hobson@hp.com

For our purposes, I'll define Web services as self-contained, modular business applications that have open, Internet-oriented, standards-based interfaces. The standards I'm referring to are Simple Object Access Protocol (SOAP) and Web Services Definition Language (WSDL). SOAP is a lightweight XML-based message format for communication between Web services. WSDL is a schema that describes the way in which a Web service is to be used. WSDL documents answer the following questions about a Web service: What data is exchanged, how do you interact with the service, and where is the service located? When I refer to term *interoperability* between Web services, I mean that you can call J2EE Web services from a .NET client application and the reverse is also true.

Analysts have projected that J2EE and .NET will each have approximately 40% of the Web services that will be implemented on their platform. This means that for a customer to be able to utilize all the Web services that will be available in the future, the platforms need to interoperate. Companies from both camps are working together to ensure interoperability. Several standards bodies have been formed to drive interoperability between the various Web services platform providers. For example, both SOAPBuilders and the Web Services Interoperability Organization (WS-I) are involved in interoperability efforts.

To prove that simple J2EE and .NET Web services can interoperate, two things must be demonstrated. One, a .NET Web services client should be able to call and receive a response from a J2EE Web service, and two, a J2EE Web services client should be able to call and receive a response from a .NET Web service.

Case Study

In the interest of reusing code and knowledge, the same application was used to prove both points. I started with an existing J2EE application that uses servlets, JavaServerPages (JSPs), and Enterprise JavaBeans (EJBs) that I converted into a Web service. This application allows a field agent to view, edit, and create orders for a fictional construction company. The field agent is also able to sort existing orders by customer name,employee id, or order status.

Calling a J2EE Web Service from a .NET Client

Making Orders a J2EE Web Service

In order to reuse as much code as possible, I used the EJBs that had been written and deployed on WebLogic Server to implement the business logic on the back end. First, I needed to create a Web service that exposed the methods of the EJBs. To create a Web service in BEA WebLogic Workshop, I used the wizard provided and named it Orders (see Figure 1). This generates a Web service file with the name Orders.jws. Using the WebLogic Server 7.0 and WebLogic Workshop Web service runtime environment, I added the EJBs to my Web service with controls. Controls are a mechanism in WebLogic Workshop for interacting with resources from within a Web service. They can be interfaces to databases, EJBs, JMS queues, or other Web services.

Calling the EJBs from the Web Service

In order to call the EJBs from my Web service in WebLogic Workshop, I need a handle to an EJB and this is what the control does. As you can see in the following code sample, the SearchEJB variable is my control; I call a particular method, getOrdersByID, and pass it the incoming parameter, ID. This is where the WebLogic Workshop tool really provides value to the developer. I didn't have to worry about getting references to the home and remote interfaces for the EJB. The control hides this complexity and simplifies the coding required to interface with the EJB. The *jws:operation* tag in the code sample is a signal to the WebLogic Server that this method is part of

the Web service's public contract and is available for clients to invoke.

```
/**
     * @jws:operation
     */
    public ordermasterfields[]
getOrdersByID(String ID)
    {
          ordermasterfields[] orderArray = null;
        try{
            LinkedList orderList =
SearchEJB.getOrdersbyID(ID);
             orderArray = getArray(orderList)
        }
        catch(Exception e){
            System.out.println("*** In
getOrdersByID method *** \n
          Exception is: " + e.getMessage());
        }
        return (orderArray);
}
```

Once I added the EJBs to the Web service, I carefully checked the data types that were being returned from the EJB methods. I wanted to make sure that each of these data types is serializable so that I can return them over the wire to the Web service client. Notice in the code above that my EJB method returns a LinkedList that is not serializable. There are two possible solutions to this problem. I could choose to either change the code in the EJB to return a serializable data type and redeploy the EJB, or I could convert the data type inside the Web service before returning it to the client. I chose the second option for simplicity: I didn't have to redeploy the EJBs and I could still use the original front-end Web application if necessary. Since LinkedLists are not serializable, I converted the Web service method to return an array of "ordermasterfields". This class, ordermasterfields, is user-defined and will be included in the WSDL document so the client knows what the user-defined types are. The *getArray* method performs the conversion.

The next step is to generate a WSDL document for use by a .NET client. In WebLogic Workshop I simply clicked on the "Generate WSDL Document" button and it automatically generated the WSDL document for me. I'm now ready to create a client that will use the Orders J2EE Web service. Notice that all we've really done is add a new Web service interface to the EJBs. As a matter of fact, the Web application front end that I started with is still usable since the EJBs have not been changed. In addition to returning HTML from the original Web application, we are now returning an XML document that contains an array of Orders via SOAP from our Web service.

Creating a .NET Client for the J2EE Orders Web Service

Now that I've created a J2EE Web service I need to create a .NET client that will access it. In Visual Studio .NET, the first step is to import a WSDL document into the ASP.NET Web services client by adding a Web reference to the client through a wizard in Visual Studio .NET. You can either point to a file on your local system or to the URL of a Web service on the Internet. I added the Web reference for the J2EE Orders Web service and Visual Studio .NET automatically generated a client proxy file. This file maps all the operations and data types from the WSDL document to a client proxy code file, Reference.cs, on the client side. Then, all that is necessary is to create a variable that uses the client proxy to call methods on the Web service. The following code fragment contains excerpts from the Reference.cs file that show where to find the Web service and defines return types for a particular method.

```
public Orders() {
  this.Url =
"http://localhost:7001/Orders/Orders.jws";
}

public ordermasterfields[] getOrdersByID(string
ID) {
object[] results =
this.Invoke("getOrdersByID", ne object[] {ID});
return ((ordermasterfields[])(results[0]));
}
```

Now that I have added the Web reference and Visual Studio .NET has created a client proxy, I need to create an instance of the Web reference, Orders, and invoke a method. This is all the code I need to write to invoke a particular method on the Orders Web service.

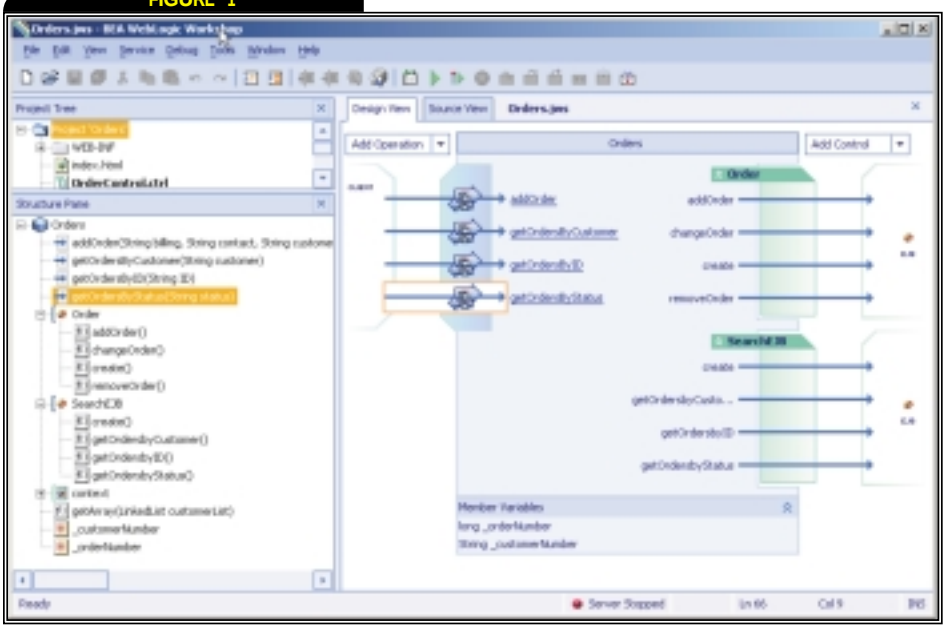
```
using OrderWebApplication.Web
Reference1;

WebReference1.Orders orders = new
WebReference1.Orders();

WebReference1.ordermasterfields[] ordersResult =
orders.getOrdersByID(userName);
```

.NET has a concept of strict separation between the presentation and the implementation of code. They use what they call a code-behind file that is tied to the WebForm page that contains all the code (see Figure 2). All of the presentation is handled in the WebForm.aspx file, which is an ASP.NET page, and all code is written in the WebForm.aspx.cs file. The idea is the same as a JSP file where HTML is the presentation or static content and dynamic content is inserted in JSP tag sections. .NET

FIGURE 1



Orders in WebLogic Workshop

simply extended that and moved the embedded code sections to a separate code file. I could still implement the ASP.NET page with combined HTML and ASP.NET code in the same file if I wanted to. I also created new presentation code with ASP.NET pages that duplicates what was already available on the J2EE platform, but this is not covered in this article.

Request and Response Code

Now that I have both a client and a Web service, let's look at what happens when I make a request to the Web service. With Web services, SOAP request and response messages are sent back and forth either asynchronously or synchronously. Here I show only the portion of the SOAP message that pertains to the Web service method I am invoking. I am calling the getOrdersByID method and passing it an ID parameter with a value of JANE.

```
<getOrdersByID xmlns="http://www.openuri.org/">
<ID>JANE</ID>
</getOrdersByID>
```

In response from the Web service, I get an array of ordermasterfields that represents all orders with an ID of JANE.

```
<ArrayOfordermasterfields
xmlns="http://www.openuri.org/">
<ordermasterfields>
   <orderNbr>1</orderNbr>
   <customerNbr>100</customerNbr>
   <employeeID>JANE</employeeID>
   <customerName>Johns Lock Shop</customerName>
   … …
</ordermasterfields>
</ArrayOfordermasterfields>
```

Figure 3 is my completed .NET client and J2EE Orders Web service.

Calling a .NET Web Service from a J2EE Client

At this point I need to prove that a J2EE Web service client can invoke a .NET Web service. In this case I didn't have a preexisting .NET application so I created a .NET Web service from scratch.

Creating a .NET Web Service

I decided to re-create the same Orders application since I already knew how this application worked on the J2EE side. First I created a new database using Microsoft Access to duplicate the tables. Second, I added ADO.NET objects to access the database and exposed those methods through a Web service page, Orders.asmx.

In order to make the method available through the Web service interface I added the [Web Method] designation in front of the method signature. The DataSet type is a new data type that is currently available only in .NET languages. This means that the Web service client must be able to translate this data type to an equivalent one in the language it implements. In this case, the J2EE client must be able to convert the DataSet type into a Java equivalent.

```
[ WebMethod ]
public DataSet getOrdersByCustomer(int
customerNbr)
{
   DataSet ds = new DataSet("OrderSet");
   oleDbDataAdapter1.SelectCommand.CommandText =
"SELECT OrderNumber,CustomerName,ExpectedShipDate
FROM OrderMaster WHERE CustomerNbr=" +
customerNbr + " ORDER BY OrderNumber";
   oleDbDataAdapter1.Fill(ds,"Orders");
   return ds;
}
```

Creating a J2EE Client for the .NET Orders Web Service

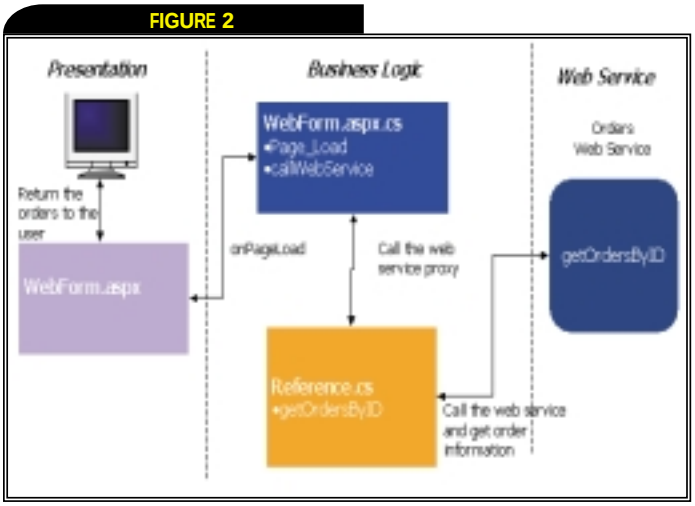
Next I began creating the J2EE client for the .NET Orders Web service. Most of the steps between WebLogic Workshop and Visual Studio .NET are the same, but the way they implement them is slightly different. WebLogic Workshop will also need a WSDL document in order to generate the client proxy, but the WSDL document must first be added to the lib directory before I can create a service control for the .NET Web service from the WSDL document. In the next code snippet I created an instance of the Service Control, created methods that mapped to the Web service methods, and then invoked the method.

```
private OrdersNETControl OrdersNETClient;

public Node getOrdersByCustomer(int customerNbr)
{
   return (Node)
OrdersNETClient.getOrdersByCustomer(customerNbr);
}
```

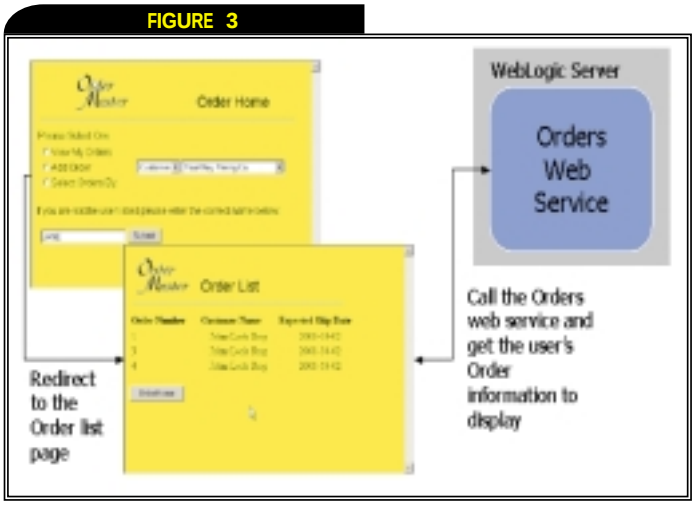
Notice that WebLogic Workshop is expecting a Node to be returned from the .NET Orders Web service. Workshop has mapped the DataSet data type from .NET to a generic XML document Node in order to be able to extract the data values. This is still a little cumbersome to figure out. I had to hunt through the Control file that is generated by WebLogic Workshop to discover this.

FIGURE 2



.NET client architecture

FIGURE 3



.NET client and J2EE Orders Web service

```
OrdersNETControl.control
/** @editor-info:link source="OrdersNET.wsdl"
autogen="true" */
import Weblogic.jws.control.ServiceControl;
/**
 * @jws:location http-
url="http://localhost/OrderRetrieval/Service1.asm
x"
 * @jws:wsdl file="#OrdersNETWsdl"
 */
public interface OrdersNETControl extends
ServiceControl
{
    public org.w3c.dom.Node getOrdersByEmployee
                    (java.lang.String
EmployeeID);
…
}
```

Request and Response Code

Now that I have both a client and a Web service, let's look at what happens when I make a request to the Web service. With Web services, SOAP messages are sent back and forth either asynchronously or synchronously. Here I show only the portion of the SOAP message that pertains to the Web service method that I am invoking. I am calling the getOrdersByCustomer method and passing it a customerNbr parameter with a value of 200.

```
<getOrdersByCustomer xmlns="http://tempuri.org/">

<customerNbr>200</customerNbr>
</getOrdersByCustomer>
Fragment 9 - Request XML message
```

In response from the Web service I get an array of Orders that represent all orders with a customerNbr of 200.

```
<OrderSet xmlns="">
<Orders diffgr:id="Orders1" msdata:rowOrder="0">
<OrderNumber>2</OrderNumber>
   <CustomerName>Set in Place</CustomerName>
   <ExpectedShipDate>2002-10-
02</ExpectedShipDate>
</Orders>
</OrderSet>
```

Considerations and Conclusions

Simple Web services interoperability with SOAP and WSDL can be proven with the following steps:

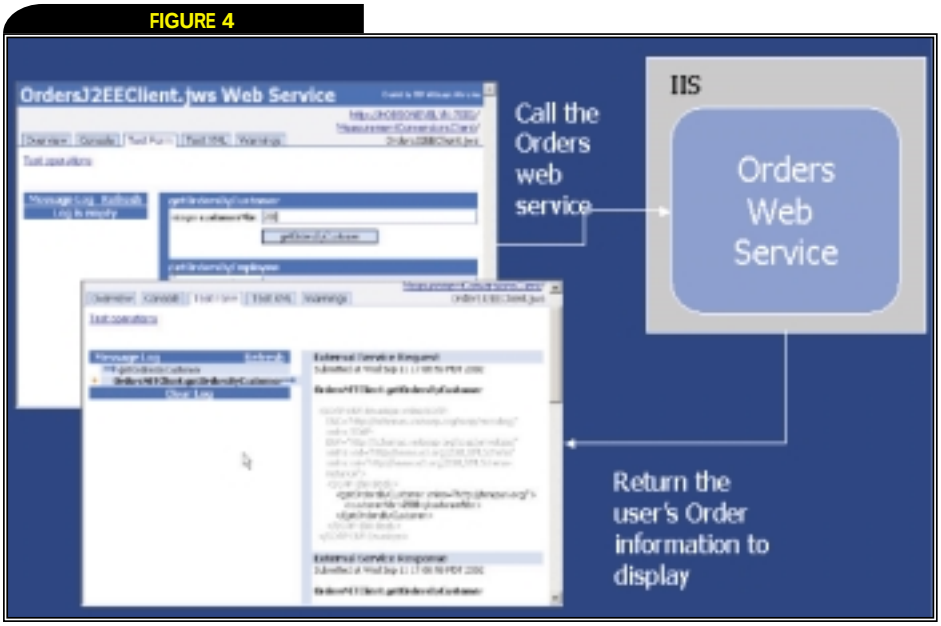
- A J2EE client can invoke a .NET Web service
- A J2EE client can use a .NET WSDL document to create a stub
- A .NET client can invoke a J2EE Web service
- A .NET client can use a J2EE WSDL document to create a stub

There are additional steps that need to be taken by the industry to enhance interoperability between J2EE and .NET. First, support should be added for all XML Schema types. For example, the XML Schema type choice has no standard equivalent in the Java language. Second, support should be added for .NET data types that do not have an equivalent Java type. The ADO.NET type DataSet also has no equivalent Java type. In my experience, WebLogic Workshop translated a DataSet as a generic XML Node that is then parsed on the Java side to pull out the DataSet values. Third, support should be added to J2EE for the extensions that Microsoft has added to their SOAP implementation.

WSDL is at the heart of service interoperability and, a common WSDL interface can increase reuse. Since WSDL defines the data types that a Web service is expecting, it enables interoperability with SOAP clients to the Web service. There are a number of tools available for both J2EE and .NET that can generate client-side proxies, which hides a layer of complexity from the developer. Other best practices include sticking with well-accepted standards (e.g., SOAP 1.2), keeping your data types as simple as possible, and providing XML Schema definitions for all of your data types.

Web services can be written successfully in either .NET or J2EE. Simple interoperability between J2EE and .NET Web services is possible today, and continued improvements are coming from the interoperability standards bodies. Using common data types that can be easily serialized makes the developer's job easier. Learning a new language or a new programming paradigm might be necessary depending on your programming background.

FIGURE 4



J2EEClient and .NET Orders Web service

References

- *BEA Systems:* www.bea.com
- D.H. Brown Associates, Inc. "Web Services and Application Integration with J2EE and Microsoft .NET." December 2002.
- Smith, David; Mark Driver; Chris Le Tocq; Whit Andrews, et al. "Weaving the Future: Microsoft's .NET Initiative," Gartner, September 20, 2001.
- *HP Developer's Central:* www. hp.com/go/developers
- *HP Java products:* www. hp.com/go/java
- Karpinski, Richard. "Mission Interoperable: Rivals Aim to Link Web Services." *InternetWeek*, March 6, 2002.
- Roman, Ed, and Chad Vawter. (June 2001) "J2EE vs. Microsoft.NET: A comparison of building XML-based Web services." The Middleware Company.
- *Microsoft Developer Network:* http://msdn.microsoft.com
- *SOAPBuilders:* www.soapbuilders.org.
- Svoboda, Zdenek. (February 2002) "Developing Interoperable Web Services – Integrating Java and Microsoft .NET." Systinet.
- *Web Services Interoperability Organization:* www.ws-i.org.

THE LARGEST WEB SERVICES, JAVA, XML AND .NET CONFERENCE AND EXPO IN THE WORLD!

Boston 2003 | London 2003 | Berlin 2003 | Hong Kong 2003

JDJ EDGE conference&expo
web services EDGE conference&expo
.NET EDGE conference&expo
XML EDGE conference&expo

Register by February 14th SAVE UP TO $400
Register by March 14th SAVE UP TO $200

3rd Annual

Web Services Edge EAST Conference & Exposition

FINAL PROGRAM

Connecting the Enterprise with Web Services, Java, XML, & .NET

March 18–20, 2003
Hynes Convention Center
Boston

CALL TODAY TO REGISTER
201-802-3058
OR REGISTER ONLINE AT
www.sys-con.com

Event Sponsors

Microsoft · Sun microsystems · SYBASE · IONA · DIGITAL EVOLUTION · COMPUWARE · ORACLE · hp invent · Rational the e-development company · ALTOVA

Media Sponsors

JAVA DEVELOPER'S JOURNAL · JavaWorld · WebServices JOURNAL · LINUX BUSINESS WEEK · XML JOURNAL · SAMS · OASIS · Perl.com · Java Developer Journal · .NET DEVELOPER'S JOURNAL · SD Times · asp.netPRO · Web Services Edge · WebLogic · wireless · CF Advisor · ColdFusion Developer's Journal · ASP Street · HSP STREET · PowerBuilder Developer's Journal

OWNED BY SYS-CON MEDIA

PRODUCED BY SYS-CON EVENTS

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

web services EDGE conference&expo

JDJ EDGE conference&expo

.NET EDGE conference&expo

XML EDGE conference&expo

THE LARGEST WEB SERVICES CONFERENCE & EXPO

WHO SHOULD ATTEND

- Software Developer
- Software Engineer
- Development Manager
- Application Developer
- Technical Director
- Analyst/Programmer
- IT Manager
- Technical Architect
- Team Leader
- Software Consultant

FEATURES & ATTRACTIONS

- **3 Days Packed with Education and Training**
- **4 Keynotes & 3 Panel Discussions**
- **60 Hard-Hitting and Current Seminars**
- **FREE .NET Tutorial with Russ' Tool Shed**
- **Web Services & XML Tutorials**
- **Java University Certification Training**
- **HOT** Industry-Leading Certification Programs
- **1 Day Mobile .NET Tutorial**
- **INFORMATIVE** Round Table Discussions
- Opening Day Welcome Reception
- SAMS Meet the Authors Hot Topics Lounge
- **COMPELLING** Case Studies & Best Practices
- **FEATURED** Product Demonstrations on the Show Floor
- **RIVETING** Real-time SYS-CON Radio Interviews

Keynotes & Highlighted Speakers

KEYNOTE SPEAKERS

John Magee

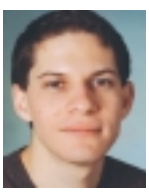
Vice President, Oracle9i

John Magee is vice president, Oracle9*i*, at Oracle. He has more than 14 years' experience in the enterprise software industry and has held positions in product development, product management, and product marketing. In his current role, Magee manages technical product marketing for Oracle's application server and development tools products, and is responsible for evangelizing Oracle technology initiatives around J2EE, XML, and Web services.

Mark Herring

Director Java, Web Services & Tools Business

Mark Herring is responsible for helping to define, set, and drive Sun Microsystems' product direction in the Java, Web Services & Tools Business. Prior to his current role, Herring was director of corporate strategy & planning, looking after Sun's interest in the Project Liberty Alliance and Network Identity. Herring joined Sun Microsystems in October 1999 as a result of Sun's acquisition of Forte Software. Forte Software was a leading provider of enterprise-class development and integration products. During his four years at the company, he ran several aspects of Forte's marketing organization, including product marketing and the Web channel.



Miguel de Icaza

Cofounder and CTO

As the founder and leader of the GNOME Foundation, Miguel de Icaza is one of the foremost luminaries in the Linux development community. With his seemingly boundless energy, de Icaza has galvanized the effort to make Linux accessible and available to the average computer user. He brings this same excitement to his role as CTO of Ximian. de Icaza was instrumental in porting Linux to the SPARC architecture and led development of the Midnight Commander file manager and the Gnumeric spreadsheet. He is also a primary author of the design of the Bonobo component model, which leads the way in the development of large-scale applications in GNOME.

Mark Hapner

Distinguished Engineer, Sun Microsystems

Mark Hapner is a Sun Distinguished Engineer and is currently lead architect for Java™ 2 Platform, Enterprise Edition (J2EE™). He has guided the overall architecture for J2EE 1.2, 1.3, and now the upcoming 1.4 release. In March of 1996, he joined Sun's JavaSoftware Division to participate in the development of the Java database connectivity API (JDBC). Following that, he was co-spec lead of the Enterprise JavaBeans specification and spec lead of the Java Message Service specification.

Simon Phipps

Chief Technology Evangelist, Sun Microsystems

Simon Phipps, currently chief technology evangelist at Sun Microsystems, speaks frequently at industry conferences on the subject of technology trends and futures. He was previously involved in OSI standards in the 1980s, in the earliest collaborative conferencing software in the early 1990s, and in introducing Java and XML to IBM.

Dave Chappell

VP, Chief Technology Evangelist, Sonic Software

Dave Chappell is the vice president and chief technology evangelist for Sonic Software. He has more than 18 years of industry experience building software tools and infrastructure for application developers, spanning all aspects of R&D, sales, marketing, and support services. Chappell has published in numerous technical journals, and is currently writing a series of contributed articles for *Java Developer's Journal*.

Eric Newcomer

Chief Technology Officer, IONA

In the role of chief technology officer at IONA, Eric Newcomer is responsible for IONA's technology roadmap and the direction of IONA's Orbix E2A e-Business Platforms as relates to standards adoption, architecture, and product design. Newcomer joined IONA in November 1999, and most recently served as IONA's vice president of engineering, Web Services Integration Products. He is a member of the XML Protocols and Web Services Architecture working groups at the W3C and IONA's Advisory Committee representative to UDDI.org.

Dean Guida

CEO and President, Infragistics

Dean Guida is CEO and president of Infragistics and was CEO and a cofounder of ProtoView Development Corporation. Mr. Guida has over 15 years of experience in the technical industry and oversees all aspects of the company's business operations and corporate direction. He is also responsible for cultivating strategic alliances and other external relationships, as well as managing corporate financial affairs.

Conference Advisory Board

Sean Rhody
Editor in Chief, *Web Services Journal*
Partner, CSC

Alan Williamson
Editor in Chief, *Java Developer's Journal*
Chief Technology Officer, n-ary

Derek Ferguson
Editor in Chief, *.NET Developer's Journal*
Chief Technology Evangelist, Expand Beyond Corp.

Bob Familiar
.NET Architect, Microsoft New England

Thomas Robbins
Senior Technology Specialist, Microsoft New England

Hitesh Seth
Editor in Chief, *XML-Journal*
Chief Technology Officer, ikigo

J.P. Morgenthal
Chief Services Architect, Software AG

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

Conference at-a-Glance

THE LARGEST AND THE MOST

		JAVA	WEB SERVICES	.NET
TUESDAY MARCH 18	8:00AM – 4:00PM	Registration Open		
	9:00AM – 9:50AM	(JV1) Squeezing the Best Out of Java Alan Williamson, Java Developer's Journal	(WS1) Web Services Infrastructure Carl Sjogreen, BEA	(NT1) .NET Framework Overview Bob Familiar, Microsoft
	10:00AM – 10:50AM	Web Services Keynote: John Magee, Oracle		
	11:00AM – 11:50AM	(JV2) Testing Your Java Using JUnit Kyle Gabhart, LearningPatterns	(WS2) Web Services Management James Phillips, Actional	(NT2) Introduction to ASP.NET Russ Fustino, Microsoft
	1:00PM – 1:50PM	WS-I Panel: "A Road Map for Web Services Standards" - Moderated by Rob Cheng, WS-I		
	2:00PM – 2:50PM	.NET Keynote: "The MONO Project" - Miguel de Icaza, Ximian		
	3:00PM – 3:50PM	(JV3) Building/Deploying the Ant Way Kyle Gabhart, LearningPatterns	(WS3) Strategies for Using Databases in a World of Web Services Mike Lehmann, Oracle	(NT3) Introduction to VB.NET Russ Fustino, Microsoft
	4:00PM – 4:50PM	(JV4) Unlocking the Secrets of JDK1.4 Raghavan Srinivas, Sun Microsystems	(WS4) Using Web Services to Integrate J2EE and .NET Enterprise Applications - Odysseas Pentakalos, SYSNET International	(NT4) How to Develop an End-to End .NET Connected Appilication Allan da Costa Pinto, Microsoft
WEDNESDAY MARCH 19	8:00AM – 4:00PM	Registration Open		
	9:00AM – 9:50AM	(JV5) Java APIs for Web Services Security Standards Sang Shin, Sun Microsystems	(WS5) Combining BPM and BRM Technologies: A Major Step Towards Corporate Agility Henry Bowers, ILOG	(NT5) .NET: The Virtualized Execution Engine Yahya Mirz, Aurora Borealis
	10:00AM – 10:50AM	Java Keynote: Mark Herring, Sun Microsystems		
	11:00AM – 6:00PM	EXPO OPEN 11:00 a.m. - 6:00 p.m.		
	11:00AM – 11:50AM	(JV6) To Not Swing Is to SWT! The Swing Alternative - IBM	(WS6) Web Services for Real-Time Data Access in an Industrial Setting Stephan Van Dijck, ABB/SKYVA	(NT6) Introduction to DotGNU Barry Fitzgerald, DotGNU
	12:00PM – 2:00PM	BREAK & EXPO		
	2:00PM – 2:50PM	.NET Panel Discussion - Moderated by Derek Ferguson, .NET Developer's Journal		
	3:00PM – 3:50PM	(JV7) Unlocking the Power of XML Hitesh Seth, ikigo	(WS7) Web Services Architecture: The Next Big Spec. from the Mouths of the W3C Eric Newcomer, IONA (moderator)	(NT7) Introduction to SSCLI Yahya Mirz, Aurora Borealis
	4:00PM – 4:50PM	(JV8) Java and .NET Derek Ferguson, Expand Beyond	(WS8) Web Services: Next Steps After the Hype Claire Dessaux, Oracle	(NT8) Mobile Development with the Compact Framework Brad Mccabe, Infragistics
THURSDAY MARCH 20	8:00AM – 4:00PM	Registration Open		
	9:00AM – 9:50AM	(JV9) Writing SOAP Services Nigel Thomas, SpiritSoft	(WS9) Web Services Best Practices Chris Peltz, HP	(NT9) Best Practices for .NET Development Joe Stagner, Microsoft
	10:00AM – 10:50AM	.NET Keynote - Jesse Liberty, Liberty Associates		
	11:00AM – 4:00PM	EXPO OPEN 11:00 a.m. - 4:00 p.m.		
	11:00AM – 11:50AM	(JV10) Working with Data the JDO Way Patrick Linsky, SolarMetric	(WS10) Web Services Startups: Telltales of the Future Simeon Simeonov, Polaris Venture Partners	(NT10) Best Practices for ADO.NET Development Thom Robbins, Microsoft
	12:00PM – 2:00PM	BREAK & EXPO		
	2:00PM – 2:50PM	Java Panel - "The Future of Java", Moderated by Alan Williamson, Java Developer's Journal		
	3:00PM – 3:50PM	(JV11) Enterprise: The Next Generation Mark Hapner, Sun Microsystems	(WS11) Open Standards for Web Services Messaging Dave Chappell, Sonic Software	(NT11) How to Debug with .NET Tony Denbow, STAR Information Technology
	4:00PM – 4:50PM	(JV12) Overcoming the Challenges of J2ME Dr. Jeff Capone, Aligo	(WS12) Web Services Security Marc Chanliau, Netegrity	(NT12) XML and Web-Enabling Legacy Applications Using BizTalk Mike Cramer, Microsoft

XML	VENDOR
(XM1) XML - A Manager's Guide JP Morgenthal, Software AG	Visit www.sys-con.com for details
(XM2) OASIS Standards Update Karl Best, OASIS	(VN2) The XMLSPY 5 Enterprise Edition Development Environment Trace Galloway, Altova
(XM3) A Definitive Introduction to XML Schemas Aaron Skonnard, DevelopMentor	(VN3) SOAP and Java: Marrying Them Off Skip Marler, Parasoft
(XM4) XML in Print - XSL:FO Frank Neugebauer, IBM	Visit www.sys-con.com for details
(XM5) XML Security Integration Challenges Phil Steitz, American Express	(VN5) Process-Centric Enterprises Eric Pulier, Digital Evolution
(XM6) Case Study: XML in Life Sciences Tim Matthews, Ipedo	(VN6) Pattern Driven Application Development Tom Shore, Compuware
(XM7) Using XML for EAI - Best Practices Dan Enache, TIBCO	(VN7) Managing the Developer Relationship Mike Bellissimo, Sun Microsystems
(XM8) Take XML with You: XML and Mobile Computing - Hitesh Seth, ikigo	(VN8) Web Services Diagnostics Dave Seidel, Mindreef
(XM9) XML, Ontologies and the Semantic Web - Ayesha Malik, Object Machines	Visit www.sys-con.com for details
(XM10) X Query Mike Champion, Software AG	(VN10) Model Driven Development of Web Services in UML for the J2ME Bill Graham, Rational Software
(XM11) XPath & XSLT 2.0 BEA Kurt Cagle, Cagle Communications	(VN11) Why Web Services Management? Jon Atkins, HP
(XM12) Third Generation XML Tools Michael Leventhal, Commerce One	Visit www.sys-con.com for details

JAVA UNIVERSITY℠ PROGRAM

Sun microsystems

9:00AM – 5:00PM

Web Services Programming Using Java™ Technology and XML

This one-day seminar provides in-depth knowledge on Web services and shows how to develop Web services using the Java programming language and XML, the technologies of portable code and portable data respectively.

Sun microsystems

9:00AM – 5:00PM

Java 2 Platform Programmer Certification Fast Path

This session, developed and delivered by Philip Heller, author of the two leading Java technology certification preparation manuals, helps to prepare you for the Sun Certified Programmer for the Java 2 Platform exam. Philip provides code-level, detailed review of the skills and knowledge needed to confidently approach the exam.

Sun microsystems

9:00AM – 5:00PM

Java 2 Platform Architect Certification Fast Path

This intense one-day session helps prepare attendees to pass the Sun Certified Enterprise Architect for J2EE Technology exam. This session provides an overview of the components comprising the J2EE architecture as a whole, emphasizes the incorporation of J2EE technology into an architecture, and reviews each of the certification exam's testing objectives.

FAST TRACKS & TUTORIALS

XML EDGE conference&expo

9:00AM – 5:00PM

XML Certified Developer Fast Path

This tutorial is for programmers who have some knowledge of XML and related technologies and would like to pass the IBM Certified Developer Test 141 on XML and Related Technologies.

Microsoft®

9:00AM – 5:00PM

Russ' Tool Shed

Join Russ as he shows you how to use Visual Studio.NET.

9:00-12:15 - Introduction to Web Services Using VS.NET

1:00-2:30 - Advanced Web Services Using ASP.NET

2:45-4:15 - .NET Remoting Essentials

RUSS' TOOLSHED TINKERING WITH VISUAL STUDIO

Microsoft .net

9:00AM – 5:00PM

Mobile .NET

In this session, Derek Ferguson, editor-in-chief of .NET Developer's Journal, will give you a thorough introduction to the use of .NET with all manner of mobile computing devices.

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

SUN MICROSYSTEMS Java University℠ Program

Java University℠ Program: Aggressive, code-level training courses for experienced developers using Java™ technology, brought to you by Sun Microsystems, Inc. Attend seminars designed by industry luminaries and recognized experts. Sessions cover Sun certification and Web services technology. Whether you're a beginner or a veteran developer, architect, or software engineer, you'll benefit from these value-packed full-day courses. Register now. Seating is limited.

The Java University℠ program complements this year's Web Services Edge conference by offering 3 full-day training lectures for experienced software developers, architects, and engineers.

Java University℠ Program Take-Aways:

- Training designed and presented by expert Java technology engineers
- Student guides full of source code, examples, references and copies of instructors' materials
- Free Web-based training courses from Sun Educational Services

Tuesday, March 18, 2003

Web Services Using Java™ Technology and XML

SANG SHIN, SUN MICROSYSTEMS, INC.

Who Should Attend

Web services designers and programmers, application developers, and programmers using the Java programming language who have experience using the Java™ 2 Platform, Enterprise Edition (J2EE™).

Prerequisites

Experience using the Java programming language and basic knowledge of XML

Overview

This one-day seminar provides in-depth knowledge on Web services and shows how to develop Web services using the Java programming language and XML, the technologies of portable code and portable data respectively.

The session will start with an introduction on fundamental concepts and characteristics of Web services. This will be followed by a detailed explanation of how to implement, describe, register, discover, and invoke Web services using core Web services standards - Simple Object Access Protocol (SOAP); Web Services Description Language (WSDL); and Universal Description, Discovery, and Integration (UDDI). In addition, the ebXML standard, which defines the framework for the global electronic marketplace will be talked about in detail. Also, the tools for building and deploying Web services will be discussed. Each topic will be presented with concrete examples and demonstrations when possible.

Attendees will also learn how to use standard Java APIs for Web services, mainly Java API for XML Messaging (JAXM), Java technology API for XML-based RPC (JAX-RPC), and Java technology API for XML Registries (JAXR) for developing and deploying Web services.

Benefits

- Learn the fundamental concepts and characteristics of Web services. Gain detailed understanding on core Web services standards: SOAP, WSDL, UDDI.
- Gain a detailed understanding of ebXML, the standard framework for electronic business.
- Learn Java programming language APIs for Web services – JAXM, JAX-RPC, JAXR

Wednesday, March 19, 2003

Java™ 2 Platform: Programmer Certification Fast Path

PHILIP HELLER, PRESIDENT, HELLER ASSOCIATES

Who Should Attend

This session is designed for programmers who have some exposure to the Java™ programming language, and are ready to prepare for the Sun Certified Programmer for Java 2 Platform exam.

Prerequisites

Object-oriented software development experience and familiarity with the syntax and structure of Java technology-based development.

Overview

The development community recognizes that competency developing solutions using Java technology is vital to productivity, reaffirms your value to your organization, and increases your career advancement opportunities. This session, developed and delivered by Philip Heller, author of the two leading Java technology certification preparation manuals, helps to prepare you for the Sun Certified Programmer for the Java 2 Platform exam. Philip provides code-level, detailed review of the skills and knowledge needed to confidently approach the exam.

Benefits

- Receive an intensive review of the advanced topics covered on the Sun Certified Programmer for the Java 2 Platform Exam
- Increase your understanding and knowledge of Java programming language syntax and structure
- Prepare for the exam by reviewing practice tests and questions
- Gain a strong understanding of Java fundamentals

Thursday, March 20, 2003

Java™ 2 Platform: Architect Certification Fast Path

SIMON ROBERTS, TECHNOLOGY EXPERT AND COURSE DEVELOPER, SUN MICROSYSTEMS, INC.

Who Should Attend

This session is designed for enterprise application architects, system analysts, experienced technologists, and developers using Java™ technology seeking certification as an architect for the Java™ 2 Platform, Enterprise Edition (J2EE™).

Prerequisites

Understand the benefits of Java technology solutions; experience with object-oriented analysis and design; familiarity with concepts of distributed computing.

Overview

Many of the solutions in today's "Net economy" are, or soon will be, developed using the Java 2 Platform, Enterprise Edition (J2EE) architecture. Gaining recognized competency architecting J2EE platform-based solutions is vital to your success as an architect, reaffirms your value, and increases your career opportunities.

Developed and presented by Mark Cade, this intense one-day session helps prepare attendees to pass the Sun Certified Enterprise Architect for J2EE Technology exam. This session provides an overview of the components comprising the J2EE architecture as a whole, emphasizes the incorporation of J2EE technology into an architecture, and reviews each of the certification exam's testing objectives. Multiple real-world case studies are used to demonstrate correctly architected J2EE technology–based solutions and pinpoint key topics presented within the architect exam.

Additionally, you will learn how to interpret exam objectives, what each of the three exam phases contains, and clear guidelines and resources to use after the course.

Benefits

- Receive an intensive review of the topics covered on the Sun Certified Enterprise Architect for the Java 2 Platform, Enterprise Edition Exam
- Increase your understanding and knowledge of successfully architecting solutions using J2EE technology
- Understand the system qualities: scalability, availability, extensibility, performance, and security
- Understand trade-offs of different architectural choices as they pertain to system qualities.
- Describe the benefits and weaknesses of potential J2EE technology–based architectures
- State benefits and costs of persistence management strategies
- Review real-world case studies of J2EE technology–based architecture
- Prepare for the exam by reviewing practice tests and questions

Major Sponsors at the World's Largest Web Services, Java, XML, and .NET Conference and Expo

2003 East Gold Sponsors

COMPUWARE

Compuware will be the best worldwide provider of quality software products and services designed to increase productivity. We will continue to create practical solutions that meet our customers needs and surpass their expectations. We will provide an environment for our employees where excellence is encouraged and rewarded and where diversity is promoted at all levels of the company. Compuware is built upon a diverse employee population with a set of key attributes that is common to every individual: Secure, Moral, Ethical, In Agreement with Economic Purpose, Team Player, Other Oriented, Positive, Optimistic, Energetic Achiever and Bright.

Microsoft

Introducing Visual Studio .NET; visionary yet practical, the single comprehensive development tool for creating the next generation of applications has arrived. Developers can use Visual Studio .NET to:
- Build the next-generation Internet.
- Create powerful applications fast and effectively.
- Span any platform or device.

Visual Studio .NET is the only development environment built from the ground up for XML Web services. By allowing applications to share data over the Internet, XML Web services enable developers to assemble applications from new and existing code, regardless of platform, programming language, or object model.

ORACLE

Oracle Corporation (Nasdaq: ORCL) is the world's largest enterprise software company, providing enterprise software to the world's largest and most successful businesses. With annual revenues of more than $9.4 billion, the company offers its database, tools, and application products, along with related consulting, education, and support services. Headquartered in Redwood Shores, California, Oracle is the first software company to develop and deploy 100 percent Internet-enabled enterprise software across its entire product line: database, server, enterprise business applications, and application development, and decision support tools.

2003 East Silver Sponsor

hp invent

HP is a leading provider of products, technologies, solutions and services to consumers and business. The company's offerings span IT infrastructure, personal computing and access devices, global services, and imaging and printing. Our $4 billion (U.S.) annual R&D investment fuels the invention of products, solutions and new technologies, so that we can better serve customers and enter new markets. We invent, engineer, and deliver technology solutions that drive business value, create social value, and improve the lives of our customers.

2003 East Featured Sponsor

Sun microsystems

Sun was founded with one driving vision. A vision of computers that talk to each other no matter who built them. A vision in which technology works for you, not the other way around. While others protected proprietary, stand-alone architectures, we focused on taking companies into the network age, providing systems and software with the scalability and reliability needed to drive the electronic marketplace.

2003 East Corporate Sponsors

IONA

IONA is the leading e-Business Platform provider for Web Services Integration. IONA Orbix E2A enables the flow of information across disparate systems and the liberation of application assets for new business opportunities. For additional information about IONA, visit our Web site at http://www.iona.com.

Rational the e-development company™

Rational Software provides a software development platform that improves the speed, quality, and predictability of software projects. This integrated, full life-cycle solution combines software engineering best practices, market-leading tools, and professional services. Ninety-six of the Fortune 100 rely on Rational tools and services to build better software, faster. Additional information is available on the Internet at http://www.rational.com

SAMS

Sams Publishing has more than 500 titles in print, and is one of the most successful computer book publishers in the world. From introductory tutorials to comprehensive reference books, Sams Publishing focuses on teaching tomorrow's programmers, developers and system administrators the skills they need to build and manage emerging technologies.

SYBASE

Sybase has always delivered solutions that help customers to share data, is platform independent, and integrates everything: platforms, application servers, components, databases, portals, processes, message brokers, and mobile/wireless. Our technologies have the insight that will promote ease of use, leverage best practices, ensure positive ROI, and help your organization build a successful, pragmatic strategy based on next-generation technologies. Check us out at http://www.sybase.com/solutions/e-business

DIGITAL EVOLUTION
The Business Value of Web Services

DE solutions and thought leadership deliver the value of Web services to business users today. DE unleashes the value of underlying IT assets, giving enterprises the freedom to innovate and evolve business processes within a standards-based, vendor neutral, loosely coupled framework. Some of the largest corporations in the world rely on DE's Web services solutions. www.digev.com

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

REGISTER ONLINE TODAY @ SYS-CON.COM

JDJ EDGE conference&expo | web services EDGE conference&expo | .NET EDGE conference&expo | XML EDGE conference&expo

Don't Miss the Developer Event of 2003

Meet with the industry experts, professionals, and visionaries connecting the enterprise with today s i-technology!

Conference
March 18–20, 2003

Exposition
March 19–20, 2003

Hynes Convention Center, Boston

WEB SERVICES EDGE CONFERENCE PAST SPONSORS & EXHIBITORS

WLDJ ADVERTISER INDEX

ADVERTISER	URL	PHONE	PAGE
Altaworks	www.altaworks.com	603-598-2582	60
BEA Systems	http://dev2dev.bea.com/useworkshop	800-817-4BEA	59
Cyclone Commerce	www.cyclonecommerce.com/bea	877-564-7700	3
Dirig Software	www.dirig.com	603-889-2777	31
HP	http://devresource.hp.com	800-752-0900	11
Intel	www.intel.com/ad/bea	408-765-8080	33
PANACYA	www.panacya.com	877-726-2292 x3344	29
Performant	www.performant.com/weblogic1	866-773-6268 x0	47
Pharma-IT	www.pharmaitsummit.com	888-670-8200	47
Precise Software Solutions	www.precise.com/wldj	800-310-4777	13
ReportingEngines	www.reportingengines.com/download/jdj2.jsp	888-864-8665	2
Sitraka (now part of Quest Software)	www.sitraka.com/performance/wldj	800-663-4723	15
Sitraka (now part of Quest Software)	www.sitraka.com/jclass/wldj	800-663-4723	9
SYS-CON E-Newsletters	www.sys-con.com	201-802-3020	42
SYS-CON Media	http://developer.sys-con.com	201-802-3020	32
SYS-CON Publications	www.sys-con.com	888-303-5282	43, 57
Web Services Edge 2003	www.sys-con.com	201-802-3069	49
WebLogic Developer's Journal	www.weblogicdevelopersjournal.com	888-303-5282	37
Wily Technology	www.wilytech.com	888-GET-WILY	4

Advertiser is fully responsible for all financial liability and terms of the contract executed by their agents or agencies who are acting on behalf of the advertiser.

LOOK WHAT'S COMING NEXT MONTH

Portalize It!
by Mark Secrist
A guide to adapting an existing Web application to a portal paradigm

Extending the Product Catalog - Supporting Configurable Products
by Monte Kluemper
Personalizing your online channels to enhance the customer experience

Customizing User Profiles with Portal
by Ryan Upton
Offering the personal touch by extending the User Management framework

Plus:

Simplifying EJB Development with Xdoclet and BEA WebLogic
by Ryan LeCompte
An easy way to configure your beans with an open-source tool that uses attribute-oriented programming concepts

Tips & Tricks: Deployment Model in BEA WebLogic Server 7.0
by Kumar Allamraju
Provide a homogeneous deployment in a clustered environment

Developing JAX–RPC-based Web Services
by Rajesh Sumra
Write a client and Web service that hide complexity

BEA WebLogic DEVELOPER'S JOURNAL

Without exception, the Internet has dramatically changed the way business information is delivered and consumed. The proliferation of free information coupled with fee-based services that have migrated to the Web has proven overwhelming for the average knowledge worker. While every desktop is now a potential research library, few employees can afford the time necessary to seek out the information that can help them do their jobs.

Make the Internet Work for Your Intranet

MANAGING YOUR INFORMATION – FOR LESS

BY **STEVE BOOM**

AUTHOR BIO

Steve Boom is senior vice president for Yahoo! Enterprise Solutions, where he is charged with the development and strategic direction of Yahoo!'s delivery of solutions to the enterprise, including Web-based communications, collaboration, and content solutions, as well as Yahoo! professional services and consulting.

CONTACT...

eworld@yahoo-inc.com

In many cases, simply finding information can consume as much as 25% of the workweek as employees scour internal databases and external information sources (Forrester Research, Inc.).

Access to information is no longer the challenge; rather, timely, efficient access to the right information at the right time is the key to transforming information into insight. Nearly 70% of the Global 2000 offer employees some level of access to fee-based premium content, but the majority turn first to the public Web. Why? It is fast, self-service, and they understand the search and navigation metaphors. At Yahoo!, we've built a global business around helping people – both at home and at work – find information relevant to the task at hand. In fact, more than 33 million people turn to Yahoo! at work (over 71% reach), accounting for over 13 million work hours in November 2002 alone.

Ironically, the trend toward Web-based self-service has not lessened demands on CIOs, research departments, and corporate librarians to provide valuable business information. Rather, there is increased pressure to deliver more targeted information that is personal, timely, and relevant to the user's needs. As a result, companies must learn how to aggregate various types of information, integrate it into existing business systems and processes, and present it to users of varying functional roles or geography. That's a pretty tall order. The good news is that today's enterprise portals are becoming more than intranets – they are the next-generation delivery platform for business information throughout the enterprise.

Powerful Combination: BEA WebLogic Portal and My Yahoo! Enterprise Edition

Due to the rapid evolution of portal technology, companies can now provide sophisticated tools for managing dynamic business information at a lower cost than ever before. As an oft-recognized leader of the enterprise portal marketplace, BEA's WebLogic Portal serves as the platform of choice for some of the largest, most well-known companies in the world. Similarly, we've learned a thing or two over the years about what makes a great portal experience for the user (ever heard of My Yahoo!?). Together, we're committed to improving how business users seek, receive, and monitor the business information they need on a daily basis.

Yahoo! recently introduced My Yahoo! Enterprise Edition, an enterprise-class suite of Web-based tools and applications for managing dynamic content. My Yahoo! Enterprise Edition aggregates content from more than 2,700 sources and seamlessly blends it into over 100 different portlets before delivery to a corporate portal via a single feed. Soon we'll be adding nearly 800 premium business content sources so companies can rely on My Yahoo! Enterprise Edition for more of their external business news and trade information.

External Information Inside the Intranet Means Greater Worker Efficiency

As budgets continue to shrink, companies must find ways to do more with less. We strongly believe that enterprise portals with tightly integrated business information are the most powerful solution.

In addition to the high-level organizational benefits like productivity gains and greater utilization of existing IT investments, enterprise portals allow employees to stay informed about their company and customers, track competitors, and monitor market trends. In combination with internal company databases, business tools, and enterprise applications, external business information helps employees make better, faster, more informed decisions. In essence, it provides the broader context necessary to apply to their day-to-day job.

I urge you to learn more about My Yahoo! Enterprise Edition – visit us at http://enterprise.yahoo.com/portal/myee.

BEA Systems

http://dev2dev.bea.com/useworkshop

Altaworks

www.altaworks.com