DEV2DEV: MAKING A DEVELOPER SITE COME ALIVE

BEA WebLogic DEVELOPER'S JOURNAL

JANUARY 2003 - Volume:2 Issue:1

weblogicdevelopersjournal.com

ENABLING ENTERPRISE PORTAL INTEGRATION WITH WEB SERVICES

Steve Buzzard 6

BEA e-World

www.bea-eworld.com

Component Source

www.componentsource.com/bea

Wily Technology

www.wilytech.com

BEA WebLogic DEVELOPER'S JOURNAL

EDITORIAL ADVISORY BOARD
LEWIS CIRNE, SHAUN CONNOLLY, TYLER JEWELL, WAYNE LESLEY LUND, SEAN RHODY, SCOTT RYAN, 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 BUZZARD, PETER HOLDITCH, RONEN LEWIT, GARY MEYER, PAUL PATRICK, SEAN RHODY, VADIM ROSENBERG, CHRIS SIEMBACK, JASON WESTRA

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

VP, SYS-CON EVENTS
GRISHA DAVIDA

CONFERENCE MANAGER
MICHAEL LYNCH

REGIONAL SALES MANAGERS
MICHAEL PESICK • RICHARD ANDERSON

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

BEA Web Services

BY JASON WESTRA
EDITOR-IN-CHIEF

It is the dawn of a new season as ***BEA WebLogic Developer's Journal*** moves into its second year. What better way to start the new year than with a focus issue on Web services? And it's not too early to do so; as we move closer to BEA's eWorld 2003 developer conference in March, I'm sure Web services will be a hot topic.

Before drafting this month's editorial, I sat down and really thought about what makes Web services compelling to me. I thought this through in numerous "contexts." The two I put myself into are those of a consumer and an application architect. Our current context affects how we perceive the world around us and, in the case of understanding the usefulness of Web services, what makes Web services compelling to us.

As a consumer, quality of service is number one. Web services for consumers should always be available, should function properly, and should be nonintrusive. With that said, when I am surfing a Web site that integrates with Google's search engine Web Service, I don't care where I'm getting my data from – just find it for me. Find it for me now.

Quality of service is nothing new to the Web. High availability and quick response times of online shopping sites, auctions, and B2B exchanges are necessities, not luxuries. Similar to Web sites, when Web services are unavailable or do not respond, their transparency to the consumer is lost and the quality of service to the consumer declines.

In a nutshell, Web services are compelling to me as a consumer because they function and I don't even know they exist. They run on the same infrastructure that has been maturing for many years – the Internet. They utilize the same protocols and security standards, such as HTTP and SSL, that have been around for years, resulting in seamless, reliable services to consumers.

Adam Bosworth, in his interview with ***WLDJ*** (Vol. 1, issue 9), mentioned the quality-of-service characteristics of Web services. They included things like clustering Web services for scalability and fault-tolerance, and queuing service requests to make them more visible, manageable, and transactional. When a Web service's quality of service is high, it is a compelling consumer offering.

Putting myself in another context, that of an application architect, I see Web services as compelling for other reasons. Web services offer the capability to easily integrate disparate information sources across the enterprise, even across enterprises. Web services provide a rich, coarse-grained messaging model for communication. This communication can even be asynchronous to allow long-running business transactions to occur between systems. They also operate over a well-understood protocol (e.g., HTTP and SOAP), and are developed with a well-understood development model, which is important to teams that are already fluent in Web programming.

WebLogic offers a Web services stack that provides high quality of service and ease of integration. BEA Web services are a natural extension of the already reliable, scalable WebLogic Server. They are built using familiar technologies such as Web applications, EJB, and JMS. These technologies make Web services viable for consumers, the enterprise, or B2B.

This month, we have several articles on Web services worth mentioning. First, we have Sean Rhody, editor-in-chief of ***Web Services Journal*** as our guest editor. Sean always has his finger on the pulse of Web services. Next, Gary Meyer provides insight into integrating commercial Web services into WebLogic applications, and Steve Buzzard talks about enabling Enterprise Portal integration. Finally, there's a book review on *BEA WebLogic Workshop* you shouldn't miss.

I hope to see you at the BEA eWorld conference in March. Be sure to attend if you can, and don't hesitate to introduce yourself if we meet in passing! I enjoy meeting *WLDJ* readers as you always provide good feedback on how to make the magazine the best in the industry.

AUTHOR BIO...

Jason Westra is the editor-in-chief of ***WLDJ*** and the CTO of Evolution Hosting, a J2EE Web-hosting 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

ENABLING ENTERPRISE PORTAL INTEGRATION WITH WEB SERVICES

BY **STEVE BUZZARD**

AUTHOR BIO...

Steve Buzzard is a J2EE application architect with Anexinet Corporation (www.anexinet.com), a leading Philadelphia-based consulting firm. Steve has over 17 years of experience in professional software development and has been working almost exclusively with the WebLogic technology stack since late 1998.

CONTACT...

sbuzzard@anexinet.com

Architecting large-scale applications in the real world is hard. Throw portal and enterprise application integration (EAI) concerns into the mix (as you often must) and you make the job all that much harder. You must make any number of difficult decisions, many of which can have a ripple effect throughout the rest of the project's life, for good or ill.

Each layer of your architecture (from front-end load balancers to back-end systems spread throughout the enterprise and, potentially, the world) must be considered in depth before you make architecturally significant choices. Even when dealing with only a subset of these issues (perhaps "only" the ones relative to portal integration), there are many questions to grapple with:

- Should I bridge my Web tier with Pipeline Components acting as business delegates to a workflow/application integration layer and let that layer handle the EAI ugliness, per best practices (or is it really that cut-and-dried in every case)?

- Should my Web tier and workflow tiers be loosely coupled (e.g., JMS) or should I trade off loose coupling, in certain cases, to take advantage of the worklist functionality the Business Process Management (BPM) API affords me?
- How exactly should I interface with my CRM, ERP, and security systems when building a Unified User Profile (UUP)?
- Does the Portal Content Management reference implementation provide me with enough functionality or do I need to evaluate third-party solutions?
- Should I take advantage of the new credential mapping provider to propagate authentication through J2EE CA Adapters? SAML for Web services? Will my third-party Single Sign-On (SSO) security system support these mechanisms? Do I have an SSO solution at all? Do I need one?

Fortunately, this is a magazine article, not the real world, so we can (and will) toss at least a few of these concerns aside in the name of focus and space. This article describes some of the tools and techniques now available with the WebLogic 7.0 Enterprise Platform that enable portal Integration using Web services. A brief prototypical example is described in an attempt to put their use in context. I narrowly define portal integration here as the capability to retrieve, transform, organize, and display information derived from disparate (often external) sources as a unified, personalized whole. I am restricting this article to Web services and so only briefly mention such powerful portal integration capabilities as third-party content and document management, although they must certainly be considered in enterprise architectures of this nature. I will briefly illustrate the capabilities provided by:

- J2EE CA Application Views (application integration)
- Workshop Application Integration Controls
- Liquid Data Views and Sources
- Application integration and Web services workflow plug-ins
- The Unified User Profile Framework
- The Web Services Portlet Wizard

Taken together, these capabilities provide a very powerful framework for loosely coupled enterprise portal integration using Web services. Please note that this article assumes the reader has at least a high-level familiarity with the capabilities of WebLogic Portal 4.0 and Integration 2.1. If not, there is a wealth of material available, both online (http://e-docs.bea.com) and in back issues of this magazine

Portal Integration: A Prototypical Example

Our example is a Case Management Portal for an IT technical support department. Trouble tickets are assigned to support engineers, based on both their technical specialty (e.g., database, UI, transaction management) and expertise level (first tier, second tier, etc.). Each engineer has an associated Profile, the data for which lives in both a Security RDBMS schema and an external CRM system. This Profile holds information on (among other things) their specialty and expertise. Senior engineers may also be designated as managers within their Profile. These engineers have the capability to analyze their subordinates' case histories, including information on average time to completion and case escalation percentage. Two external trouble ticket systems hold the actual data on the individual cases. One system is relatively new and Web services enabled, and the other is older, with a proprietary interface. In addition to core case management functionality, each engineer's portal can be personalized with additional portlets consisting of public tech bulletins, internal bug reporting updates, and the like.

Application Views: "Virtually Anything" Exposed

J2EE Connector Architecture (J2EE CA) Adapters provide a bridge between J2EE components and external Enterprise Information Systems (EIS). EISs requiring an adapter interface often communicate using proprietary protocols and data formats and often have proprietary authentication mechanisms. The WebLogic J2EE CA Adapters handle the protocol translation, often handle data format transformations, and may take advantage of the Credential Mapping Provider in WebLogic to propagate authentication information to the EIS. XA Transaction propagation is possible too, if the EIS has an XA capability.

The J2EE CA 1.0 specification doesn't provide for a standardized interface to the adapter (although an optional one is specified) nor does it provide for a standard messaging format or asynchronous events originating from the EIS. The 1.5 specification (in "proposed final draft 2" status at the time of this writing) has plugged up many of these holes and will be required in J2EE 1.4.

The WebLogic Integration Application View Framework provides a layer on top of the J2EE CA Adapter and fills in the holes in the 1.0 specification (many of the additions in the 1.5 spec were driven by app view capabilities). When you create an application view, you specify XML schemas corresponding to logically associated business services and events that live within the EIS. Services are invoked by passing XML documents corresponding to the request schema and results are provided in the

form of an XML document corresponding to the response schema. Events are delivered asynchronously to the client, again in the form of an XML document corresponding to an agreed upon schema. Application views are created using the browser-based Application Integration console, where services and events are linked to adapter invocations and the corresponding schemas are specified.

Application view services can be invoked and event listeners registered using the Application Integration API. Application views can also be used declaratively within Business Process Management (BPM) Workflows and can be exposed as Web services. Both techniques are described later in this article.

In our Case Management Portal example, we expose both our legacy trouble ticket and CRM systems' proprietary interfaces as application views, each providing a set of business services and asynchronous events appropriate to the associated system.

Workshop Application Integration Controls: Application Views Exposed as Web Services

WebLogic Workshop simplifies the development, deployment, and debugging of Web services using the expressive Workshop IDE. Workshop also provides transparent message buffering and stateful Web services with conversational capabilities. WebLogic Workshop developers are provided with several out-of-the-box "controls" that allow you to easily expose any number of back-end J2EE components as Web services. One such control allows the Workshop developer to expose application view services and events as Web services. This allows a developer to treat interaction with all external systems in a uniform fashion using Web services.

In our Case Management Portal example, we can use the Workshop Application Integration Control to expose the application view fronting our legacy trouble ticket system's proprietary interface as a Web service, allowing us to interface with both trouble ticket systems in a uniform manner.

Liquid Data: "Virtually Anything" Exposed as "Virtual Anything Else"

Liquid Data, a powerful new component of the WebLogic Platform, provides the capability to create "views" on top of a multitude of "data sources" (application views, database schemas, FTP sites, files, Web services, etc.). These "views" can also be chained (i.e., you may have a view of a view). Once defined, both stored and dynamic queries can be built against these views. Queries can be formulated and invoked at runtime via provided EJB and JSP Tag Library-based APIs. Queries can also be published as Web services. Liquid Data's foundation is built upon an implementation of the XQuery specification (www.w3.org/TR/xquery). The Data View Builder constitutes Liquid Data's IDE and provides a Workshop-like GUI where you can build Views against data sources and stored queries against Views (a developer can hand-code advanced queries directly using the XQuery syntax). The Data View Builder also provides the capabilities to Test and debug the views and stored queries.

The key capabilities provided here, for the purposes of this article, are the ability to create Liquid Data Composite Views on top of existing application views and Web services. A View can pass through the information that meets the needs of a particular Portlet or User Profile and transform only that information requiring adjustment. This can be done declaratively, without having to modify the actual Application view or Web service (or file, or database, etc.).

In our Case Management Portal example, a Support Engineer Unified User Profile View that maps to both the Security RDBMS Schema and the adapter-fronted application view that is exposing the CRM System can be created. Similarly, one or more Case Information Views can be created that map to both the Web services-enabled trouble ticket system and the legacy system, whose interface is exposed via an application view.

Workflow and Web Services: BPM Integration

BPM workflows control the flow of the enterprise business processes, hooking in the actual business logic itself through integration plug-in points. Workflows are built using the BPM Studio GUI. The Studio provides a "Visio"-like interface in which to build your workflow in a drag-and-drop fashion. Application view services can be called declaratively from a Workflow Task, and Workflow Event Nodes can be triggered via Application View Events by means of an Application Integration Plug-in, provided out-of-the-box. Similarly, Web services can be called from a Workflow Task by means of a plug-in downloadable from the BEA dev2dev site (http://dev2dev.bea.com). The dev2dev Web services plug-in package also provides a GUI that allows you to expose Application View services as Web services (a limited subset of the functionality provided by the Workshop AI Control).

The portal in our example may interface with a Trouble Ticket Assignment Workflow using the BPM API from a Pipeline Component. A workflow task retrieves the trouble ticket information from the two trouble ticket systems. The task invokes the appropriate Web service operations from the newer system and application view services from the other. This workflow can be built declaratively from the BPM Studio GUI without any hand-coding whatsoever.

The Unified User Profile: Classification and Personalization Integration

The properties that constitute a user's Profile within Portal live in a predefined RDBMS Schema. A user's profile properties are used by Portal's Personalization and Classification components (they are used to determine who you are, what you are, what interests you have, etc.). You can extend this into an enterprise-wide profile via the Portal's Unified User Profile (UUP) framework. This framework allows a developer to plug in user attributes from alternative sources (e.g., LDAP, CRM/ERP systems, etc.). In a nutshell, the developer implements an EntityPropertyManager EJB that in turn does whatever it needs to do to retrieve the extended user attributes. This EJB is referenced by the ProfileManager EJB (you add your EntityPropertyManager reference to this EJB's deployment descriptor environment).

Now that you have started on your EntityPropertyManager EJB, what techniques should you use in it to actually retrieve the user attributes?

- You can use JAX-RPC to retrieve the information from a Web service, if the external system(s) are Web services enabled or if you exposed it as such using Workshop, Liquid Data, or the Web services BPM Plug-in GUI.
- You can use the Liquid Data Query API if you have exposed the external systems as a Liquid Data View.
- You can use the Application Integration API if the external systems have corresponding J2EE CA Adapters fronted by Application Views.
- You can directly interact with the J2EE CA Adapter.
- You can use a proprietary methodology.

–continued on page 36

BEA Systems

http://dev2dev.bea.com/useworkshop

Web Services & WebLogic

BECOMING AN INTEGRAL PART OF THE J2EE SERVER LANDSCAPE

BY **SEAN RHODY**

The Web services world is currently cluttered with code-intensive solutions that require intimate knowledge of lower-level protocols to successfully deploy applications as Web services. Much like the initial situation of the World Wide Web, when a detailed knowledge of the HTML specification was crucial to successful publishing, Web services is mired in UDDI, WSDL, and SOAP. These protocols are important, but just as Notepad was replaced (at least for most developers) by more productive tools such as Dreamweaver, or even FrontPage, so is the landscape of Web services evolving with tools that support higher productivity development of Web services.

With the bundling of WebLogic Workshop along with WebLogic Server 7.0, it's clear that BEA considers Web services to be an integral part of the J2EE server landscape. And now there are several ways to accomplish this with J2EE. First of all, there's the traditional method of development that includes the creation of EJB objects combined with some simple wrapper development to encapsulate the EJB and expose it as a Web service available via SOAP and HTTP. Not much work; after all, a JSP application does almost the same thing, minus the SOAP handling. And in fact Web services will be part of the JDK specification in the next release.

WebLogic Workshop provides a development tool that's an alternative to the pure Java approach of creating EJBs and exposing them. It starts from the perspective of a Web service and generates the Java code needed to support the service deployment into the EJB container. Rather than focusing on code creation, it focuses on application assembly using EJBs and JMS queues as components.

This approach provides some advantage to the Edge developers – those concerned with exposing APIs to external consumers via Web services. It is often the case that these developers do more of an organizational task than a development one – organizing and stringing together various components into a service, rather than developing the components themselves. That's because the Edge developer's task is exposing interfaces and APIs rather than coding the business logic of an organization. For these developers, the graphical nature of WebLogic Workshop outweighs the power of a pure Java IDE such as JBuilder because it provides more of the appropriate functionality for assembly.

And while that's good, WebLogic Workshop still needs to go another mile or two before it is the essential tool for Web services development. Within the WebLogic family, the business rules functionality is implemented using WebLogic Integration, but does not carry over into WebLogic Workshop at all. You must still write code to develop business logic.

That's unfortunate, because this separation currently robs WebLogic Workshop of badly needed Web services functionality – orchestration and integration. For the moment, coding the interaction is all well and good, but the real power of WebLogic Workshop won't be unleashed until the ability to orchestrate a Web service using BPEL4WS (which should be part of Integration in the near future) exists. And not just in WebLogic Integration but also in WebLogic Workshop. While it's understandable that there will be a need for a standalone BPM product such as WebLogic Integration, it would be much more useful and helpful if that product (or at least a subset of its functionality) were a part of WebLogic Workshop, so that the full power of the rules engine could be leveraged in every Web service design. We need more in WebLogic Workshop – we need deep support for Web services, and that has to include orchestration. We also need support for transactions, and a better security structure tied to Web services security and SAML.

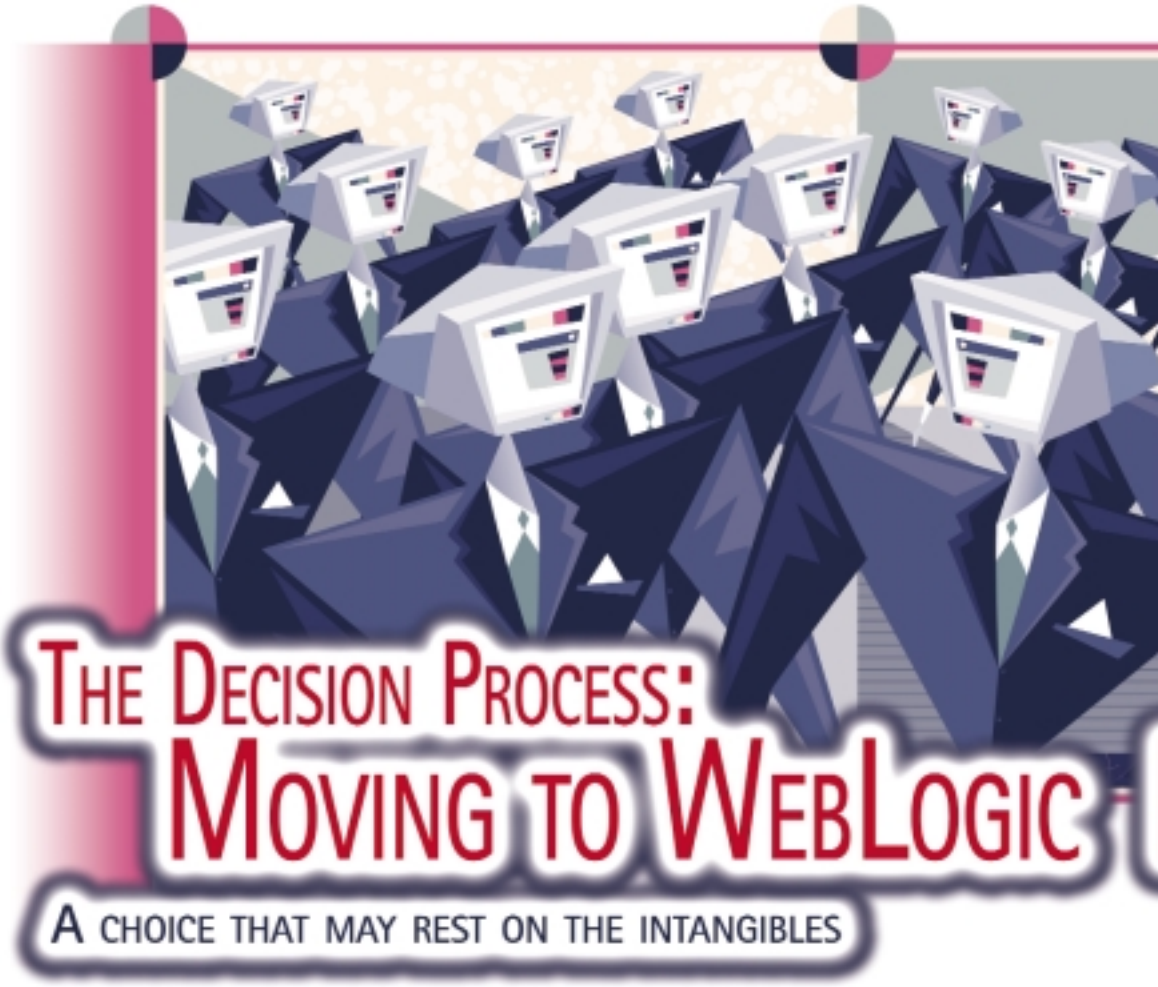
AUTHOR BIO...

Sean Rhody is the editor-in-chief of ***Web Services Journal*** and is a respected industry commentator and speaker. Sean is also a partner with Computer Sciences Corporation, where he works as the Web services advocate for the Consulting Group. He recently published his latest book, *Beginning Java Web services*.

CONTACT: sean@sys-con.com

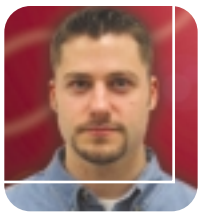
Sitraka (now part of Quest Software)

www.sitraka.com/performance/wldj



The Decision Process: Moving to WebLogic

A choice that may rest on the intangibles



BY **CHRIS SIEMBACK**

BI, a company that specializes in business improvement programs for a wide range of companies, leverages J2EE for many of its software solutions. In fact, we've had several J2EE-based applications in production for years. Our flagship online media redemption vehicle is one such application, and is one of our largest. The application is an online catalog, which participants can use to redeem media for products. This application was the catalyst for our move to WebLogic 7.0.

In this article, I'll discuss the following:

- Why we decided to change application servers
- How we came up with the contenders
- How and why we ultimately chose BEA's WebLogic Server 7.0
- Porting the application

AUTHOR BIO...

Chris Siemback is an applications architect. He has worked on numerous J2EE-based applications and his work has also been featured in *Java Developer's Journal.*

CONTACT...

chris.siemback@biperf.com

The Decision to Move

Before this process began, our application was running happily on a competing application server, which will remain nameless to protect the innocent. There were multiple JVMs running on one physical Solaris box. Not a small application, it interacts with three separate databases (distributed transactions galore!), our internal Order Management (OM) system, and an LDAP security realm. Furthermore, there are upwards of 50 EJBs and hundreds of supporting classes. The database schema is rather large, and yes, we do have a shopping cart!

So why did we decide to move off of our previous application server? After a long period of relatively successful application uptime, we ran into a new situation. Lots of traffic! While we were happily humming along in our environment, our capable sales staff sold our product to a large account, almost instantly creating a much larger demand for our application. Our single-server environment was not initially designed for quick and easy scalability, and it was in danger of being overwhelmed.

We quickly began to do extensive formal load testing of our environment and applications. We made some great strides in enhancing scalability and performance-tuning the code base, but nonetheless a server offering superior clustering would be needed to handle the increased traffic.

The decision to change application servers was also a matter of developer mindshare. The primary developers were not happy with the current application server. Ease of development, applica-

tion server bugs/quirks, outdated specification support, and manual processes (who requires you to do pregeneration of EJB stubs and skeletons anymore?), and more, all led to this decision. Productivity, from the developer's point of view, was not very agreeable. This, combined with business buy-in on the need for a truly enterprise-level environment led to the business decision to move to a new application server and clustered architecture.

Choosing the Contenders

The first order of business was to select the candidate vendors. Considering the size of our team, we decided we needed to narrow down the choices as much as possible. A list of some of the features we most required in a new platform included:

1. ***Scalability:*** Need to cluster and be easy to maintain
2. ***Fault tolerance:*** Clustered scenario must eliminate downtime
3. ***Standards support:*** How well does the platform conform to latest specifications?
4. ***Direct technical support***
5. ***Developer efficiency:*** How easy is it to develop on?

We looked at OC4J briefly, since we're running on an Oracle back end. We also considered JBoss, a popular open-source EJB container. It contains many nice features, but we couldn't quantify how many large companies were actually using it in production. Also, the lack of direct support was a factor in not pursuing an open-source solution. In the end, we decided to look at the two big dogs on the block: IBM's WebSphere 4.0 and BEA's WebLogic Server 7.0.

Defining the Process

A committee of lead developers and system administrators headed up the evaluation process for the contenders. It consisted of the following basic elements:

- A matrix of features, server requirements, and nice-to-haves
- A sales pitch made by each vendor
- References from other companies using the products
- A "bake-off" between the contenders

The feature matrix was critical in the decision process. It contained elements such as desired J2EE specification, clustering/load-balancing capabilities, JDK support, technical support, and so on. We rated both products based on this matrix. Table 1 provides a small example of such a matrix. Some items, such as whether the vendor supports the EJB 2.0 specification or not, are simple yes/no evaluations. Others, like the effectiveness of a vendor's online support, are more subjective and were rated on a 10-point scale.

WebLogic 7.0 fully implements the J2EE 1.3 specification while WebSphere 4.0 was still on version 1.2. We had a strong desire for an EJB 2.0 implementation, giving us such goodies as container-managed relationships, EJBQL, JMS, and message-driven beans. Also, the Servlet 2.3 specification opened the door for us to use such valuable Web-based options as filters. We saw these as extremely valuable items on the list, and the version of WebSphere available to us just didn't provide them. The IBM representatives were quick to point out that all of these features were going to be available in their next release (version 5.0), which was slated for a couple of months after this process was taking place. Ultimately, we decided we couldn't wait for the introduction of a new version. BEA's WebLogic was an existing, stable product that was current with today's J2EE standards. BEA won this round. *Note:* at the time of this writing, WebSphere 5.0 was still not GA!

We also invited both BEA and IBM to send over a technical sales team to perform their canned presentations on the benefits/capabilities of their respective products. While this step was certainly informative, I'm always a little skeptical of sales pitches. Salesmen will sell you the world even when they can't deliver it. Overall, the presentations were simply a showcase for each vendor's bells and whistles. BEA stuck mainly to the application server side of things, which was fine. They brought in a small mix of sales staff and technical engineers that answered our technical questions. IBM, however, sent an entire team of people to provide the presentation. That was impressive. IBM also placed a heavier emphasis on their integrated IDE. I've been using Eclipse as my IDE for some time now, and I really like it. I have to hand it to IBM, the integration with their IDE (a souped-up version of Eclipse) and the application server was great. The presentation, along with the team sent with it, made IBM the winner of this round.

Next, we conducted reference checks with other companies that were using the product. While this sounds like a great idea at face value, the kicker is that the vendor-provided list of references is just that – "vendor provided"! Obviously, neither company is going to provide us with a client reference unless they had rosy things to say about the product. In order to at least partially negate this, we decided to conduct the interviews on a one-on-one basis without a vendor representative present. Needless to say, both vendors were portrayed in a favorable light, but it seemed BEA was even more so. BEA came out on top in this round.

Finally, we conducted a small "bake-off." The purpose of this was to display the setup, ease of development, and use of the application server and its deployment tools. Each vendor sent in a team to port as much of our application as possible in three days to their platform. Since our internal developers were helping each team, we received some hands-on experience working with each server. We didn't expect to have a fully ported application at the end of three days, but we wanted to see actual application server installations and development processes on a real application, and not some "sample" application that comes prepackaged, ready to deploy. BEA sent in a small team and was able to get a small amount of the application ported. IBM, on the other hand, sent in a larger team that had a lot of experience with porting applications. They were able to achieve much more in the same amount of time. While the outcome of the bake-off was

TABLE 1

ATTRIBUTE	WEBLOGIC SERVER	XYZ SERVER
J2EE certified		
JDK support		
EJB 2.0 support		
JAX implementation		
JMS implementation		
JMX support		
Load balance (multiple JVMs, servers)		
Clustering		
Caching		
Real-time failover		
Hot-deploy capability		
Extensive online support		
Quick vendor response		
24x7 Vendor support		
Related user community/forum		
Integrates with IDE tools		
Easy configuration of multiple and independent application instances		
Built-in support for team development and collaboration		
Remote Console for deployment and admin		
Command-line deployment/admin		
Monitoring tools		
Cost of developer licenses		
Perfomance profiling tools		
3rd party add-ons		
Integration with current database		
Integration with current Web server		

Feature matrix

largely due to the number and type of personnel that comprised each team, we had to award this round to IBM.

Making the Decision

If making this decision were as simple as checking off requirements on a checklist, selecting an application server would be a science. Believe me, it's ***not*** a science by any stretch of the imagination! There were several factors, many of which were intangible, that led us to choose BEA WebLogic 7.0 over IBM WebSphere 4.0.

Several members of our development group placed heavy emphasis on the J2EE 1.3 specification, which hurt the IBM camp's chances because WebSphere 4.0 simply did not have J2EE 1.3 support. There was no guarantee when the next version of WebSphere would be released, and many of the development staff were wary of using a newly released product until several service packs were out. WebLogic has supported this specification for some time now.

Another factor in our decision was vendor market share and developer mindshare. At times it felt as though we were the only ones using our previous application server. It made finding solutions to common problems and bugs difficult. Active discussion groups can save a lot of time during development. While both IBM and BEA currently have comparable market share, we found that BEA had a larger market of applications that leveraged EJBs. BEA has numerous active discussion groups online and with the addition of BEA's dev2dev site (http://dev2dev.bea.com), there really is a wealth of information available. In this area, BEA definitely has an edge, although IBM is closing that gap.

Another key intangible factor was developer experience. Personally, I've worked with every version of WebLogic since 4.5.1. Several of the lead developers also had development and deployment experience with WebLogic. We found that that was not the case with WebSphere. There was a smaller learning curve for a majority of our developers, who already knew WebLogic, which we knew would allow us to port our application faster.

Okay, I left out one of the most important factors. Cost. As developers, cost was a non-issue to my team and me; however, the business folks seemed to be concerned with this! This is where I, as a technically savvy person, stepped away and let the sales people do their business. In the end, the costs of the two products turned out to be very comparable – after we made it clear that the two companies were competing against each other. Capitalism at its best!

Porting the Application

With our future application server chosen, it was time to actually port our application to WebLogic 7.0. One of the core concepts of J2EE is application portability. It was time to put that notion to the test.

Although we always try to avoid it, there were a couple of areas where we had written to proprietary APIs. Programmatically logging in a user to a Web application using a security realm is one of those examples. Fortunately for us, WebLogic has a similar capability that we were able to use by simply wrapping our code.

Ultimately, we really didn't need to change much of our code at all. Most of the deltas to our source tree came in the form of modifications to our application packaging and the vendor-specific deployment descriptors. While porting our large application took a little bit of time and effort, the process was relatively smooth. That is a testament to the portability of J2EE applications and WebLogic's support of this standard.

Summary

Overall, BI has been very pleased with the decision to move to WebLogic. WebLogic Server provides the scalable and reliable environment that we require and provides us with the latest support of specifications and APIs. Based on our experience with moving to another application server, you should keep the following in mind:

1. ***Narrow the list of contenders to a manageable size:*** The smaller the list, the easier it becomes to selectively compare the products. It greatly simplifies the overall decision process.
2. ***Use a feature matrix to quickly size-up vendor options:*** This is also a handy way to discover subtleties in each server. Remember, however, that some features are more important than others. Weigh them accordingly!
3. ***Keep in mind all aspects of the server, including scalability, standards support, ease of development and maintenance, as well as proprietary features your application can take advantage of:*** While both servers may support the specification, implementations are vendor specific!
4. ***Be diligent about conducting interviews with vendor references:*** If possible, try to get neutral parties.
5. ***Total Cost of Ownership (TCO) is a factor when considering a large-scale application:*** Remember, TCO involves hardware as well as per CPU licensing. You want an application server that can scale to large numbers on as few machines as possible to lessen the cost of hardware.

I for one am looking forward to working within our new environment!

Interwoven

www.interwoven.com

USING REAL-WORLD COMMERCIAL WEB SERVICES IN BEA PLATFORM 7.0

CLIENT-SIDE WEB SERVICE DEVELOPMENT IS EASIER THAN YOU THINK

BY **GARY MEYER**

AUTHOR BIO...

Gary Meyer is a senior technical director for eFORCE, where he helps Fortune 1000 clients by delivering fixed-price, fixed-time projects for enterprise portals, enterprise content management, EAI, CRM, enterprise infrastructure, and Web services. Gary has been the lead architect for multiple Web services projects.

CONTACT...

gmeyer@eforceglobal.com

Web services are, among other things, a steamy bowl of alphabet soup. When developers set out to explore this new area of software development, they are immediately confronted with a number of new terms such as SOAP, WSDL, UDDI, JAX-RPC, and more. While these terms can initially seem confusing and potentially intimidating, most experienced developers are surprised to learn how easy it actually is to begin developing with Web services. Developers are often pleasantly surprised to discover how little of these "inner workings" one need know before getting started as an effective Web services developer. This is especially true with the improved support for Web services in BEA WebLogic Platform version 7.0.

When talking about Web services, there are principally two roles: the client and the server. The "client" uses the Web service while the "server" provides the Web service for use by various clients within the enterprise and beyond. This article focuses exclusively on the client side of Web services and does so through a real-world example. This example is the validation of U.S. geographical addresses via a commercially available Web service.

Nearly all enterprise business systems manage and process geographical address data. Examples include customer addresses, billing addresses, shipping addresses, delivery addresses, and more. The relevant business question is, what is the business benefit of ensuring these addresses are correct? Will correct addresses reduce shipping costs? Will correct addresses also reduce delivery costs? Furthermore, will

they reduce other costs associated with redundant addresses? In 2003, one should ask oneself, with the availability of address validating Web services, why should any of the business-critical addresses of a large corporation be at risk of being incorrect?

One example of the small number of commercial addresses validating Web services available today is DOTS for Address Validation provided by ServiceObjects. ServiceObjects, an early innovator of Web services, provides this Web service as well as a number of similar utility Web services. Examples include stock quotes and company information, weather information, IP address-to-location determination, nearest ATM location identification, reverse telephone number lookup, yellow pages, e-mail address validation, shipping comparisons, package tracking, and more. The example presented will focus solely on address validation.

Identifying and Understanding the Web Service

The first step in using a Web service is identifying the Web service to use. The most important item to obtain is a reference to the Web Services Definition Language (WSDL) file. The WSDL file describes the interface, or the functional capabilities, of the Web service. In the DOTS for Address Validation example, the WSDL is located at:

```
http://ws2.serviceobjects.net/av/AddressValidate.asmx?WSDL
```

For input, this Web service takes a formatted input address and a license key through a number of parameters. Although an individual Web service can support multiple methods, this example has only one method, ValidateAddress(). Each Web service method has a number of arguments. In the ValidateAddress() example, the method takes five parameters, each of type String. These include address, city, state, postalCode, and licenseKey. An example input address is that of the author's employer, eFORCE, with corporate headquarters located at 4120 Point Eden Way, Hayward, CA, 94545. In addition to the input address, a license key is required. For this specific Web service the value "0" designates a trial license key which can be used for testing purposes. This example illustrates one way vendors are accommodating licensing of Web services as standards evolve.

For output, the Web service returns a response. In the illustrated example the response may contain an Address comprised of the fields of the validated address or an Err if any errors occurred. When passed the above address of eFORCE, the Web service responds with 4120 POINT EDEN WAY, HAYWARD, CA, 94545-3703 having validated the address and filled in the proper ZIP+4 value.

It is interesting to note that while WSDL is a semantic equivalent of several predecessor definition languages, such as CORBA IDL, and it is based upon XML, it is not meant for human readability. Reading and writing WSDL are best left to tools.

Setting Up the Environment

There are several different tools for building and using Web services from within BEA WebLogic Server version 7.0. This example will leverage the command-line utilities.

The first step is to set up the environment. Execute either setWLSEnv.cmd or setWLSEnv.sh depending upon your operating system. This script is located in the directory NL_HOME/server/bin where NL_HOME is the top-level directory of the WebLogic Platform installation. At this point the Web services client generation utility, clientgen, should be in the CLASSPATH. This can be confirmed with the command:

```
java weblogic.webservices.clientgen
```

This should produce usage help for clientgen.

Generating the Proxy

The easiest way to utilize a Web service is to create a static-binding proxy. To generate such a proxy for the example address validation Web service, the following command is utilized:

```
java Weblogic.Webservices.clientgen \-wsdl
   http://ws2.serviceobjects.net/av/Address
   Validate.asmx?WSDL \
-clientJar av.jar \
-packageName com.serviceobjects.www
```

This will produce an av.jar file that will need to be placed in the CLASSPATH.

Writing a Robust Wrapper

It is often a good practice to create wrappers for underlying third-party implementations. It is also often a good habit to test components outside the scope of the application server where possible (see Listing 1). The first class used in the wrapper is an address value object.

In this example, the value object is immutable. The immutable pattern is leveraged so the AddressFactory must be utilized by client classes to obtain an Address object.

The AddressFactory is the class utilized by client classes to obtain a valid address. Notice that this class includes a simple main() which will be used later for simple unit testing (see Listing 2).

To obtain a validated address, client classes simply call the factory method of the AddressFactory as follows:

```
Address address = AddressFactory.getAddress(
   "4120 Point Eden Way",
  "Hayward", "CA", "94545");
```

The AddressFactory class uses several classes in its implementation. InvalidAddressException and AddressValidationException are application exceptions used to designate a bad input address and an internal processing error respectively. In addition to these two exception classes, the AddressFactory uses an AddressValidator interface and a DOTSAddressValidator implementation of that interface.

The AddressValidator interface represents a business interface for address validation. Underlying implementations must implement the validate() method.

```
package com.eforceglobal.wldj.address;

interface AddressValidator {

  Address validate(String address, String city,
String state,
    String zip) throws InvalidAddressException,
AddressValidationException;

}
```

The core functionality that then ties the vendor-independent wrapper to the underlying DOTS implementation is the DOTSAddressValidator class. In this class, the implemented validate() method delegates to the previously generated proxy code using WebLogic's clientgen tool (see Listing 3).

The first initialization task that the init() method performs is setting the two system properties to instruct the JAX-RPC to use the WebLogic imple-

mentation. This initialization must be done when testing or running outside WebLogic Server and only needs to be done once – on application start-up.

The second initialization task that the init() method performs is to initialize a reference to a DOTSAddressValidate. This interface is a JAX-RPC service. The generated proxy service implementation, DOTSAddressValidate_Impl, is an implementation of javax.xml.rpc.Service. The interface is a factory and is, in many ways, analogous to the home interface in EJB client-side programming. This type of initialization only needs to be done once as this factory reference can be shared across concurrent requests.

Once these two initialization tasks have been performed, the class is ready for ongoing processing. For each client request, the validate() method performs several steps.

The first step that the validate() method performs is to create a new reference to the Web service "business" interface, here DOTSAddressValidateSoap. This step is analogous to the utilization of a home interface to obtain a reference to a remote interface in EJB client-side programming. Notice that a remote is created for each user request. This enables multiple concurrent threads in this method to share the factory but each has different implementations of the business interface.

The second step that the validate() method performs is to delegate to the generated proxy code by calling the validateAddress() method. The generated code will serialize the request from Java into SOAP, post it to the server, receive the response, and deserialize it back from SOAP into Java. Notice that the programmer did not need to delve into any details of SOAP data structures or even the JAX-RPC API.

In addition to the steps described above, the validate() method maps between underlying technology-specific exceptions to business exceptions. In providing such a wrapper, the Web tier, or GUI, client programmer does not need to know whether the AddressFactory is implemented via an EJB, Web service, database call, or other approach. This illustrates the power of encapsulation being applied to Web services.

Once the response has been obtained, it is returned to the client.

Trying It Out

To compile this class, it is necessary to have both the generated proxy described above as well as WebLogic's JAX-RPC implementation in the CLASSPATH. The generated proxy is the av.jar file created above. The JAX-RPC implementation is in webserviceclient.jar and is distributed as part of WebLogic Server version 7.0. Once these two JAR files are in the CLASSPATH, the above classes can be compiled.

The example can be tested from the command line, with the command:

```
java com.eforceglobal.address.wldj.AddressFactory
"4120 Point Eden Way" Hayward CA 94545
```

Next Steps

One enhancement that could be made to the above implementation is to support "best effort" address validation. For instance, if the Web service is down, the address is not validated.

A second enhancement that is possible is to provide support for caching. The wrapper could first check a local memory cache and then call the Web service if not present in the cache. This could save money on a Web service that charges on a per-use basis as well as provide higher performance provided the data latency is acceptable. However, care would need to be taken in making sure the implementation complies with the Web service's license.

A solution that completely abstracts the ServiceObjects implementation provides "best effort" validation as well as does caching is possible in only 200 lines of code.

The sample code presented in this article as well as solutions for the two enhancements above are available from the author via e-mail.

More Information

In addition to the capabilities of ServiceObjects, there are a number of other commercial Web services providers. Amazon.com now provides access to their product catalog via Web services. Google provides limited access to their search engine. Microsoft provides maps and driving directions. You can use the techniques described in this article to integrate with those Web services from WebLogic.

In addition to the above Web services, two excellent industry sites can be used for tracking progress now being made on Web services – SalCentral and XMethods. These sites have detailed information about Web services that can easily be reviewed and tested.

Summary

Commercial Web services are now available. Some of these services, such as address validation, are useful in nearly every enterprise application.

To use these Web services, client-side Web service programming is not difficult. With the capabilities now in WebLogic, it is possible for developers to avoid many of the details of SOAP, WSDL, and JAX-RPC. Additionally, with the application of patterns such as facade, value object, immutable, factory method, cache, smart proxy, and more, it's easy to abstract other developers from robust implementations.

Web services are now here, perhaps only a few hundred lines of code away from today's enterprise applications based upon BEA WebLogic Server.

Web service programming is not difficult. With the capabilities now in WebLogic, it is possible for developers to avoid many of the details of SOAP, WSDL, and JAX-RPC

References

- *BEA WebLogic Server 7.0 Web Services Documentation:* http://edocs.bea.com/wls/docs70/webservices.html
- *ServiceObjects:* www.serviceobjects.com
- *Amazon.com Web Services:* www.amazon.com/Webservices
- *Google Web Services:* www.google.com/api
- *XMethods:* www.xmethods.org
- *SalCentral:* www.salcentral.com

Sitraka (now part of Quest Software)

www.sitraka.com/jclass/wldj

Listing 1

```
package com.eforceglobal.wldj.address;

import java.io.Serializable;

public class Address implements Serializable {

  private String address = "";
  private String city = "";
  private String state = "";
  private String zip = "";

  private Address() {}

  Address(String address, String city, String state, String zip) {
    this.address = address;
    this.city = city;
    this.state = state;
    this.zip = zip;
  }

  public String getAddress() { return address; }
  public String getCity() { return city; }
  public String getState() { return state; }
  public String getZip() { return zip; }

  public String toString() {
    return getAddress() + "\n" +
      getCity() + ", " + getState() + "  " + getZip();
  }
}
```

Listing 2

```
package com.eforceglobal.wldj.address;

public class AddressFactory {

private AddressFactory() { }

  public static Address getAddress(String address, String city, String
state,
    String zip) throws InvalidAddressException,
AddressValidationException {

    AddressValidator validator = new DOTSAddressValidator();
    return validator.validate(address, city, state, zip);

  }

  public static void main(String[] args) {
    try {
      System.out.println(AddressFactory.getAddress("4120 Point Eden
Way",
        "Hayward", "CA", "94545"));
    } catch (InvalidAddressException e) {
      System.out.println("Invalid address!");
      System.out.println(e.getMessage());
    } catch (AddressValidationException e) {
      System.out.println("Address validation failed!");
      System.out.println(e.getMessage());
    }
  }

}
```

Listing 3

```
package com.eforceglobal.wldj.address;

import java.io.IOException;
import java.rmi.RemoteException;
import com.serviceobjects.www.DOTSAddressValidate;
import com.serviceobjects.www.DOTSAddressValidate_Impl;
import com.serviceobjects.www.DOTSAddressValidateSoap;
import com.serviceobjects.www.Err;

class DOTSAddressValidator implements AddressValidator {

  private static final String LICENSE_KEY = "0";
  private static DOTSAddressValidate factory = null;

  DOTSAddressValidator() { }

  public static synchronized void init() throws IOException {
    if (null == factory) {
      System.setProperty("javax.xml.soap.MessageFactory",
        "Weblogic.Webservices.core.soap.MessageFactoryImpl");
      System.setProperty("javax.xml.rpc.ServiceFactory",
        "Weblogic.Webservices.core.rp.ServiceFactoryImpl");
        factory = new DOTSAddressValidate_Impl();
    }
  }

  public Address validate(String address, String city,
    String state, String zip) throws
    InvalidAddressException, AddressValidationException {

    if (null == factory) {
      try {
        init();
      } catch (IOException e) {
        throw new AddressValidationException(e.getMessage());
      }
    }

    try {
      DOTSAddressValidateSoap remote =
factory.getDOTSAddressValidateSoap();
      com.serviceobjects.www.Address dotsAddress =
remote.validateAddress(
        address, city, state, zip, LICENSE_KEY);
      Err err = dotsAddress.getError();
      if (null != err) {
        throw new InvalidAddressException(err.getNumber() + ":" +
          err.getLocation() + ":" + err.getDesc());
      }
      return new Address(dotsAddress.getAddress(),
dotsAddress.getCity(),
        dotsAddress.getState(), dotsAddress.getZip());
    } catch (RemoteException e) {
      throw new AddressValidationException(e.getMessage());
    }
  }

}
```

Hewlett-Packard

www.hp.com/plus_explorers

LinuxWorld

www.linuxworldexpo.com

LinuxWorld

www.linuxworldexpo.com

A common complaint in the transaction newsgroup is, "I've done my database updates in a JTA transaction, but they didn't complete as a unit!"

Transactions: That's enough of your source!

FEATURES SUCH AS LRO KEEP WLS AHEAD OF THE PACK

BY **PETER HOLDITCH**

AUTHOR BIO

Peter Holditch joined BEA as a consultant in the Northern European Professional Services organization in September 1996. He now works as a presales architect in the UK. Peter has a degree in electronic and computer engineering from the University of Birmingham.

CONTACT...

peter.holditch@bea.com

In many cases, the explanation for this unfortunate loss of ACID is that the database connections that were used in the logic weren't obtained from a transactional data source, or Tx Data Source as it's abbreviated in the console. The shorthand explanation for this is that any database connection that is to participate in a distributed transaction needs to expose an xa interface, so you need a data source that can pass back an xa connection and that references a pool of xa database connections. The data source that is capable of passing through an xa connection is, you guessed it, the Tx Data Source. The bottom line: when you set up a database connection, use a Tx Data Source and an xa-compliant JDBC driver if there's any chance that you want to access it in a distributed transaction. In fact, I would boldly go so far as to say that you should always use xa drivers (where available) and Tx Data Source data sources to avoid future problems as code and infrastructure reuse change the interactions between the elements of your system.

By way of illustration, the example server shipped with WebLogic Server 7.0 provides a demo connection pool of connections to the PointBase sample database called demoXApool. The Driver class name used by this pool is com.pointbase.xa.xaDataSource, which is the name of the PointBase driver class that exposes the xa interface. This pool is in turn accessed through the Tx Data Source called examples-dataSource-demoXAPool, which exposes the xa capabilities of the PointBase driver up to the calling layers, allowing database updates to participate in JTA distributed transaction processing.

Get Something for Nothing?

If you've been following this example in your console, you may have noticed a checkbox in the configuration of the Tx Data Source entitled "Emulate Two-Phase Commit for non-XA Driver". If you were feeling optimistic the moment you saw that, your mind may have started racing and you may have reached the conclusion that if you have a non xa-capable driver, you need to simply check this box and all will be well, by virtue of WebLogic Server's ability to somehow emulate xa at the click of a checkbox. If so, my friend, I advise you to wake up and smell the heuristics, preferably quickly, before you lose the integrity of your valuable business data. You may have been led even more quickly to this conclusion in the 6.x versions of the server where, if memory serves, this option was simply entitled "enable two-phase commit". It would be surprising if all the effort that database vendors put into supporting the xa standard was able to be emulated simply by checking a box, and of course the world being generally free from such surprises, this is not the case.

Recall from past articles that the xa interface allows a transaction manager to commit a database transaction in two phases. Instead of just saying "commit" – the standard transaction demarcation verb available in all database drivers – the TM can, through the offices of xa, say "prepare" and then when it has finally made up its mind say "commit". Without the xa interface, there is no way to achieve this, and in fact the "emulation" of two-phase commit provided by WebLogic Server is simply to say "Yes" to an instruction to prepare and then pass an instruction to commit through to the (non-xa) database connection as an ordinary commit instruction. If all goes well, this will yield something that looks like a distributed transaction.

If all goes badly, this will be the beginning of your (or your WebLogic administrator's) nightmare. By way of illustration of what could go wrong, imagine a failure mode whereby the non-xa database gets corrupted, or fails after the prepare has happened. When the time comes for the commit call to be issued, there will be no way back. The commit will fail due to the corruption or disconnection, but by this point all the other databases will be gaily committing their own

portions of the transaction, despite the loss of the update. Real ACID would demand that all the other databases roll back to preserve the transactional integrity. As you can see, we have lost our ACID, and with it our consistent data. In fact, this "simulation" will only be guaranteed to work if nothing in the processing environment goes wrong during certain critical windows around the commit processing. Since transactions are supposed to be all about recovering from failures, especially tricky failures during critical windows, the simulation is of no real value. The advice must be, use it as a convenience during development if you don't have an xa database driver at hand, but *never* check that box in a production system.

Last One to the Moon by Magic

So the xa transaction simulation certainly isn't anything close to magic moon dust. But there is something that is… Last Resource Optimization or LRO.

Imagine a transaction spanning several resources (generally, a resource is a fancy word for a database connection) where all but one of the resources are xa compliant. Now imagine that at prepare time the transaction manager (TM) was to successfully prepare all the xa resources. Without the non-xa resource, with all the prepares in OK, the decision to commit would be logged by the TM, and the commits would be issued. With the non-xa resource, we could limit the window for failure down to a very small duration if we were to log the decision and then immediately commit the single non-xa resource with a single phase commit instruction. If this single phase commit were to succeed, then we could go ahead with the normal second phase commitment of the rest of the resources. With this scheme, we would only be vulnerable to failure in the small window between the time the commit decision is logged and the time the non-xa commit completes. After that point, if anything failed, the non-xa resource would have been committed and eventually the xa resources would be rolled forward by the standard recovery processing. In fact we could go a step further and record the transaction ID in the non-xa resource. Now we could recover the transaction with certainty, because during recovery processing the TM could check in the non-xa datastore to see if the non-xa commit succeeded or not – if the in-doubt transaction ID was stored there, the commit happened, otherwise it didn't. On that basis, the rest of the resources could be recovered to their pre or post transaction states as appropriate.

In a world filled with nothing but xa resources, this LRO technique would be nothing more than an interesting curiosity, and most databases are xa compliant, so why am I boring you with this arcana? Well, many applications touch more than just databases – in these days of systems composed from a combination of bought and built functionality, most bespoke code accesses both databases (which are indeed generally xa compliant) and packaged applications (which generally are not). In fact, the whole point of the BEA WebLogic Platform is that most applications will involve elements of integration. In this scenario, to lever the maximum value out of the application server's transaction management capabilities, LRO would be a good feature.

A Little Bird Speaks of Clear Blue Water

As it turns out, a little bird told me that the next release of WebLogic Server is scheduled to support the small-window variant of LRO. When this is released, you will be able to build applications that use the transaction manager to protect data integrity across databases and an application view connected to an application through a J2EE CA adapter,

> "The whole point of the BEA WebLogic Platform is that most applications will involve elements of integration"

even if the underlying application doesn't support xa. Or distribute a transaction across databases and a nontransactional message bus, or across multiple databases where one doesn't support xa, via a new supercharged version of the WebLogic JTS driver.

Its features such as LRO, fitting as they do neatly beneath the J2EE API definitions without being specified by any of them, that sort the men from the boys and keep WebLogic Server ahead of the app server pack – a pack composed not only of competitive vendor's products but of open-source implementations and the like. J2EE support may be a commodity, in the same way as support for ANSI SQL in an RDBMS is, but it is in the application server beneath – the engine room of the modern enterprise – that commodity stops and clear blue water divides the best from the rest.

International

Web Services Edge 2003

CONNECTING THE ENTERPRISE WITH WEB SERVICES, JAVA, XML, AND .NET

March 18-20, 2003
Boston, MA

web services EDGE conference&expo

JDJ EDGE conference&expo

XML EDGE conference&expo

.NET EDGE conference&expo

Featured technologies and topics will include:

- Interoperability
- Enterprise Networks
- Securing Web Services
- Integrating existing networks
- Leveraging your existing software
- Real Time Web Services
- Where and When Should I Use Web Services?
- Web Services: From Consumption to Publication
- UDDI

Hynes Convention Center
Boston, MA

For more information visit
www.sys-con.com

Over 200 participating companies will display and demonstrate over 500 developer products and solutions.

Over 3,000 Systems Integrators, System Architects, Developers and Project Managers will attend the conference expo.

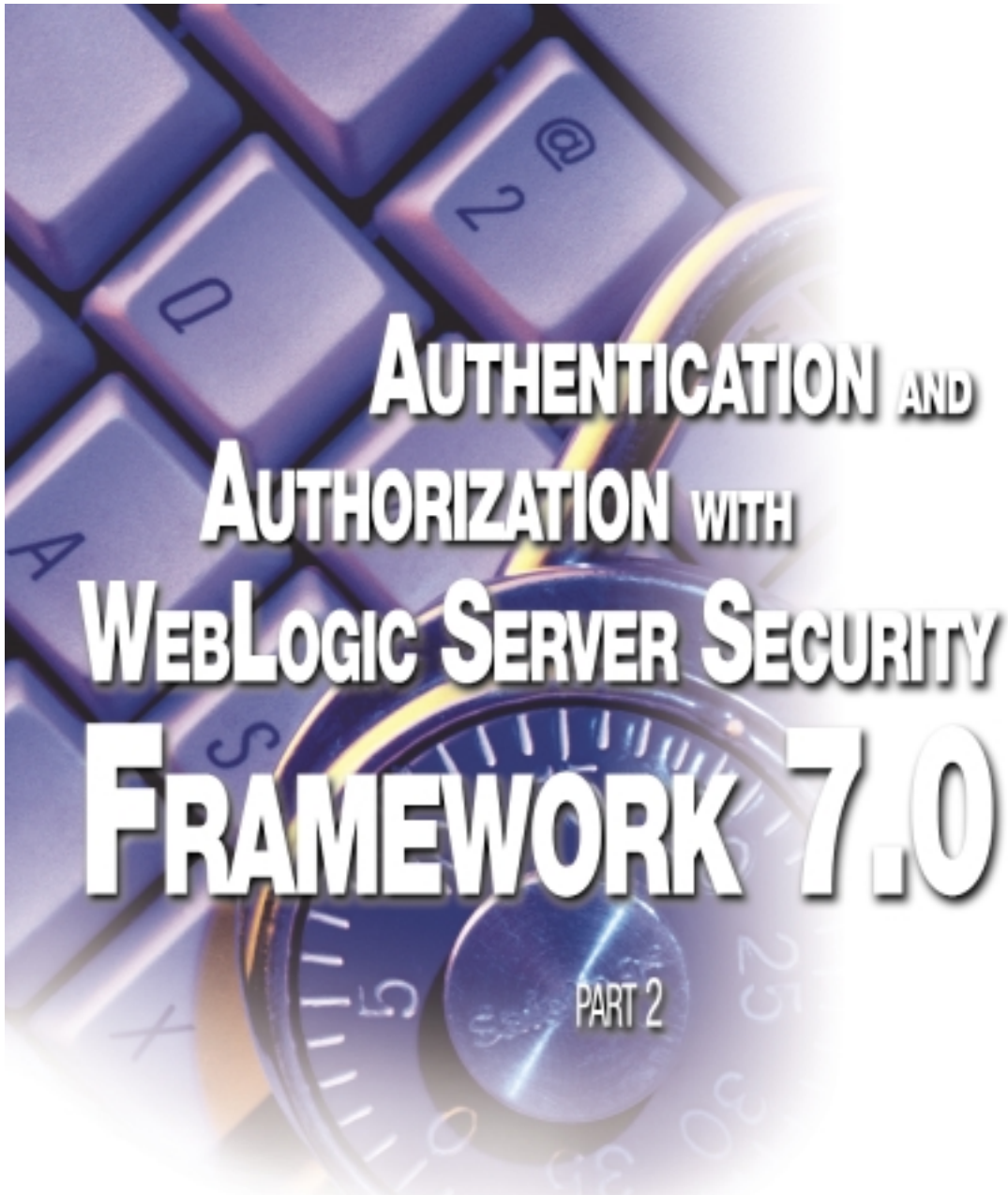
Over 100 of the latest sessions on training, certifications, seminars, case-studies, and panel discussions promise to deliver REAL Web Services Benefits, the industry pulse and proven strategies.

Contact information: U.S. Events: 201 802-3069 or e-mail grisha@sys-con.com • European & Asian Events: 011 44 208 232 1600

OWNED BY SYS-CON MEDIA
PRODUCED BY SYS-CON EVENTS

Boston call for papers opens November 11, 2002.

Submit your papers online at:
www.sys-con.com/webservices2003east



BY **PAUL PATRICK**
& VADIM ROSENBERG

AUTHOR BIO...

Paul Patrick is the chief security architect for BEA. He was the architect of BEA's (and earlier Digital Equipment Corporation's) Object Broker CORBA ORB, coarchitect of WebLogic Enterprise (Tuxedo), and has been the security architect for WebLogic Server since version 6.0.

Vadim Rosenberg is the product marketing manager for BEA WebLogic Server. Before joining BEA , he spent 13 years in business software engineering, most recently at Compaq Computers (Tandem Division) developing a fault-tolerant and highly scalable J2EE framework.

CONTACT...

paup.patrick@bea.com

This article appears courtesy of BEA dev2dev Online.

In the last issue of **WebLogic Developer's Journal** (Vol. 1, issue 12) we looked at some of the major features and functional areas of the new Security Framework in WebLogic Server 7.0.

Now let's take a closer look at how WebLogic Server 7.0 implements the primary task areas of a security system, which are authentication (determining a user's identity as a valid user), authorization (determining a user's role or roles and computing the appropriate access privileges based on the policies in place), and other essential areas of the WebLogic Security Framework.

Authentication

Authentication is the process of ascertaining that an identity can be proven to be who it claims to be. In many security systems, the proof of identity is accomplished through the use of usernames and passwords. WebLogic Server 7.0 builds upon the authentication classes of the Java Authentication and Authorization Service (JAAS), which is a standard extension to the security in the Java Software Development Kit, version 1.3. The JAAS standard provides the support needed to submit a username and credential, typically a password, when authenticating a user to WebLogic Server. WebLogic Server goes beyond this by providing a link to legacy systems so a particular system's authentication credentials can

be transparently used by WebLogic Server.

In addition, the new architecture supports the concept of perimeter-based authentication, where the actual authentication process occurs at an application perimeter such as a Web server, firewall, or VPN, and outside of WebLogic Server. The architecture provides support for multiple "security token types" (e.g., Microsoft Passport, SAML Assertions, or tokens from third-party commercial authentication products). WebLogic Server 's support for perimeter-based authentication supports the ability to propagate security tokes over multiple protocols, such as HTTP, and IIOP-CSIv2.

By default, WebLogic Server 7.0 supports a number of different forms of authentication:

- Certificate based authentication directly with WebLogic Server using the SSL/TLS protocol
- HTTP certificate-based, proxied through an external Web server
- Username/password authentication

WebLogic Server's security architecture is flexible enough to benefit developers who have special authentication situations. It provides integration with a number of different authentication technologies, including NT domains, Active Directory and other commercial LDAP directory servers, and Unix security services. For example, an organization might prefer to use the Unix security system for user authentication. In this scenario, WebLogic Server can be configured to defer authentication of users to Unix.

Levels of Authentication

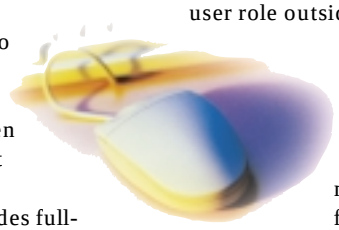
Since authentication requirements vary greatly, the server is easily configurable to allow different levels of authentication depending upon the needs of the deployment. When a client, administrator, or peer wants to access a resource hosted on WebLogic Server, it may be necessary to authenticate the user's identities. WebLogic Server provides two default mechanisms for authenticating users: passwords and digital certificates.

The most common form of authentication is password-based authentication, which allows users to enter a username and password as a means of authentication when requesting access to a given resource on the server. Password-based authentication is defined as Basic authentication by the HTTP protocol and the basis of form-based authentication. This form of authentication is very easy to implement and manage, but provides a weak form of authentication due to the lack of enforcement of good password practices..

For stronger authentication, WebLogic Server supports the use of digital certificates. These certificates are issued from a trusted third-party called a Certificate Authority (CA). A CA issues certificates that can be used to prove the identity of servers or users. WebLogic Server does not restrict the number of CAs or the number of levels in a certificate chain. This provides more flexibility for businesses that choose to be their own CA.

Authentication via digital certificates is the key to authentication using the Secure Sockets Layer (SSL) and Transport Level Security (TLS) protocols supported by WebLogic Server 7.0. WebLogic Server is capable of utilizing digital certificates that are compatible with the X.509v3 international digital certificate standard. This compatibility allows use of certificates from any CA that supports this standard. Both the SSL and the TLS security protocols support the concept of mutual authentication.

In order for mutual authentication to be used as the means to authenticate the client, both client and server must present a digital certificate that indicates their identities before the connection is enabled between the two. When certificate-based authentication is used, the identity from the client's digital certificate can be validated through a customization point called a certificate authenticator that is registered with an Identity Asserter. The Identity Asserter creates a JAAS CallbackHandler object that is populated with the necessary information to map the identity asserted by the digital certificate to a WebLogic Server local user identity. The CallbackHandler is then passed to a JAAS LoginModule where the asserted identity in the CallbackHandler is verified to be one of a local registered user. If the verification is successful, the identity is then inserted as a Principal object into a JAAS Subject.



WebLogic Server 7.0 provides full-strength, 128-bit encryption for customers in the United States and Canada, and international customers that are permitted to use full-strength encryption by U.S. trade laws. A 56-bit encryption scheme is compatible with "low-strength" encryption, allowing universal compatibility.

Authorization

Authorization determines what a user is allowed to access. J2EE defines the use of a role-based approach for container-based resources where a user must hold one or more roles in order to access the resource. WebLogic Server takes the role-based security model defined by J2EE and couples it with the ability to use policy to define the mapping of principals to roles and rules that specify the criteria that must be met to access a resource as a means to create a powerful and consistent authorization model.

The role mapping and authorization policies of WebLogic Server determine the level of access a particular user has to all resources hosted on WebLogic Server, for example:

- Enterprise JavaBeans
- Web applications (servlets, JSP)
- Miscellaneous J2EE resources (JCA adapters, JMS destinations, JNDI, JDBC)
- Non-J2EE resources like Web services

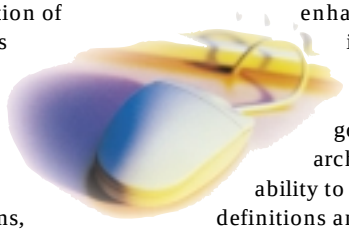
It's important to remember that the authorization model currently defined by the J2EE standard is not uniform across all of the resources. The declarative security described in deployment descriptors only covers limited J2EE resources like EJBs, servlets, and JSPs. Many J2EE vendors provide implementations of the J2EE authorization scheme that are static in that they have to be set through J2EE deployment descriptors before the application is deployed, and cannot be changed afterwards without modifying the deployment descriptor and potentially having to redeploy the application. Previous versions of WebLogic Server supported this model as well. In addition, there are currently no provisions in the J2EE authorization model to implement any kind of business access rules other than those purely based on the user role outside of the application itself.

The separation of security enforcement in the application from the business logic simplifies security management and makes the application less fragile to changes in authorization and business policy. The authorization capabilities provided by the new security architecture also allow business parameters to optionally be taken into account to authorize a specific user's access to a specific resource under specific conditions. This allows the role-mapping and authorization policies to support more "real-life" conditions required for doing business.

FOR ALL LEVELS OF USERS

Role definitions are specific to a unified security realm that represents the broadest policy scope to which security policies apply (e.g., the entire company, the company's Western Division, the company's Pacific Rim branch). J2EE defines that roles can be defined as *resource scoped* – associated with given resources. WebLogic Server 7.0 extends this model with support for *globally scoped roles* that are associated with all resources. Roles can be assigned statically for specified users or groups at administration time, or dynamically, based on the context of the request at run-time.

Role assignments are dynamically computed according to security policies, set by the administrator using the WebLogic Server Policy tool as the user seeks to access various resources. This new authorization scheme replaces the use of Access Control Lists (ACLs) for protection of non-container–based resources that was used in previous releases of WebLogic Server.



WebLogic Server allows the concept of roles to be implemented as an identity, or as a named collection of permissions, thus creating a 'capabilities-based' authorization system. Because the implementation of roles can be different depending upon the implementation, it is necessary for there to be a tight coupling between role and authorization decisions.

While J2EE-required security support is fully available with WebLogic Server, the power of the new Security Framework is in dynamic authorization and role mapping rules that are based on context, such as request parameters. The authorization and role-mapping policies can be defined by a security administrator through the use of the WebLogic Server administrative console and without having to edit XML deployment descriptors or involving a developer in order to change application business logic, and with no application redeployment required. Because it is often necessary for security policies to be changed dynamically as new users are added or situations change, WebLogic Server can dynamically update users and roles by leveraging an external, centralized security database or service, such as LDAP, or by using the embedded directory server.

In order to provide a consistent approach to authorization and role mapping across all resources hosted on WebLogic Server, the various containers within WebLogic Server delegate all authorization and role decisions to the new security framework. The authorization and role-mapping decisions are then delegated to the Service Providers that can be plugged into the security framework by implementing the appropriate Service Provider Interfaces. This mechanism allows the integration of commercial security products so that WebLogic Server can be customized to take advantage of a customer's existing security infrastructure rather than forcing the customer to change.

Preservation of the developer's experience and allowing J2EE applications that were initially developed for other J2EE implementations to take advantage of the enhanced security functionality found in WebLogic Server 7.0 are critical. As a means to address these goals, the new security architecture supports the ability to consume all previous definitions and assignments contained in deployment descriptors, yet still lets administrators manage role definitions and assignments without editing deployment descriptors.

Security Auditing

WebLogic Server 7.0 can be configured to allow administrators to construct security audits for their systems. By default, it is configured to audit authentication attempts, access attempts, security management modifications, and other significant security events. It is not only possible to see which user was attempting access to a specific resource, but also to find out the parameters of that request. Administrators can use the BEA-supplied Auditing plug-in, which creates an audit record for login and access failures along with security management changes. More sophisticated auditing schemes can be implemented using third-party auditing tools. As with the other security provider plug-ins, administrators can configure multiple Auditing plug-ins and set various conditions for their use. This offers administrators fine-grained control over when and whether to take action on an event.

"An application server security solution that … can assure the confidentiality, integrity, and availability of the server and its data"

Connection Filter

WebLogic Server 7.0 also provides administrator control over permissions to establish a connection to the server. The Connection Filter feature allows enforcement of rules that govern whether a connection should be established based on client IP access or protocol. This capability can be used to restrict the location from which connections to the application server are made, enhancing security of the application. Connection Filter features are configurable from the Administration Console.

Attack Counter Measures

In enterprise applications, denial-of-service and other attacks can render significant damage. WebLogic Server is designed to address certain forms of denial-of-service attacks. From the Administration Console, the administrator can set timeouts and other parameters that will counteract attempts to attack the integrity and availability of the application running on it. In addition, WebLogic Server provides measures to counter both online and offline guessing of users and passwords. Offline password protection is protected through use of encryption and other security techniques specifically targeted to counter dictionary attacks. Counter measures for online password guessing include the ability for administrators to control the number of times a login attempt can fail within a given period of time before the account is disabled or locked out.

Support for Java Standards

WebLogic Server 7.0 utilizes the security services of JDK version 1.3 for the Java 2 Platform, Enterprise Edition (J2EE). Like the other J2EE components, the security services are based on standardized, modular components. WebLogic Server utilizes these Java security service methods according to the standard, and adds extensions that handle many details of application behavior automatically, without requiring additional programming.

WebLogic Server's support for JDK 1.3 security means that application developers can take advantage of Sun Microsystems' latest enhancements and developments in the area of security, thus leveraging a company's investment in Java programming expertise. By following the documented Java standard, WebLogic Server's security support has a common baseline for Java developers. The innovations that WebLogic Server provides rest on the baseline support for JDK 1.3.

International

Web Services Edge 2003

CONNECTING THE ENTERPRISE WITH WEB SERVICES, JAVA, XML, AND .NET

March 18-20, 2003
Boston, MA

web services EDGE conference&expo

JDJ EDGE conference&expo

XML EDGE conference&expo

.NET EDGE conference&expo

Featured technologies and topics will include:

- Interoperability
- Enterprise Networks
- Securing Web Services
- Integrating existing networks
- Leveraging your existing software
- Real Time Web Services
- Where and When Should I Use Web Services?
- Web Services: From Consumption to Publication
- UDDI

Hynes Convention Center
Boston, MA

Boston
March 18-20

London
June 3-5

Berlin
June 24-26

Hong Kong
Coming soon...

For more information visit
www.sys-con.com

Over 200 participating companies will display and demonstrate over 500 developer products and solutions.

Over 3,000 Systems Integrators, System Architects, Developers and Project Managers will attend the conference expo.

Over 100 of the latest sessions on training, certifications, seminars, case-studies, and panel discussions promise to deliver REAL Web Services Benefits, the industry pulse and proven strategies.

Contact information: U.S. Events: 201 802-3069 or e-mail grisha@sys-con.com • European & Asian Events: 011 44 208 232 1600

OWNED BY SYS-CON MEDIA
PRODUCED BY SYS-CON EVENTS

Java Developer's Journal · WebServices Journal · XML Journal · .NET Journal · SAMS · OASIS · Java World · XML Journal · WebSphere Developer's Journal · WebLogic Developer's Journal · wireless Business Technology · LINUX Business Week · SF Advisor · ColdFusion Developer's Journal · PowerBuilder Developer's Journal

Boston call for papers opens November 11, 2002.

Submit your papers online at:
www.sys-con.com/webservices2003east

JDK 1.3 Security Packages

THE JAVA SECURE SOCKET EXTENSION (JSSE)

JSSE is a set of packages that support and implement the SSL and TLS v1 protocol, making those protocols and capabilities programmatically available. WebLogic Server 7.0 provides Secure Sockets Layer (SSL) support for encrypting data transmitted across WebLogic Server clients, and other servers.

JAVA AUTHENTICATION AND AUTHORIZATION SERVICES (JAAS)

JAAS is a set of packages that provide a framework for user-based authentication and access control. WebLogic Server uses only the authentication classes of JAAS, as the authorization function is implemented through access control features described earlier in the Authorization section.

THE JAVA SECURITY MANAGER

JSM is the security manager for the Java Virtual Machine (JVM). The security manager works with the Java API to define security boundaries through the java.lang.SecurityManager class, enabling programmers to establish a custom security policy for their Java applications.

Unified Administration for Security Services

WebLogic Server 7.0 supports unified administration of security services, through the Administration Console, for third-party and customized security services, as well as for BEA's implementations. The Administration Console is a security command center, providing tabs and tools for functions such as the Role Mapper, the Auditor, the Authenticator, the Authorizer, the Credential Mapper, and the Key Store. For compatibility with earlier WebLogic Server 6.x environments, the Administration Console also provides a Realm function.

With WebLogic Server 7.0, the Administration Console becomes the central (and the best) tool for creating and modifying the security sections of configuration files and deployment descriptors. Configuration files no longer have clear text passwords. WebLogic Server automatically creates protected passwords when it writes this information to the configuration file. Passwords are created on a per-realm basis.

The Administration Console also supports configuration and management of the Keystore that stores public and private keys, and of the security store (an embedded LDAP directory server). The Auditing plug-in that keeps a trail of security-related activities performed by users and applications is configured using the Administration Console.

Security Management and Storage

Security management in the Administration Console is implemented using the classes of the Java Management Extensions (JMX) package. JMX supports both console and programmatic access to management functions via Management Beans (MBeans). Plug-ins can extend base MBeans to meet their needs. This means that any security function in the Administration Console can be extended. Third-party security vendors or security-specific application programmers can add or replace pages in the Administration Console as desired.

Also, through a full-power command line or programmatic control, administrators and programmers can develop command scripts or programs that execute multiple security-related operations with WebLogic Server without having to use the Administration Console. While convenient and well suited for individual administration operations, the Administration Console can be too slow for repeatable operations, especially if those can be triggered by some event and need to be executed programmatically.

Security storage is provided by the Key Store and by the WebLogic Server system data store. The Java standard key store stores public and private keys with encrypted storage. The WebLogic Server system data store is an embedded LDAP directory server optimized for use within WebLogic Server 7.0. The system data store provides unified storage (per security realm) of security information and proof material for users, groups, roles, and policies. Customers can also use an existing LDAP directory service for retrieval of storage of user and group information.

> "Like the other J2EE components, the security services are based on standardized, modular components"

Migrating from WebLogic Server 6.x Security to 7.0 Security

Customers who installed security under WebLogic Server 6.x have three choices for transitioning to WebLogic Server 7.0:

- ***Running in "native" mode:*** Converting completely to WebLogic Server 7.0 security
- ***Running in "compatibility" mode using the Realm Adapter:*** Supports only existing 6.x security features
- ***Running in "mixed" mode:*** Using the Authorization service of WebLogic Server 7.0 while retaining the 6.x authentication service

Because of the significant changes in how security is provided in WebLogic Server 7.0, a set of specialized adapters, called "realm adapters," are also provided that allow security realm implementations from previous releases of WebLogic Server to be utilized with the new security framework. The realm adapters permit the use of authentication and authorization capabilities found in WebLogic Server 6.x realm implementations and currently are used to support integration with NT domains and Unix security systems.

An Export utility (not part of WebLogic Server 7.0, but provided on BEA dev2dev Online at http://dev2dev.bea.com) enables customers to extract users (but not passwords), groups, and ACLs from WebLogic Server 6.x and import the information into the embedded LDAP Directory Server in WebLogic Server 7.0.

Summary

WebLogic Server's open, flexible security architecture delivers advantages to all levels of users and introduces an advanced security design for application servers. WebLogic Server 7.0 provides out-of-the-box implementations for each of the security SPIs. Companies now have an application server security solution that, together with clear and well-documented security policies and procedures, can assure the confidentiality, integrity and availability of the server and its data. As a result,

1. Security policies are created and managed by Security Administrators
2. Security policies are flexible, dynamic, powerful rules that can be changed without recoding and redeployment
3. Integration with existing security solutions is greatly simplified

ATTN: Developers

STEP UP to the mik and be... HEARD!

Go to
http://developer.sys-con.com

Calling Sleek Geeks Everywhere!

Make sure you have your finger on the pulse of i-Technology...bookmark http://developer.sys-con.com today.

i-Technology
News

i-Technology
Views

i-Technology
Comment

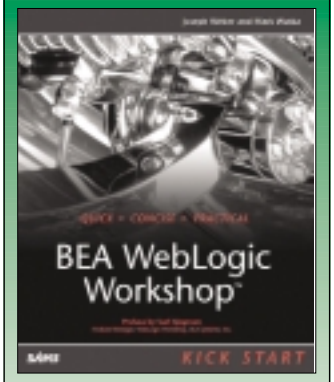
i-Technology
Debate

SYS-CON MEDIA

BEA WebLogic Workshop Kick Start

GET A KICK START ON BEA WEB SERVICES!

REVIEWED BY **JASON WESTRA**



Publisher:
Sams Publishing, 2002

ISBN: 0-672-32417-2

SRP:
$34.99

Authors:
Joseph Weber, Mark Wutka with Foreword from Carl Sjogreen, product manager for WebLogic Workshop

To order this book, please visit www.samspublishing.com

I learned about WebLogic Workshop in December of 2001 while interviewing BEA CTO Scott Dietzen. At the time, it was code-named Cajun and, according to Scott, the tool would revolutionize Web services and J2EE development. Cajun has since been renamed BEA WebLogic Workshop and become an integral part of BEA's strategy, which focuses on making J2EE and Web services easy for beginners as well as experienced developers.

In *BEA WebLogic Workshop Kick Start* (hereon referred to as Workshop Kick Start), authors Joseph Weber and Mark Wutka have written a book that harmonizes with BEA's vision and will surely become an integral part of any effort utilizing Web services on BEA's platform. The book focuses on Web services developers who range in Java and J2EE experience from beginner to expert. Workshop Kick Start has nice appendices for reference on common Java language constructs, XML basics, WSDL (Web Service Description Language), and SOAP, so the essential technologies for getting started with Web services on WebLogic are covered.

The book is divided into three parts. The first section introduces you to Web services WebLogic Workshop. The second covers building applications with features such as controls, debugging, and database access, essentials in most enterprise-level Web services. Finally, Workshop Kick Start walks you through advanced features offered by WebLogic Workshop such as accessing EJBs, asynchronous communication via JMS, and accessing your Web services from different clients. It finishes up with an online ordering application to show a fully operational application built on WebLogic Workshop.

Weber and Wutka do a fine job of introducing each section and laying the foundation for the topics covered. For instance, in section one they cover the advantages and disadvantages of synchronous versus asynchronous Web services with neutrality, even though BEA's focus is on asynchrony for its integration and reliability qualities. Following the discussions, you are led through basic examples utilizing nearly every feature of WebLogic Workshop.

Although not an exhaustive list, the WebLogic Workshop features covered in Workshop Kick Start include creating and customizing projects; BEA's Java Web Service files (.jws); using Workshop controls such as the Timer, Database, Service, JMS, and EJB control; and developing, testing, and debugging within WebLogic Workshop's Design and Source Views.

Workshop for Everyone

Experienced developers often fear that development tools, which hide complexity, will compromise their control over the application. Just to eliminate the skepticism here and now, I will say that BEA WebLogic Workshop has done an awesome job of hiding the complexity in a nonintrusive fashion. For instance, the concept of "controls" to access J2EE constructs such as sending and reading from JMS Topics and Queues, or accessing EJBs, makes even inexperienced J2EE developers effective at their tasks. However, these controls are described via @JavaDoc tags or metadata in comments, which is a nonintrusive means of describing the attributes of the controls. Although property sheets in the Design View configure the Web service visually, experienced developers can code in the Source View and simply type in the metadata tags to configure controls without the visual Designer. Workshop Kick Start walks you through exactly how to utilize both the Designer View for visual development as well as the hard-core @JavaDoc tags that the experienced developers will choose.


THE INSIDER INTELLIGENCE YOU NEED...
TO KEEP AHEAD OF THE CURVE
FREE E-Newsletters
SIGN UP TODAY!
Go to www.SYS-CON.com
The most innovative products, new releases, interviews, industry developments, and plenty of solid i-technology news can be found in SYS-CON Media's Industry Newsletters. Targeted to meet your professional needs, each e-mail is informative, insightful, and to the point. They're free, and your subscription is just a mouse-click away at www.sys-con.com.
SELECT THE INDUSTRY NEWSLETTERS THAT MATCH YOUR NEEDS!
CHOOSE ONE – OR TRY THEM ALL!
Java Developer's Journal Industry Newsletter
WebLogic Developer's Journal Industry Newsletter
WebServices Journal Industry Newsletter
Unix Review Industry Newsletter
XML Journal Industry Newsletter
ColdFusion Developer's Journal Industry Newsletter
wireless Industry Newsletter
Java Developer's Journal Industry Newsletter
Don't Delay!
Subscribe for FREE!
at www.sys-con.com
Exclusively from the World's Leading i-Technology Publisher
SYS-CON MEDIA

BOOK REVIEW

Workshop Examples Abound

Workshop Kick Start comes with an install CD, so don't waste your time downloading the over 280MB WebLogic Platform 7.0 from BEA's site. Instead, just pop in the CD, install WebLogic Platform 7.0, and request an evaluation license from BEA. It comes with BEA EvalNet to support you during your evaluation of the software!

To help guide you through WebLogic Workshop's features, the book incorporates examples both from the standard WebLogic 7.0 install and custom examples from the authors themselves. Don't rely on the book for the location of the sample code! It says the code exists on Sams Publishing's Web site, but it isn't there. Instead, save your Internet bandwidth once again and just copy it from the install CD provided with the book.

Once you have everything installed, you are ready to work through the book's examples chapter by chapter. I found the examples to be a great introduction to the features of WebLogic Workshop, but as an experienced developer I wanted more. Granted, the book provides you with everything you need to develop Web services, ranging from simple to complex. However, WebLogic Workshop offers much more than what the book covers. I recommend you also refer to the documentation as you work through the exercises in the book. This documentation may shed light on outstanding questions you have, like "Why doesn't my TopicReceiver.jws receive any messages from my JMS Topic?" Well, maybe it's because the example code in the book missed a step. It doesn't show you how to subscribe to the Topic. That aside, the examples are quite intuitive and function just fine in the Workshop Test View.

IMHO

Get this book if you are going to work with Web services on BEA WebLogic Platform 7.0. BEA is focusing its energy on making J2EE and Web services development easy and Workshop is central to this strategy. If you are an experienced J2EE developer with no Web services experience, you'll be able to develop complex BEA WebLogic Workshop Web services in a few days. If you are new to both J2EE and Web services, you'll be productively accessing prebuilt J2EE components in a matter of hours. To that end, I salute the authors of *BEA WebLogic Workshop Kick Start* for a fine book and BEA for a great Web services and J2EE development environment.

WEB SERVICES

– continued from page 8

The portal in our example segregates trouble ticket assignments by specialty and seniority using the Unified User Profile. Engineers in a particular specialty designated as a manager become members of that specialty's Management Entitlement Segment, allowing them access to Engineer Case History Portlets. These portlets allow a manager to analyze a particular engineer's job performance based on his or her handling of individual cases over time. As described earlier, the EntityPropertyManager EJB in this example may use JAX-RPC to get at our profile information, exposed as a Liquid Data Web Services View.

Web Services Portlets: Web Tier Integration

Web Services Portlets, as the name suggests, consume Web services and display their results as content. These portlets can be developed quickly using the Portal EBCC Portlet Wizard and come in several varieties, from the very basic to those providing dynamic, asynchronous interactions with user-defined data types. Web services portlets can also take part in Workshop-style conversations.

The Model 2 Web Tier Architectural Pattern is followed today in most well-designed Web applications. Apache Jakarta's "Struts" is in widespread use as a framework that enforces this approach. The Portal's Webflow/Pipeline Framework works in a similar fashion. The basic tenet of the Model 2 pattern is a separation of concerns (controller, model, and view) and the main concern of the "view" (a portlet in our case) is to display the current state of its associated model(s). The invocation and consumption of one or more Web services directly from a portlet would seem to violate this tenet, and in many cases it does. There are situations, however, where they are very appropriate:

- The portlet is standalone (really a separate "mini-application").
- The Web service is providing the current state of the model (the View Helper strategy of the Front Controller J2EE Design Pattern).
- The result of the Web service invocation is the fully formatted Portlet User Interface itself.

In our prototypical example, a support engineer has chosen to receive a tech alert feed from an RDBMS vendor that their system often interfaces with. This tech bulletin feed is displayed as a separate portlet within their portal. It is a standalone service, unrelated to any other portlet in our portal, and the information is retrieved from an external, Web services–enabled source. Another overriding reason that a Web services portlet fits the situation here is that the information returned from this Web service is the user interface.

Summary

This article introduced a set of capabilities within the WebLogic Enterprise Platform that together provide some powerful new integration options available to you when architecting an Enterprise portal solution. The intent was not to present a single, all-encompassing architecture (a sort of low-rent "Pet Store" for portal integration), nor was it to imply that the use of any one set of tools and techniques is necessarily the right approach in every situation (or even, perhaps, most situations). There are simply too many factors to take into account for a given environment to deal in absolutes.

Enabling EAI for portal integration via Web services, when implemented judiciously, can result in a very flexible architecture. When implemented without carefully considering the larger picture, however, important issues (performance, scalability, security and transactional interoperability, etc.) can easily fall through the cracks. This is, of course, only as true for Web services as it is for any other technology. The experienced architect understands this and is swayed neither by the Web services hype nor its corresponding backlash.

WebServices JOURNAL
.NET J2EE XML

The Best .NET Coverage Guaranteed!

LEARN WEB SERVICES. GET A NEW JOB *!*

Subscribe today to the world's leading Web Services resource

Get Up to Speed with the Fourth Wave in Software Development

- **Real-World Web Services**: XML's Killer App*!*
- How to Use **SOAP** in the Enterprise
- Demystifying **ebXML** for success
- **Authentication**, **Authorization**, and **Auditing**
- **BPM** - Business Process Management
- Latest Information on **Evolving Standards**
- Vital technology **insights** from the nation's leading Technologists
- Industry **Case Studies** and **Success Stories**
- Making the Most of **.NET**
- **Web Services Security**
- How to Develop and Market Your Web Services
- **EAI** and Application Integration Tips
- **The Marketplace**: Tools, Engines, and Servers
- Integrating **XML** in a Web Services Environment
- **Wireless**: Enable Your **WAP** Projects and Build **Wireless Applications** with Web Services*!*
- Real-World **UDDI**
- **Swing**-Compliant Web Services
- *and much, much more!*

Only $69.99 for 1 year (12 issues)*

* Newsstand price $83.88 for 1 year

Subscribe online at
www.wsj2.com or
call 888 303-5252

*Offer subject to change without notice

SYS-CON MEDIA

SYS-CON Media, the world's leading *i*-technology publisher of developer magazines and journals, brings you the most comprehensive coverage of Web services.

A continuous integration process involves automatic building and testing with tools like Ant and JUnit. These tools need a tool that will manage and control the build process – this is where Cruise Control comes in.

CruiseControl: The Complete Continuous Integration

AUTOMATING A WLS BUILD AND TEST ENVIRONMENT

BY **RONEN LEWIT**

AUTHOR BIO...

Ronen Lewit, CEO of Risotech LTD, is an IT expert specializing in enterprise software architecture and development. Risotech LTD provides services for software developers worldwide, bridging the technological gaps and providing the know-how in exploring and building robust state-of-the-art infrastructures and products.

CONTACT...

ronenl@myrealbox.com

CruiseControl, an open source tool from SourceForge, is a framework for a continuous build process. It provides an easy-to-use tool that integrates Ant, JUnit, and others to provide a simple utility that builds and tests your software throughout the development cycle.

For a complete build process you might want to clean the build directory, fetch sources from the source configuration tool, compile sources, package applications, start the application server, deploy the packaged applications, perform tests, generate reports, and clean up. Simpler (intermediate) build processes can pass the clean task and part of the test tasks.

All of these tasks should be part of your Ant build configuration.

For an easy transition between different dev/test/prod environments it's helpful if you use different configuration settings for your application properties and application server settings and let Ant prepare these settings for you. For example, for the build.xml:

```
<filter token="server.domain"
 value="${server.domain}"/>
<filter token="server.listen.port"
 value="${server.listen.port}"/>
<filter token="server.name" value="${server.name}"/>
```

And for the config.xml:

```
<Domain Name="@server.domain@">
 …
     <Server ListenPort="@server.listen.port@"
Name="@server.name@">
```

where all server.* properties reside in different configuration files for use with any different environment (dev.properties, test.properties, etc.)

This will put you in a position where you could run continuous build process without human hands.

Now, let's see how CruiseControl helps us with these builds.

CruiseControl, release 2.0, is more flexible and open. Its building blocks are plug-ins (that you can always extend or replace). It is shipped with several plug-ins that, in most cases, will do the work without the need for an update. The provided plug-ins are grouped into five types:

- ***Bootstrappers:*** Designed to be run before the build, regardless of whether a build is necessary. These trappers are used for updating the configuration files from the source configuration (like build.xml, which you'll use just as you start the build…).
- ***SourceControl implementations:*** Poll a code repository and determine if anything has changed within your project that would require another build. These plug-ins can be configured not to fetch changes until *X* minutes have passed since the last source update (useful for large source updates).
- ***Builders:*** Actually perform the building and testing of your code. These plug-ins are responsible for starting an Ant task based on your configuration of Ant. These tasks will run in a different process than the CruiseControl (eliminates memory problems and provides better control over the Ant usage).
- ***LabelIncrementers:*** Handle incrementing the label for you to use to tag your source, as part of the continuous process.
- ***Publishers:*** Publish the results of your build. Plug-ins such as HTML-based e-mail, link-ref e-mail, and plain HTML are possible output formats.

CruiseControl generates a log file for each

THE WORLD'S LEADING INDEPENDENT WEBLOGIC DEVELOPER RESOURCE

Helping you enable intercompany collaboration on a global scale

- **Product Reviews**
- **Case Studies**
- **Tips, Tricks and More!**

SPECIAL OFFER! SAVE $31*

OFFER EXPIRES NOV 30, 2002

Now in More Than 5,000 Bookstores Worldwide – Subscribe *NOW!*

Go Online & Subscribe Today!

***Only $149 for 1 year (12 issues) – regular price $180.**

SYS-CON MEDIA

WebLogicDevelopersJournal.com

SYS-CON Media, the world's leading publisher of *i*-technology magazines for developers, software architects, and e-commerce professionals, brings you the most comprehensive coverage of WebLogic.

build process. This log file can be used to follow the build results as well as publish them using the publishers.

CruiseControl comes with a prepared Web application that enables you to publish your build results on an intranet site. In order to install the Web application, you should follow the installation instructions and pay special attention to the replacement of the log path element in the web.xml file and the correct JSP version usage in the buildresults.jsp file.

After you have configured the Web application, you can install it on your target intranet Web server. Make sure this server shares a file system with the process, so the CruiseControl process can update this Web application. The generated report can include information about:

1. List of changed sources and the check-in notes
2. Building blocks of the application (.jars, .ears, etc.)
3. JUnit reports (to integrate JUnit reports into the standard CruiseControl reports, use the Merge plug-in to append the JUnit log files to CruiseControl log file)

The CruiseControl Web application simply takes the log file generated by the build process and transforms its format for a clean, hyper-text format.

Example

Listing 1 is an example of a CruiseControl configuration file that should satisfy most projects. Table 1 defines the main terminology in CruiseControl

Resources

- *CruiseControl:* http://cruisecontrol.sourceforge.net
- *Jakarta Ant:* http://jakarta.apache.org/ant
- *JUnit:* www.junit.org
- Fowler, Martin. Continuous Integration. www.martinfowler.com/articles/continuousIntegration.htm

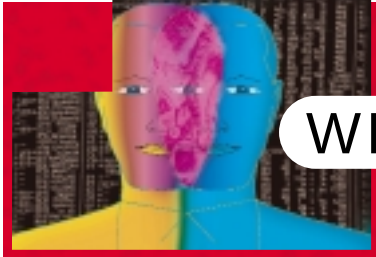
TABLE 1

TAG NAME	DESCRIPTION
<project>	Basic unit of work for CruiseControl.
<bootstrappers>	Lists all the bootstrappers in use for this project.
<currentbuildstatustrapper>	Provides a simple txt file to indicate the current status of the build process.
<modificationset>	Sets the source for the build sources (in this example CVS) and the "quiet period" (as mentioned before).
<schedule>	Sets timing for performing Ant tasks. Mention the multiple attribute, it states that clean-build will be done once in five intervals. You can add as many Ant tasks as you like.
<log>	Sets the logging directory as well as the merging file/directory (for usage with JUnit as mentioned before).
<publishers>	Sets the type of publishers.
<currentbuildstatuspublisher>	Reports about the current build status using the Web application.
<email>	Generates an e-mail message with a link to the current build page at the Web server.

Terminology

Listing 1

```
<cruisecontrol>
   <project name="myProject">
      <bootstrappers>
         <currentbuildstatusbootstrapper
file="logs/currentbuildstatus.txt"/>
      </bootstrappers>
      <modificationset quietperiod="30" dateformat="yyyy-MMM-dd
HH:mm:ss">
         <cvs cvsroot=":pserver:myusername@myhost:/usr/local/cvsroot"
               defaultbranchonly="" localworkingcopy=""
               tag="" property="" propertyondelete=""/>
      </modificationset>
      <schedule interval="30" intervaltype="relative">
    <ant file="build/build.xml" target="continuous.clean-build" multi-
ple="5">
          </ant>
    <ant file="build/build.xml" target="continuous.complete-build" multi-
ple="1">
          </ant>
      </schedule>
      <log dir="logs">
         <merge dir="/temp"/>
      </log>
      <publishers>
         <currentbuildstatuspublisher
file="logs/currentbuildstatus.txt"/>
         <email mailhost="myHost" returnaddress="ronen@nospam.please"
           defaultsuffix="" reportsuccess="" spamwhilebroken=""
           buildresultsurl="http://continuous-server:7001/">
            <always address="managers@nospam.please"/>
            <failure address="developers@nospam.please"/>
            <map alias="" address=""/>
         </email>
      </publishers>
      <plugin name="cvs" classname="net.sourceforge.cruisecontrol.source-
controls.CVS"/>

      <plugin name="currentbuildstatusbootstrapper"

classname="net.sourceforge.cruisecontrol.bootstrappers.CurrentBuildStatusB
ootstrapper"/>

      <plugin name="ant"
classname="net.sourceforge.cruisecontrol.builders.AntBuilder"/>

      <plugin name="email"

classname="net.sourceforge.cruisecontrol.publishers.LinkEmailPublisher"/>

      <plugin name="currentbuildstatuspublisher"

classname="net.sourceforge.cruisecontrol.publishers.CurrentBuildStatusPubl
isher"/>

      <plugin name="labelincrementer"

classname="net.sourceforge.cruisecontrol.labelincrementers.DefaultLabelInc
rementer"/>
   </project>
</cruisecontrol>
```



WLDJ interview

A Look at dev2dev

Diana Reid

Jill Steinberg

Mary Haggard

THE DEV2DEV TEAM OFFERS THE INSIDER SCOOP ON BEA'S DEVELOPER WEB SITE

We recently had a chance to talk to just some of the team behind BEA's dev2dev portal and program. ***Diana Reid*** is senior director of Developer Marketing. Before joining BEA, she spent six years at Microsoft in a variety of developer-focused roles. ***Mary Haggard*** is senior director of dev2dev Online. Mary came to BEA from the Westside acquisition in late 2001 and was also at Microsoft for several years. ***Jill Steinberg*** is editor in Chief of dev2dev Online. She was previously editor-in-chief of *JavaWorld*. As a whole, the entire development team has an average of five to seven years of developer-related experience across the industry.

WLDJ: Let's start with an overview of what dev2dev is, what it's doing, what the content is.

Reid: When we talk about dev2dev, we look at it from a higher level, rather than just the Web site, because for BEA, dev2dev is really our comprehensive developer program. It consists of three key areas, including the online site, community programs, and developer products. We're focused on delivering a variety of resources for developers with a number of different activities and programs in order to ensure their success with the BEA WebLogic platform.

It's really all about us reaching out to developers, building relationships with them, ensuring they have the tools and technologies that they need to get their jobs done, and helping them enhance and grow their skill sets.

Steinberg: It's really a whole support resource center for developers.

WLDJ: When was dev2dev actually introduced?

Reid: We launched the dev2dev online site, which is the cornerstone for the dev2dev program, in February 2002 at BEA eWorld.

WLDJ: And it's grown from there?

Reid: We have rolled out a number of new programs and offerings on a fairly regular basis since then. Things like the User Group Program have really ramped up in the last several months – we now have about 16 of these in the U.S., and will launch this program internationally in 2003. The dev2dev Days Road Show took place in 11 cities this fall, and smaller events will reach another 13 cities this winter.. We also have a monthly dev2dev webinar series, as well as other educational events and programs, in the works.

Haggard: I think one of the things that makes us very different from anything else in the industry is that we pride ourselves on being "by developers for developers." All of our user groups, all of our content, all of our programs, everything that we put together, is from developers who are either at BEA or are close to us but external. We also do a lot of research in the com-

munity to find out exactly what they need to be successful with BEA's products. That's something that is very important to us; we are always very tightly tied to our community and making sure that all the resources that we provide help meet their needs and get their jobs done faster.

Reid: We feel strongly that these are the people who are out in the trenches dealing with enterprise development and deployment issues, and have their finger on the pulse of what's happening with new technologies, what the enterprise development needs and requirements are. We want to make sure that we are listening to developers as we build support resources and feed this back into our product development team. These relationships and communications are really important because they will drive a lot of BEA's future product direction.

WLDJ: What kind of response have you had from the community since the launch, how many people have registered, and how long did it take to reach a number that you felt satisfied with?

Reid: Currently we have about 500,000 registered developers, definitely pretty exciting growth. We've had a couple of significant spikes – monthly run rates of 20% or more in growth. It's been a tough year in the industry overall for a lot of the vendors. So while we'd gladly welcome even greater growth, we are really excited about the response we've had in terms of new developer registrations, sold-out dev2dev Days attendance, rapidly expanding User Group membership, and Web site traffic – including unique visitors and time spent on the site.. We're also seeing our developer community contributing content to the site, as well as to new technical books such as the recently published *BEA WebLogic Workshop Kick Start* (Sams), *Teach Yourself BEA WebLogic Server 7.0* (Sams), and *WebLogic Server 7 Administration* (McGraw Hill) titles, so we are definitely thrilled about the feedback we're getting across the board. When you look at BEA's overall dev2dev program, beyond some of the hard metrics, such as number of registered members or site traffic, we've seen a lot of exciting things come out of our efforts.

WLDJ: Those are impressive numbers.

HAGGARD: In terms of the dev2dev Online site, the average time that people are spending on the site has actually quadrupled since February so a lot of the content that we have been adding is obviously useful. We see people visiting the site multiple times per month and that is also something that has been growing since we launched. All of the community aspects and the quality of information that we worked hard to build have been sitting well with our customers.

WLDJ: Tell us about the dev2dev User Group program, and how you're tying user groups into the online site.

Reid: We currently have 16 user groups - and each is founded and led by BEA customers and partners. BEA provides an infrastructure and set of resources that supports these groups, and then the regional leaders drive the content and meetings, based on what their particular group's interests and needs are.

In terms of where things fit, we're working on a number of ways that we can feed the regional offline groups back into the global online community. Some of the things that we're building out are "mini-communities" and pages on the dev2dev site for user groups, as well as creating online discussion forums to extend offline meeting topics. From a grassroots perspective, a lot of what we're doing with the user groups is talking directly with developers about our technologies. As an example, some of our engineering and development teams are working with the user groups to get their feedback on the next version of WebLogic Server or to understand how people are using WebLogic Workshop. We are also getting our user group leaders involved in ***WLDJ*** and in writing articles for dev2dev. And, we will bring all of the user group leaders together at eWorld 2003 to do some additional brainstorming on how the developer community can leverage some of these technologies and share best practices.

WLDJ: Tell us a little bit about how the content of the site is put together.

Steinberg: The majority of the content on dev2dev is unique to the site. We want to provide original information for developers that they won't necessarily find in other places and that is specific to BEA and partner technologies, as well as general J2EE programming tips. We have a section called the Code Library where we deliver code samples. We also offer full access to the BEA product documentation and evaluation downloads. In addition, our community section provides details on some of the programs and activities that Diana mentioned earlier, including user groups, developer events, the BEA University Program, newsgroups, and others.

WLDJ: Which areas are the most heavily trafficked?

Steinberg: We're finding that developers are extremely inter-

"We understand how these things work and what areas are a little more complicated than we might want them to be today, and are feeding that information back into the development team"

DEV TEAM OFFERS THE INSIDER SCOOP ON BEA'S DEVELOPER WEB SITE

"What our team is doing is pulling all of those resources together and making dev2dev into your 'one-stop shop'"

ested in getting sample code, product documentation, as well as training and troubleshooting information. They are also interested in the original technical articles we have – such as one we recently posted on migrating the dev2dev site from WebLogic 6.1 to 7.0.

WLDJ: Who are the primary people building dev2dev?
Haggard: Besides Jill and myself, we also have a senior technical content guy – Will Iverson, himself a J2EE developer – who looks at what the strategies for each of our progress areas should be in terms of the technical content we should cover. He works with the BEA teams and external writers to make sure we cover the product in-depth and highlight the right issues.

Our own Web site development team is also a key resource – a team of several developers led by a long-time J2EE veteran, Peter Ziatek – who build out the infrastructure and code behind the site. As I said before, "by developers for developers" is sort of our mantra, and our team works with all of the WebLogic Platform technologies every day, and does a lot of work to document the site. The articles that we've written detailing how we are using WebLogic technologies in our development efforts, or migrating from one version to the next, have been some of the most popular content on the site.

WLDJ: There are at least four groups that come to mind that can benefit from dev2dev: WebLogic developers, BEA partners, people who are not BEA users yet but are interested in understanding more about the product, and BEA itself. Let's say I work for a company and we have implemented BEA for the past two years. I register at dev2dev; what are some of the immediate benefits?
Haggard: The highest value offffering is access to our evaluation software; they can get full copies of all BEA products with very generous licenses to use for their development efforts. That's definitely our number one benefit. In addition, we offer a variety of premium content, special offers and discounts on development resources, publications and training content or software from different partners, and sample chapters from new books from key publishers.
Reid: Yes, as I mentioned previously, we have relationships with all the leading technical book publishers and have been working with them to get texts developed on a number of BEA technologies. We then offer sample chapters to developers who register in the other membership program, as well as some very steep discounts on purchasing those books.
Steinberg: An additional benefit of membership that sort of ties everything together is the monthly dev2dev Dispatch newsletter. This newsletter provides a snapshot of the most popular content on the site and highlights new events, programs, or content as well as special offers just for dev2dev members, such as discounts on training, publications, and conferences. This is a great way for developers to stay up-to-date on what we are adding to our site and what opportunities they have for information from the developer program.

WLDJ: Let's say that I have just become a BEA partner and I am a business development manager, or a product manager for one of the third-party vendors who integrates a product with BEA. What benefits does dev2dev have for BEA Star partners?
Reid: While BEA dev2dev isn't a direct subset of the BEA Star Partner Program, we do offer a variety of opportunities for partners to plug into what we're doing with dev2dev, including building technical content for the site, conducting joint webinars or training events, getting involved in regional user groups, and offering evaluation software and other tools for dev2dev members.

WLDJ: Let's say that my company is looking to migrate to BEA WebLogic. Where would I start?
Haggard: A user's first question is really, "Can your product solve my problems?" The second is, "How?" We work to answer those questions. One of the things we're finding out about people who come to the site is that most of them are new to WebLogic development. We've tried to focus our content on someone who is getting started. There are complete Getting Started guides, and the webinars do a great job of educating people in a live Q&A format before they start coding. The sample code is there to give them a leg up. We see a lot of new folks up in the newsgroups – and a lot of the BEA developers and staff are up there too to answer their questions and get them going.

WLDJ: In addition to the benefits you offer external groups – new users, current users, and partners – it sounds like there is also a benefit that BEA gets – the feedback that you can incorporate into the products.
Reid: I think you've hit the nail on the head. This is incredibly important to us and we're working to devise a lot more ways to streamline the communications and make sure that our engineers, our product management people, and our support people are hearing what the developers

DEV TEAM OFFERS THE INSIDER SCOOP ON BEA'S DEVELOPER WEB SITE

WLDJ ADVERTISER INDEX

ADVERTISER	URL	PHONE	PAGE
Altaworks	www.altaworks.com	603-598-2582	52
BEA Systems	http://dev2dev.bea.com/useworkshop	800-817-4BEA	9
BEA Systems	www.bea.com/liquiddata	800-817-4BEA	51
BEA e-World	www.bea-eworld.com	404-240-5506	2
Component Source	www.componentsource.com/bea	888-850-9911	3
EnginData Research	www.engindata.com		47
Hewlett-Packard	www.hp.com/plus_explorers	800-752-0900	21
Interwoven	www.interwoven.com	408-774-2000	15
LinuxWorld	www.linuxworldexpo.com	800-657-1474	22-3
ObjectFocus	www.objectfocus.com	408-973-8200	41
Performant Software Solutions	www.performant.com/weblogic1	866-773-6268	41
PowerBuilder Developer's Journal	www.sys-con.com/pbdj	888-303-5282	40
Sitraka (now part of Quest Software)	www.sitraka.com/performance/wldj	800-663-4723	11
Sitraka (now part of Quest Software)	www.sitraka.com/jclass/wldj	800-663-4723	19
SYS-CON E-Newsletters	www.sys-con.com	201-802-3020	35
SYS-CON Media	http://developer.sys-con.com	201-802-3020	33
SYS-CON Publications	www.sys-con.com	888-303-5282	26, 48-9
SYS-CON Reprints	www.sys-con.com	201-802-3026	47
Web Services Edge 2003	www.sys-con.com	201-802-3069	27
Web Services Journal	www.wsj2.com	888-303-5282	37
WebLogic Developer's Journal	www.weblogicdevelopersjournal.com	888-303-5282	39
Wily Technology	www.wilytech.com	888-GET-WILY	4
Wireless Developer's Journal	www.sys-con.com	888-303-5282	31
Yahoo	www.enterprise.yahoo.com/testdrive	866-267-7946	25

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

JAAS Fundamentals
by Bill Kemp and Rich Helton
Using the Java Authentication and Authorization Service to protect and validate your resources.

Writing A Custom Jaas LoginModule To Support Secure Database Authentication in WebLogic 7
by Tim Pijpops
Write your own database authenticator using the WebLogic, JMX, and J2EE security APIs.

Using SOAP Message Handlers with WebLogic 7
by Michael Gilbode
A standard, flexible method of implementing Web Services that allows additional processing when exposing a back-end component as a Web service

Tips 'n' Tricks: New Deployment Model in BEA WebLogic Server 7.0
by Kumar Allamraju
Prevent inconsistent deployment states across servers, especially clustered servers

Simplifying EJB Development with Xdoclet and BEA WebLogic
by Ryan LeCompte
An open-source tool that makes use of attribute-oriented programming concepts

BEA WebLogic DEVELOPER'S JOURNAL

have to say – and then acting on it. One of the things that Mary said earlier that shouldn't be understated is the fact that BEA dev2dev is built on all of our technologies. There is an article up on the site now about how the BEA dev2dev development team moved the site over to Version 7 of our platform. We are absolutely using our own technologies - or the infamous "eating our own dog food." We understand how these things work and what areas are a little more complicated than we might want them to be today, and are feeding that information back into the development team, as well as guiding developers through migration or upgrade processes..

WLDJ: What other enhancements – both online and in the community – do you have planned?

Haggard: One of the most exciting things we're doing is getting dev2dev much more tightly integrated with the products, and offering a really solid set of technical information through the products themselves. We're also going to make a big push out into the development community to find some external writers who can provide valuable content from a variety of viewpoints, skill levels, and perspectives.

Reid: One of the biggest changes that a developer who has been with us for a while has probably seen is that, in the past, BEA had at least five different places, if not more, for developers to hunt and search around for information. What our team is doing is pulling all of those resources together and making dev2dev into your "one-stop shop". If you have training needs, if you have support needs, if you are new at developing for WebLogic, or if you are just looking for some basic development information, this is always going to be your best starting place. We are going to catalog and prioritize a lot of these things for you to make things simple and easy to navigate, and to ensure that you spend the least amount of time searching for the answers you need.

WLDJ: Is there anything we didn't cover that you want to tell our readers?

Reid: I think the key thing that Mary said is that this initiative is truly by developers, for developers. While BEA certainly gets a lot out of it from the perspective of feedback and incorporating things into our products, this is absolutely about helping developers understand where we are going and how to use these technologies. It's about how to grow their own skill set and further their development careers. It's about enabling developer-to-developer communication and building out that community of experts. And it's about providing support and resources so that developers are successful in their jobs.

Haggard: In March I met a BEA developer who said something to me that I haven't forgotten since. He said, "I don't come to dev2dev for fun, I go to E!Online for fun. I come to you because I need you to help me solve my problems and if you do that for me, I will be back." I think that's something that we all think about as we build and run dev2dev, and if that's not what we are doing, we want to hear about it from the community because we want to make it better. People should know that we definitely want feedback, we definitely want to hear from people, we want to know from our community how to make this better.

• • •

To submit feedback to the dev2dev team, email dev2dev@bea.com or use the Feedback form located at http://dev2dev.bea.com/surveys/feedback.jsp.

• • •

Also on the dev2dev team:

Scott Fallon is vice president of developer relations at BEA Systems, Inc. Before BEA, he worked at Crossgain, a startup founded by Adam Bosworth (which was sold to BEA in the summer of 2001). Prior to that, Fallon held positions at Microsoft, including managing the company's breadth ISV program, developer marketing, and product management. Other technology roles included product management responsibilities at Microrim Hewlett-Packard.

Will Iverson is senior content manager of dev2dev Online. Will's development career dates back to the earliest stages of Java. He has worked in a variety of development, product management, and evangelism roles for Apple, Sun, Symantec, and others.

Ryan O'Hara is senior product manager of BEA dev2dev products. He previously ran finance for XML startup Crossgain, was Visual Studio business manager for Microsoft, and has held a variety of other finance and marketing roles.

Peter Ziatek is product development manager for dev2dev. Peter oversees all of the development effort that goes into dev2dev and is responsible for all platform upgrades and site functionality. He joined BEA two years ago after spending time at Silicon Graphics and AT&T.

"I come to your site because I need you to help me solve my problems and if you do that for me, I will be back."

– A developer

DEV TEAM OFFERS THE INSIDER SCOOP ON BEA'S DEVELOPER WEB SITE

IPLE MAGAZINES ONLINE

AND SAVE UP TO $400 AND RECEIVE UP TO 3 FREE CDs!

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

Pick a 3-Pack, a 6-Pack or a 9-Pack ✔

3-Pack

Pick any 3 of our magazines and save up to $275.00

Pay only $175 for a 1 year subscription plus a FREE CD

- 2 Year – $299.00
- Can/Mex – $245.00
- All Other Cnty. – $315.00

6-Pack

Pick any 6 of our magazines and save up to $350.00

Pay only $395 for a 1 year subscription plus 2 FREE CDs

- 2 Year – $669.00
- Can/Mex – $555.00
- All Other Cnty. – $710.00

9-Pack

Pick all 9 of our magazines and save up to $400.00

Pay only $400 for a 1 year subscription plus 3 FREE CDs

- 2 Year – $839.00
- Can/Mex – $695.00
- All Other Cnty. – $890.00

TO ORDER:

☐ 3-Pack	☐ 1YR	☐ 2YR	☐ US	☐ Can/Mex	☐ Intl.
☐ 6-Pack	☐ 1YR	☐ 2YR	☐ US	☐ Can/Mex	☐ Intl.
☐ 9-Pack	☐ 1YR	☐ 2YR	☐ US	☐ Can/Mex	☐ Intl.

Choose the Multi-Pack you want to order by checking next to it below. Check the number of years you want to order. Indicate your location by checking either US, Canada/Mexico or International. Then choose which magazines you want to include with your Multi-Pack order.

☐ **Java Developer's Journal**

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **Web Services Journal**

U.S.- Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. -One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **.NET Developer's Journal**

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **XML-Journal**

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **WebLogic Developer's Journal**

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

☐ **ColdFusion Developer's Journal**

U.S. - Two Years (24) Cover: $216	You Pay: $129 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189 /	Save: $75 + FREE $198 CD
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

☐ **Wireless Business & Technology**

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $192	You Pay: $139 /	Save: $53 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $79.99 /	Save: $16
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **PowerBuilder Developer's Journal**

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

SYS-CON MEDIA

News & Developments

BEA Enhances Certification Program

(San Jose, CA) – As part of its ongoing commitment to the IT community, BEA Systems, Inc., the world's leading application infrastructure software company, has introduced six new levels of role-based technical certifications, as well as significant enhancements to the BEA Certification Program.

Through the BEA Certification Program, candidates can achieve certification as a Certified Specialist, Certified Administrator, or Certified Architect on the BEA WebLogic Enterprise Platform, including BEA WebLogic Server, BEA WebLogic Portal, BEA WebLogic Integration, and Tuxedo. BEA Education Services offers training through traditional and innovative methods including on-site and public instructor–led classroom training, blended/e-Learning, self-study, and newly launched Web-based training offerings.

For more detailed certification program requirements or instructions on how to register for courses, please visit http://education.bea.com/education/certification_program.jsp.

PerformaSure 1.7 Extends Reach to WebLogic Enterprise Platform 7.0

(Toronto) – Quest Software Inc., a provider of application management solutions, has announced the release of Quest's PerformaSure 1.7, a transaction-centric performance diagnosis tool for analyzing distributed J2EE applications, with greatly expanded platform and application server support.

Companies using BEA WebLogic Enterprise Platform 7.0 can use PerformaSure to detect and diagnose performance problems in their J2EE applications. It provides J2EE performance teams with critical investigative insight into how J2EE applications, application servers, and JDBC calls interact to serve end-user requests in distributed transactional systems.

PerformaSure 1.7 also adds new data export features and performance enhancements. Any metrics data can be exported to a formatted text file, enabling users to compare key metrics over time using data analysis tools such as Excel. www.sitraka.com/performasure/media. www.quest.com

BEA WebLogic Server Earns COE Certification

(San Jose, CA) – BEA Systems has announced that BEA WebLogic Server has earned Common Operating Environment (COE) certification from the U.S. Defense Information Systems Agency (DISA).

COE is a technology framework used by the Department of Defense (DoD) to ensure that command-and-control systems throughout the Air Force, Army, Marine Corps, and Navy integrate in a seamless, reliable, and secure manner. It has become the de facto standard framework for strategic DoD computing systems.

BEA WebLogic Server underwent a comprehensive review by DISA and earned the elite certification after it was able to demonstrate that the software met or exceeded hundreds of technical requirements covering encryption, interoperability, performance, reliability, scalability, and security. www.bea.com

BT Selects BEA Systems as Its Technology Standard

(San Jose, CA) – British Telecommunications plc (BT) has chosen BEA WebLogic Platform 7.0 as its technology standard throughout the entire BT Group.

BT, one of the world's leading providers of telecommunications services and one of the largest private sector companies in Europe, will integrate the company's systems and applications on BEA WebLogic Platform, enabling BT to improve response times, reduce support costs, leverage shared resources, and continue to utilize valuable legacy systems.

BEA WebLogic Platform will also be used within BT for internal Web services projects designed to integrate legacy data with front office applications, and enable BT to realize a greater return on investment from existing implementations. www.btplc.com

Cardiff Software Develops LiquidOffice v2.1 for WebLogic

(San Diego, CA) – Cardiff Software Inc., a leading provider of business process-automation solutions, has joined the Star Partner Program of BEA Systems, Inc., as a One Star member. In addition, Cardiff has developed its LiquidOffice eForm automation solution to run on BEA WebLogic, providing large enterprises with a BEA-ready solution for creating, deploying, and automatically managing the routing, tracking, and approval processes for electronic forms.

Leveraging LiquidOffice's eForm technology, businesses operating in form-driven environments can reduce the costs associated with the manual processing and routing of forms. Designed for medium- and large-sized enterprises, LiquidOffice is optimized for enterprise connectivity and can easily scale to meet organizational requirements. www.cardiff.com

BEA to Run Germany's Telephone Directory System

(San Jose, CA) – Deutsche Telekom has selected BEA WebLogic Server as the application infrastructure for the German telephone directory system. The system, named Datenredaktion (DaRed), guarantees that the names, addresses, and phone numbers of more than 50 million German telephone subscribers, including mobile subscribers, are up-to-date and correct in all relevant German telecommunication directories.

Core DaRed applications will be built on BEA WebLogic Server with Linux S390 on an IBM mainframe. This system was implemented by T-Systems, the IT systems division of the Deutsche Telekom Group. www.t-systems.com.

BEA Systems

www.bea.com/liquiddata

Altaworks

www.altaworks.com