# WebSphere® DEVELOPER'S JOURNAL

*The World's Leading Independent WebSphere Developer Resource*

WebSphereDevelopersJournal.com

DECEMBER VOLUME 1 ISSUE 11








$8.99US $9.99CAN 12>
0 09281 03422 3

SYS-CON MEDIA

# Your Software Is Only as Secure as Your Developers

## *Software Company Allegedly Linked to Al-Qaida*

BY JACK MARTIN

As we go to press, Microsoft has issued its 65th security bulletin of the year disclosing a security flaw of critical severity in most versions of its popular Windows operating system. Microsoft urges users of Windows 2000, Millennium, 98, and NT 4.0 to download a software patch from the company's security Web site. Attackers could exploit a vulnerability in the software that underlies many database functions and take over the user's computer. Microsoft's newest version, Windows XP, does not have the problem.

In the U.S. it was recently discovered that a substantial percentage of airport security workers – who are supposed to assure that no one boarding a flight is using false identification – are using false identification themselves. Quite a few are also criminals; hundreds have been arrested in the past couple of weeks. These people were hired to protect us from people just like them.

The largest security hole in the country's information technology infrastructure is not some new type of virus or an outside hacker trying to get into corporate or government systems; it's developers working on software projects all over the world. In the current technology downturn it has become very popular to turn to offshore outsourcing, sending IT projects all over the world using the Internet as the conduit. Every Fortune 500 company and U.S. government office uses software created by people of unknown backgrounds and intentions. It is a standard practice on large technology projects for one company to act as the prime contractor and have multiple specialty companies act as subcontractors. For example, Microsoft regularly has source code written in India.

There are absolutely no development security standards in place anywhere in the industry. What is checked is that the price is as low as possible and that the code works. Nobody ever checks with the FBI, CIA, or anyone else to assure that the developers working on the projects do not have a second agenda. For all you know, Osama bin Laden himself or people who think like him are writing part of the source code for your current project, gaining intimate knowledge of every vulnerability of the system being built and any other systems it will interface with. Remember, you stand on line at airports all over the U.S. for hours to go through make-believe security checks staffed by people with false identification and criminal backgrounds.

I know this may sound unbelievable, but that is exactly what the U.S. Customs Service thinks it found last night in a raid on Ptech Inc. They believe Ptech is linked to one of bin Laden's alleged money men, a Saudi multimillionaire named Qassin al-Kadi. If true, it is not a good thing. Ptech has worked with Aetna, Allegheny Energy Supply, Booz Allen Hamilton, the FAA, the FBI, IBM Global Services, MetLife, Motorola, NATO, Pricewaterhouse Coopers, Southern California Edison, Sprint, the Air Force, the Department of Energy, the Department of Education, the Department of Veterans Affairs, the House of Representatives, the Forest Service, the Postal Service, and Weyerhaeuser.

After this revelation it should be evident to any rational person that it is not enough for any software house – no matter how big – to say they have some type of unaudited security system in place and that we shouldn't worry. What the industry needs is to work with government to put in place an independent third party to act as a central clearing house that at least seriously attempts to assure that terrorists or terrorist sympathizers are not actively mining software projects with endless trouble.

Anyone who says that the thought of methodical software terrorism and cyber warfare is far-fetched, that the industry does not need to be hyper-vigilant, and that it could never happen only has to look at the past and see that it was not too long ago that U.S. television featured shows that focused on the first World Trade Center bombing. These shows always ended with the assurance that the people who did it were safely in jail and there was no way they could ever knock down one of the buildings. Well, guess what? The next time they tried to knock them down – they succeeded. It took 10 years for them to figure it out, but they did.

ABOUT THE AUTHOR... Jack Martin, editor-in-chief of ***WebSphere Developer's Journal***, is cofounder and CEO of Simplex Knowledge Company, an Internet software boutique specializing in WebSphere development. Simplex developed the first remote video transmission system designed specifically for childcare centers, which received worldwide media attention, and the world's first diagnostic-quality ultrasound broadcast system. E-MAIL... jack@sys-con.com

WebSphere® DEVELOPER'S JOURNAL

ADVISORY BOARD
Richard Arone, James Martin, Greg Williams

EDITOR-IN-CHIEF JACK MARTIN
EDITORIAL DIRECTOR JEREMY GEELAN
EXECUTIVE EDITOR GAIL SCHULTZ
CONTRIBUTING EDITOR PATTI MARTIN
SPECIAL PROJECTS EDITOR RICHARD ARONE
MANAGING EDITOR JEAN CASSIDY
EDITOR NANCY VALENTINE
ASSOCIATE EDITORS JAMIE MATUSOW
JENNIFER STILLEY

WRITERS IN THIS ISSUE
Melissa Borza, Luke Chen, Jim Clune, Rini Gahir, Marina Gil-Santamaria, Ken Greenlee, Jay Johnson, Joe Krozak, Jack Martin, Patti Martin, Ruth Willenborg, Andrew S. Wong

SUBSCRIPTIONS
For subscriptions and requests for bulk orders, please send your letters to Subscription Department.
SUBSCRIBE@SYS-CON.COM
Cover Price: $8.99/Issue Domestic: $149/YR (12 Issues)
Canada/Mexico: $169/YR Overseas: $179/YR
(U.S. Banks or Money Orders)
Back issues: $15 U.S. $20 All others

PRESIDENT AND CEO FUAT A. KIRCAALI
COO/CFO MARK HARABEDIAN
SENIOR VP, SALES & MARKETING CARMEN GONZALEZ
VP, INFORMATION SYSTEMS ROBERT DIAMOND
VP, SALES & MARKETING MILES SILVERMAN
VP, EVENTS GRISHA DAVIDA
PRODUCTION CONSULTANT JIM MORGAN
FINANCIAL ANALYST JOAN LAROSE
ACCOUNTS RECEIVABLE KERRI VON ACHEN
ACCOUNTS PAYABLE BETTY WHITE
ADVERTISING ACCOUNT MANAGERS MEGAN RING-MUSSA, ROBYN FORMA
ASSOCIATE SALES MANAGERS CARRIE GEBERT, ALISA CATALANO, KRISTIN KUHNLE
CONFERENCE MANAGER MICHAEL LYNCH
REGIONAL SALES MANAGERS MICHAEL PESICK, RICHARD ANDERSON
ART DIRECTOR ALEX BOTERO
ASSOCIATE ART DIRECTORS LOUIS F. CUFFARI, RICHARD SILVERBERG, TAMI BEATTY
WEB DESIGNERS STEPHEN KILMURRAY, CHRISTOPHER CROCE
CONTENT EDITOR LIN GOETZ
CUSTOMER SERVICE REPRESENTATIVE MARGIE DOWNS
CUSTOMER RELATIONS 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
WebSphere® Developer's Journal (ISSN# 1535-6914) is published monthly (12 times a year).
Postmaster send address changes to:
WebSphere Developer's Journal, SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645

# A CONVERSATION WITH IBM'S JOE ANTHONY

**The program director for WebSphere Marketing discusses WebSphere Application Server 5.0**

*Joe Anthony talks with Jack Martin, editor-in-chief of* ***WebSphere Developer's Journal****, about e-business on demand, Grid computing, and IBM's competition.*

WSDJ: WHAT CAN YOU TELL ME ABOUT WEBSPHERE APPLICATION SERVER V5?

**JA:** Let me put the announcement in context. Application servers that originally were used as Web front ends have evolved into handling more of the core application infrastructure. Nowadays, customers need a much broader integration platform that includes things like Web services choreography and portal and commerce infrastructure. So the application server is the underpinning of a customer's overall core infrastructure.

WSDJ: WHAT IS WEB SERVICES CHOREOGRAPHY? AND HOW IS YOUR IMPLEMENTATION HANDLING IT?

**JA:** Many of our customers are taking a lot of different core back-end assets and new applications and making them into individual pieces as Web services. How do you tie all that together with a workflow, how do you choreograph them, how are all of these individual Web services going to work together for an end-to-end flow? Workflow is not a new concept – it's been around for a while, but the idea of taking all the individual elements and doing that with Web services is relatively new. It allows customers to leverage a lot of their existing assets.

WSDJ: COULD YOU GIVE ME AN EXAMPLE OF HOW SOMEONE WOULD USE THAT IN THE REAL WORLD?

**JA:** You can take portlets, as an example, and expose them as a Web service. You could take some CICS (Customer Information Control System) artifacts that you might be working with and pull out information and expose that as a Web service. You can start taking individual applications and flow the entire end-to-end piece together. For example, you might have pulled some travel information out of CICS, combined it with some other information, and flowed the information together to start leveraging the application server infrastructure as part of an overall integration platform.

WSDJ: THAT'S A GREAT STEP FORWARD.

**JA:** It's very interesting – and it's very powerful for our customers. If they don't want to start from scratch with their different IT projects, Web services choreography makes it possible for them to more tightly integrate these different aspects. They don't have to devote time to human intervention at many different points along the way; they can let the application handle it end-to-end.

WSDJ: IT SOUNDS AS IF YOU'VE FIGURED OUT A WAY FOR THEM TO LEVERAGE AN EXTENSIVE PORTION OF THEIR EXISTING INVESTMENT.

**JA:** Yes, and we're enabling them to expose these assets as Web services and providing them with the ability to choreograph them. In addition, we're very active participants in the new Web services standards organizations such as WS-I, W3C, and OASIS. We're working with the Web services standards groups to make continued enhancements in those areas.

WSDJ: HOW DO YOU SEE THAT PLAYING AGAINST YOUR FRIENDS IN REDMOND?

**JA:** We will continue to cooperate with Microsoft when it comes to defining Web services standards, but we feel that we will also be very competitive when it comes to the actual implementations. We feel that with our Web services implementations running on more platforms than just Windows, we have a lot more to offer our customers – we give them the freedom of a much more flexible infrastructure, to support platforms ranging from Linux to AIX to Windows. They can choose what kinds of hardware and operating systems they want to run. We'll make sure that the applications they choose to write can be deployed on them.

WSDJ: I THINK A LOT OF PEOPLE AT MICROSOFT, WHEN THEY THINK ABOUT WEB SERVICES, THINK IT'S SPELLED DOT-N-E-T. HOW DO YOU ADDRESS THAT?

**JA:** We think that .NET is fine for running

Web services that require simple connectivity on Windows devices. But it's certainly not the only choice. We want to make sure that our customers can choose to deploy on Windows if they want to. But we want to make sure that they have more flexibility than that, and that they enjoy the security, reliability, and scalability that WebSphere provides. We would advocate that customers run Web services on a WebSphere J2EE environment – not a pure .NET environment.

WSDJ: WHEN I THINK ABOUT IBM, I THINK ABOUT AUTONOMIC COMPUTING, IN WHICH IBM HAS MADE MAJOR STRIDES IN THE PAST 12–24 MONTHS. HOW IS AUTONOMICS PLAYING IN THIS SPACE RIGHT NOW?

**JA:** It's very powerful. Let's step back for a little bit: if you think about the autonomic nervous system of the body, you want the computing environment to have those same types of characteristics – for an IT infrastructure to run and heal itself. A lot of IT spending is devoted to areas other than just a pure purchase of hardware and software; a lot of it goes into the IT administration's need to run the infrastructure, continually update it, configure it, optimize it, etc. So if you can infuse that intelligence into the software and hardware, it's going to be a lot better for the overall IT administration. It can enable a business to be operated based on business rules and business objectives, rather than requiring an IT staff to focus on all the day-to-day hardware and software tuning. We feel it's key to have software that is self-configuring, self-healing, self-optimizing, and self-protecting.

In the area of self-configuration, it's very important for the software to be able to monitor what is going on in its environment and then give tuning recommendations based on what's occurring. Now eventually you would want the tuning recommendations to be automatically implemented, but we don't feel most IT staffs are ready for us to be making a whole lot of changes under the covers before they're comfortable with that level of technology. We will be making recommendations for distributed platforms that can be implemented based on a customer's experiences. We can also make sure that access methods and overall responses are optimized.

In the area of self-healing software, a key attribute is the ability of the system to monitor itself. If it detects something wrong in individual application performance, it can try to tweak individual things, but at some point you may decide that you just need the application to be taken down and restarted. The same application may be present in the rest of the cluster, so the individual user is not impacted because the application will be rolled over to another one.

When you look at self-optimization, you want to prioritize work based on the type of user – so you can give gold service to banking customers who have the largest accounts versus silver service for those who don't – as well as prioritize usage based on the type of application. If you have a Web application where people are just browsing for information, you don't want to give that kind of user a higher priority than someone who is using a shopping cart on a commerce location. So the ability to prioritize Web traffic is very important.

WSDJ: SO YOU'RE DOING SOME TYPE OF LOAD BALANCING THERE?

**JA:** It's load balancing, as well as utilization monitoring, which allows you to make sure individual applications don't bottleneck a given resource.

In the area of self-protecting, we have the ability to detect denial-of-service attacks against the Web application infrastructure, so you can make sure that requests are well-formed, etc., at the front-end server before you pass the request to the back-end server. You can make sure that the attacks are relegated to the periphery and not tying up your back-end application servers and applications as you're trying to execute, thus minimizing the disruption to your environment.

WSDJ: IT SOUNDS AS IF COMPUTING IS ALMOST READY TO BECOME LIKE A UTILITY.

**JA:** Yes, that's a very important part of what we're doing with e-business on demand. We feel that the overall environment is moving to the point where on-demand businesses really need to be highly adaptive enterprises that can instantly sense and respond to changing conditions. These businesses are flexible and integrated enough to dynamically change based on the need. So as conditions change in the infrastructure, the businesses can respond. They can also respond to requests, so as customer demand is peaking during certain seasons, etc., you can dynamically change your internal environment as well as draw upon extra external resources to help meet that demand.

So it's the ability to think about your IT infrastructure as a resource that you can get as the next-generation utility, to go ahead and tap resources – whether they be internal or external – on demand and repurpose them as needed; it's the ability to move servers from one environment supporting one application to supporting another. It's thinking about that computing infrastructure without having to worry about exactly what platform, which application, etc., rather thinking about the overall context and business process that's key to your business. It's being sure you have the assets you need to fulfill your customers' needs.

WSDJ: HAS ANYONE TRIED THAT YET?

**JA:** Take a look at the partnership we have with Akamai, which has already started to leverage some of their individual static and dynamic caching services. They're now incorporating WebSphere into their network to execute business logic on WebSphere servers. The customer can choose to have a certain level of IT infrastructure and then as they need additional capacity, they have the ability to draw upon WebSphere in the Akamai network. In addition IBM has announced application hosting services and several partners are working with customers in an on-demand infrastructure.

WSDJ: HOW DO YOU SEE THIS COMPUTING-AS-A-UTILITY ROLLING OUT OVER THE COURSE OF THE NEXT 24–48 MONTHS?

**JA:** Software used by customers inside the firewall will offer additional features that will make it easier to manage an internal IT shop as applications that are available on-demand internally. At the same time, because of the work we are doing with different systems integrators, etc., customers will have the option of working with integrators who are building that same kind of business-on-demand model.

WSDJ: IT SOUNDS LIKE WE'RE ABOUT READY TO ENTER INTO A WHOLE NEW WORLD AGAIN.

**JA:** We are – and it's a very interesting world. People are going to be able to focus on what they do best and not necessarily have to build up their own internal infrastructure and worry about having a five-times-over capacity within their environment. When you look at it, most of our

larger customers who are running IBM zSeries mainframes do an extremely good job of managing their infrastructure. Utilization of a machine like the zSeries is very high. As you start moving down in the level of servers, etc., by the time you get down to regular Intel-based servers, you're starting to look at utilization rates that might be on the order of 10 percent. If you can get to where you don't need to have that level of excess capacity in your infrastructure, it makes a very powerful statement about what you can do to run your business more efficiently.

WSDJ: I UNDERSTAND FROM A FEW PEOPLE INSIDE OF IBM THAT SETI AT HOME IS ABOUT TO GO COMMERCIAL WITH APPLICATION SERVER V5. IS THAT TRUE?

**JA:** That would be an interesting concept, but the existing SETI is more of a research project and it does not have the kind of commercial infrastructure that most of our customers would want to apply in their own IT infrastructure.

WSDJ: SO YOUR DEPLOYMENT OF GRID IS DIFFERENT FROM SETI'S?

**JA:** If you take a look at what you need to deploy a Grid, you have to have more security built in, you have to have authentication, and authorization of the assets you want to make available. You also have to worry about the accounting metrics you're going to put in place, and how you're going to manage the whole environment. So there's a lot more that goes into it than something as simplistic as SETI. SETI did set a good precedent; it helped us learn an awful lot about what you'd want to put into a Grid environment.

WSDJ: IT'S AN EXTRAORDINARY EXPERIMENT.

**JA:** But if you start thinking about what an IT director wants to do with his own internal environment – that 90 percent of free Intel server resources that is available within his enterprise – he doesn't want that used up by something like SETI. He wants a much more controlled environment that he can manage in the overall infrastructure and make sure it's being put to very productive use and not just going haphazardly all over his network. He wants a better control than that, and the Grid that's being defined is doing a good job of that.

WSDJ: WHAT TYPE OF THINGS COULD YOU FORESEE SOMEONE USING GRID FOR INSIDE THE ENTERPRISE? TAKE A TYPICAL MIDMARKET COMPANY THAT HAS ABOUT 500 INTEL BOXES THAT SPEND MOST OF THEIR TIME IDLE ALL DAY. WHAT WOULD BE A COUPLE OF READY-FOR-PRIMETIME THINGS THAT THEY COULD DO WITH IT TODAY?

**JA:** It depends on the kind of company, but as an example, if you were doing design simulations, they are often done on larger servers. With the amount of spare cycles that you have in most infrastructures, you could run the design simulation on lower-powered servers on a distributed basis and better leverage those resources, particularly during off hours, but even during normal hours. You can set the machines up to so that you can leverage the cycles that are not being used, but if someone then comes back and starts using the machines, you can back off and put it in a background task.

I think another thing that's helped Grid a lot recently was that about a year ago the protocols around the Grid environment adopted a Web services infrastructure. The value of the interoperability between applications is so powerful there – adopting that kind of a paradigm for Grid is just going to make the overall adoption and industry push that much stronger. So we're very much behind Grid and we're very active in the overall GGF (Global Grid Forum) with it.

WSDJ: MY PERCEPTION OF GRID IS THAT THE BASIC BUSINESS PROPOSITION IS THAT YOU ALREADY OWN THIS HORSEPOWER AND THESE ASSETS, SO WHY NOT USE THEM.

**JA:** When you think about e-business on demand, you can do that with the resources you own. It also affords customers the ability to tap into the e-business-on-demand assets that they may be getting from other sources as well. So all these things – when you take a look at the Grid, the autonomics, the e-business on demand – it's a very powerful framework. It all fits together very nicely.

WSDJ: LET'S CHANGE GEARS FOR A MOMENT. OBVIOUSLY, YOU GUYS HAVE SPENT THE PAST 12–24 MONTHS THRASHING BEA, AND YOU'RE AHEAD OF THEM OFFICIALLY AT THIS POINT – HOW DO YOU SEE YOUR NEW OFFERING STACKING UP AGAINST THEIRS?

**JA:** We're very excited about what we see. The proof is in the numbers, and our nonstop growth demonstrates the fact that customers think IBM offers the best and broadest solution in the industry. WebSphere has had 14 consecutive quarters of double-digit growth and BEA has had five consecutive quarters of declining license revenues, so you can see from an overall marketplace adoption perspective where the marketplace is going. I think that with the breadth of the platform coverage that we bring to the table around Web services, portal, commerce, etc., we offer a very broad platform for customers who are looking to invest in their infrastructure. Really, I think that there are only two companies that have this kind of breadth, and they're IBM and Microsoft. And if you take a look at IBM, we feel we're allowing customers to make sure they can operate in a much more open environment. It does include Windows, but it's not exclusive to Windows. And when you take a look at how to reuse a company's assets and get to a more open development environment – our open-source contributions to Eclipse and the tight integration we have WebSphere Studio v5 as well as with DB2, Tivoli, and Lotus – together those things make a very powerful solution for the IT investment and the value that the customers can see.

WSDJ: ANY FINAL THOUGHTS ON APP SERVER V5 AND STUDIO V5?

**JA:** When you take a look at what IBM has accomplished with WebSphere v5 and WebSphere Studio v5, it has evolved into a very powerful platform for an on-demand business. Companies can look at taking all of their core assets and deploying them as Web services. We provide the ability to choreograph them and integrate them into their core business and have set the foundation for how autonomics and Grid play in this overall on-demand business for customers. So we're enabling customers to move into this new on-demand business world and we feel that leveraging the WebSphere platform is going to give them a leg up on their competition.

*Legacy migration at the speed of light*

# Fast Path to Data

BY ANDREW S. WONG

Enterprises have invested much time and money in legacy systems. According to industry analysts, over 70% of the world's data is contained in legacy systems. IBM states that there are currently more transactions processed by CICS (IBM's Customer Information Control System) systems today than by the entire Internet. CICS handles more than 30 billion transactions per day, and is used by 492 of the Fortune 500.

ABOUT THE AUTHOR

Andrew Wong is vice president of engineering at ClientSoft (www.clientsoft.com), a leader in delivering fast and secure integrated applications to legacy systems. He designs and develops software solutions specific to the e-business and electronic commerce industry. Before joining the company in 1990, Andy worked for the MITRE Corporation in Massachusetts, where he was a member of the technical team specializing in research and analysis through computer simulations. Andy holds a BS in electrical engineering from the Cooper Union School of Engineering and an MS in the same major from Boston University.

E-MAIL
clientsoft@lewispr.com

Nine out of 10 ATM transactions are done using COBOL. IT departments in large corporations manage over 30,000 CICS code modules written in COBOL, Assembler, or PL/I. The oldest modules are 25–30 years old, but still process mission-critical data every day and almost certainly will continue to run for years to come. You get the idea – legacy systems still have an important role to play in business today.

This enormous amount of code, running on mainframe and AS/400 systems, is collectively known as "legacy applications." Between 70–80% of IT budgets is spent yearly to maintain and evolve this code.

Many corporate IT departments have turned to Windows and Unix-based machines to complement their existing mainframes for developing and deploying Web and composite applications. The mission-critical legacy applications residing on the mainframe must be leveraged to provide both data and business logic to power these new Web and composite applications. It is too costly to simply recode legacy applications in languages designed for these new platforms.

Legacy application integration is the solution. Existing legacy applications must be made available to these new Web and composite applications. Corporate IT departments must be able to isolate specific business transactions from within one or more applications and expose them on the Web or create composite applications. Web services is the technology powering this new form of application integration.

The challenge lies in how to provide Web services from existing legacy applications – without relying on terminal emulation. The traditional approach to legacy connection has been through emulation. In the past 12–18 months, emulation-based products have emerged that can provide Web services from underlying host transactions. While this is a major step forward, the need still exists for high-performance integration with the host. ClientSoft has pioneered a new breed of legacy integration tools to provide "direct integration" with mainframe applications. This direct integration approach allows the creation of Web services that interact with existing mainframe applications, without relying on terminal emulation.

## The "Screen Scraping" Image

In the early 1990s, emulation-based technology was in its infancy and was characterized as "screen scraping." Screen scraping is the process of taking the data from the screens of a legacy program that was formatted for a "green-screen" terminal, and reformatting the data for use in a new user interface. The input from the new interface must also be reformatted to allow the underlying legacy program to understand the request. Screen scraping became a derogatory term. However, at the time using the data stream in a screen scraping project was the only surefire way to access legacy transactions.

The notion of a middle-tier application server changed many of the perceptions of screen scraping. Vendors emerged with automated development environments that leveraged the capabilities of a middle-tier legacy application server to provide "intelligence" around the navigation of the legacy screens. These tools were based on terminal emulation but provided many additional features designed to improve both the development and deployment of applications based on legacy integration. The term screen scraping was replaced with emulation-based or screen-based as labels for the new evolution of legacy tools. Let's explore how these emulation-based tools work.

## EMULATION-BASED LEGACY INTEGRATION

- During the development process the integration tool is used to navigate the host-based application and automatically capture each screen map file. This map file contains all of the input and output fields that make up the actual screen. The developer is able to encapsulate a complete business transaction across multiple host screens and expose it in a presentation or programmatic interface. A few vendors have delivered Web services as a programmatic interface option.
- The runtime component of most emulation-based tools is a middle-tier legacy application server, typically installed on a Windows or Unix machine. This software-based server acts as the traffic cop during deployment. It interacts with the legacy system using the appropriate terminal emulation protocol, and with the front-end application using TCP/IP.

- The presentation or programmatic interface can be in many forms – a Java or HTML application, JavaBean, COM object, or as a Web service. At runtime, this front-end interface will communicate with the middle-tier legacy application server via TCP/IP.

### CHALLENGES OF EMULATION-BASED LEGACY INTEGRATION

- ***Performance of the new interface:*** The use of terminal emulation is a limiting factor of application performance and scalability. The data stream can only handle 1920 characters of data at any one time, based upon the 80x24 size of a legacy host screen. This fundamentally limits how much information can be passed to and from the legacy application via terminal emulation.
- ***Dual maintenance burden:*** The geometry of the host screen requires programmers to engage in manual row-and-column based development projects that fail if the position of the fields on the underlying host screens is altered. This forces application developers to modify the emulation-based interface any time a change is made to the underlying legacy application.
- ***Middle-tier application server problems:*** The middle-tier application server transformed emulation-based legacy integration into an acceptable technology for enterprise integration. Unfortunately, it has generated a few new problems for developers. The addition of a middle tier degrades the overall performance of a terminal emulation session. It also adds another break point for the application, as Microsoft and Unix servers have not achieved the reliability that the mainframe represents.

Emulation-based legacy integration tools have proven valuable in application integration projects. Enterprises have used these tools to unlock legacy applications for integration with WebSphere Application Server into a new J2EE application and other packaged applications, such as a Siebel CRM application or a new portal application.

## The Demand for Direct Integration

These proven emulation-based solutions have created a demand for direct integration with the mainframe. Direct integration with host applications provides a higher-performing integration solution and increased stability. Enterprises want to maximize performance of their legacy integration solutions and remove the dual maintenance burden. Many companies are turning to development and deployment environments for direct mainframe integration. Let's examine how the direct access approach works in contrast to emulation-based integration:

### DIRECT ACCESS LEGACY INTEGRATION: HOW IT WORKS

- A Windows-based development environment interacts directly with mainframe CICS and IMS (Information Management System) applications through one of the following APIs: COMMAREA, FEPI, or 3270 Bridge – rather than through emulation. The development environment is wizard-driven, with a visual drag-and-drop interface. The integration tool automatically generates J2EE components, as well as Web services, from existing mainframe transactions.
- The runtime component resides on the mainframe inside a CICS region. This allows the integration tool to interact with transactions using direct APIs rather than emulation. There is no middle-tier requirement – just the mainframe-based runtime and the component or Web service. All communication from the component or Web service is through TCP/IP – no SNA or LU 6.2 required.
- The generated components and Web services can be used as standalones or in an application server with WebSphere or another J2EE application server.

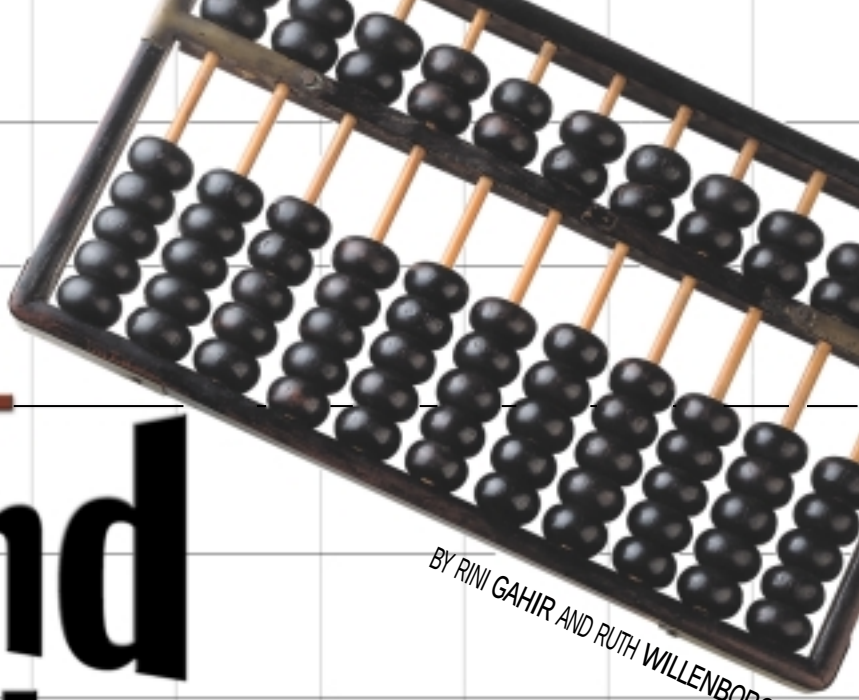
### BENEFITS TO CONSIDER

- Significant performance improvements over emulation-based integration solutions.
- Reduced risk of failure: having no middle-tier Windows or Java-based server requirement removes a potential break point.
- Full compatibility with WebSphere Application Server.
- Enables WebSphere to interact directly with transactions on the mainframe without the need for emulation or extra coding (nonintrusive access) and without modifying the business logic or altering the existing code in any way. However, developers can add new logic in Java or HTML to complement what works on the old system. New functionality can be tied in without changing the back-end code.
- Nonintrusive to the original legacy code.
- Automatic generation of WebSphere-compliant EJBs.
- Automatic generation of J2EE components.
- Automatic generation of Web services.
- Ease of use: direct integration tools allow both legacy and object-oriented programmers to work with back-end legacy transactions and expose them as components or Web services with little or no knowledge of COBOL, Java, or Web services.
- The development environment isolates a specific unit of work (business transaction) across one or more applications for faster execution at runtime.
- Real-time information access via the Internet, extranet, or intranet.
- The ability to preserve and leverage investments in legacy systems.

## Conclusion

At the most basic level, legacy integration solutions offer the ability to expose valuable business transactions to new applications via components or Web services. Emulation-based tools have been the preferred method to accomplish application integration with legacy systems to date. The advent of direct-access legacy integration solutions presents a dramatic paradigm shift. IT enterprises now have a high-performing, architecturally pure solution for legacy integration. Emulation-based and direct access solutions both play a role in providing a complete legacy integration solution.

# WebSphere Performance Diagnostics – Going Beyond the Metrics



BY RINI GAHIR AND RUTH WILLENBORG

## Focusing on end-user transactions

ABOUT THE AUTHOR

Rini Gahir is the product manager for PerformaSure at Sitraka (now part of Quest Software). With more than eight years experience in ERP, e-commerce, and software application marketing, Rini's responsibilities include identifying market needs, defining new technologies to meet customer requirements, and steering product development strategies.

E-MAIL

rini.gahir@sitraka.com

J2EE has arrived, and is gaining strength and popularity every day. J2EE does an excellent job of solving enterprise computing problems. It supports legacy applications and interfaces, multiple operating systems, distributed and clustered environments and high-volume mission-critical applications with support for security and managed operations.

Regardless of the type of Web application model being adopted, there are some issues that need careful consideration during the design and implementation of an e-business site. These include the reliability, security, capacity, scalability, and cost of the system and network. E-business activities and Web services are essentially real-time processes in which performance and availability problems have a high cost. Frustrated users can translate into lost customers and lost revenue. Set performance, availability, and security goals very early in a project because they are closely related to the design of the application.

Successful performance management helps you detect and correct performance problems. In defining your performance assurance solution, consider the following tenets:

- If you don't measure performance, you won't detect that you have problems.
- If you don't know where to measure performance, you can't diagnose problems.
- If you don't know how to find the root cause of a problem, you can't fix problems.

Success can hinge on the ability to detect, diagnose, and resolve application problems quickly. This can be a particularly challenging proposition if you're taking advantage of component-based J2EE applications. Due to their increased complexity, these applications are much more difficult to diagnose than earlier application environments. The dynamic nature of these applications, and the underlying infrastructure, creates a need for advanced management software that automatically discovers transactions and transaction components while adapting to changes without intervention.

### Quality of Service – the J2EE Challenge

Speed, around-the-clock availability, and security are the most common indicators of the quality of service of your Web site. E-business sites based on J2EE technology are complex, with multiple interconnected layers of software and hardware components. The nature of e-business workload is also complex, due to its transactional nature, security requirements, and the unpredictable characteristics of service requests from end users. Therefore, planning for capacity and the performance aspects of your application is a vital step in achieving performance assurance.

In a multitiered distributed application architecture, application logic is divided into components according to function. Which tier a J2EE application component ultimately resides on is contingent on the component's function as it relates to the J2EE environment. The J2EE environment has three main tiers: a client tier, a middle tier, and an enterprise information systems (EIS) or back-end tier.

The bulk of the J2EE engine resides on the environ-

ment's middle tier. The middle tier provides client services through Web containers, which house such J2EE components as servlets and JavaServer Pages (JSPs), and business-logic services through Enterprise JavaBean (EJB) containers, which house EJB components or enterprise beans.

Adopters of the J2EE architecture want to know quantitative, credible answers to many questions. These include:

- How well do my application servers manage scale-up in client connectivity, server processor count, load balancing across multiple nodes, size and duration of transient state, transaction complexity, etc.?
- How should a system manager determine the size of systems for the J2EE platform and EJB architecture?
- Given method calls between beans, how does performance vary as a function of location? E.g., within the same EJB server/container, across different containers inside the same EJB server, across different EJB servers, etc.
- What is the performance difference between session and entity beans (assuming that design patterns permit a choice of either)?

In addition to the J2EE stack, you also want to understand its relation to the other distributed hardware and software components such as firewalls, Web servers, application servers, and databases.

## Business Solution Requirements

Running a successful e-business requires an integrated performance management solution that spans applications, systems, services, and the network. Your diagnostic solution must provide insight into the execution paths of logical transactions. This requires the ability to map a transaction by its constituent elements – such as networks, systems, Web servers, application servers, and application components. Also required is the ability to collect timing on each element and to recognize and call out unusual events.

- ***End-user response time:*** This is the actual time it takes for a customer to complete an application transaction. Measuring online customer response time and application availability provides the first step toward controlling performance risks. By documenting compliance with service-level agreements and quality of service, businesses can increase customer retention, build customer loyalty, and increase upgrade and cross-selling opportunities.

- ***Multiple layers of e-business services:*** The holistic end-to-end view of your e-business system is provided by aggregating and correlating the performance and state of the network, systems, and application elements involved in delivering a business service. The interaction between these layers produces the customer's online performance experience. Performance data must be collected and analyzed to establish baseline performance and help IT management size the environment. Real-time data collection and analysis allows response to and correction of potentially devastating problems. This improved control translates into higher performance services that can mean higher product revenues.

- ***Proactive management:*** Identifying and resolving performance problems before the customer experiences them is the key to performance management. Proactive performance management allows administrators to identify performance slowdowns and correct problems before customer service disruptions occur.

- ***Flexible reporting and collaboration:*** The testing and deployment of J2EE-based systems requires the involvement of cross-functional teams such as developers, network engineers, QA engineers, system administrators, and DBAs. The need to share objective data and collaborate with different teams helps to frame system performance in the context of the distributed environment. Also, you need to share analyzed data in a report format with management on a regular basis. For example, a technical administrator needs detailed technical data for troubleshooting, an IT manager needs trend information for capacity planning, and a line-of-business manager needs service performance summary reports for sales analysis. Flexible reporting makes it easier to share knowledge throughout the organization so a variety of personnel can make business decisions and uncover additional selling opportunities based on service performance.

## The Need for a New Type of Tool

Since these Web applications are customer facing, managing their performance is critical. However, available tools for pinpointing and diagnosing performance problems in the J2EE environment have been limited.

In the development environment, Java profilers have existed for many years. These tools, such as Quest (Sitraka) JProbe, are excellent for improving code paths or finding memory leaks in the development environment.

Once an application is in a test or production environment where significant load is applied, profilers are no longer effective. Therefore, many load-related performance issues, such as application scalability, poor tuning, or configuration problems are difficult to diagnose and resolve.

Traditional solutions for analyzing Web application performance under load have two main challenges:

- ***Real-Time Monitors Only:*** Traditional IT management solutions are designed and built to manage and monitor real-time server availability. These systems provide relevant metrics on the state of the particular server being monitored. What they lack is a view into application response-time problems from an end-user perspective.

- ***Silo tools:*** Typical IT organizations have a collection of tools that are very "device-centric," such as network monitors, database tools, and even application server monitors. While these silo tools provide extensive details about each server, they do not look at the entire end-to-end service request or business process. Without this holistic view of an end-to-end transaction, it becomes nearly impossible to solve performance problems before the end-user experience is impacted.

Regardless of the granularity that these tools may offer, management of these increasingly complex *n*-tier

ABOUT THE AUTHOR

Ruth Willenborg is a senior programmer manager at IBM. She has 15 years of experience in software development and is currently a member of the WebSphere Development Organization. Ruth is responsible for the WebSphere Application Server Performance team. She is coauthor of *Performance Analysis for Java Web Sites* (Addison-Wesley, 2002).

E-MAIL
rewillen@us.ibm.com

infrastructures is typically applied in a disparate, silo approach, with multiple departments owning the operations responsibilities for various layers. This often results in reactive finger-pointing and departmental blame when performance issues arise. What is required is a solution to measure end-to-end transaction response times with visibility into all layers of the J2EE stack, including the application servers. But metrics and data are not enough. The need to correlate this information in the context of the application and its infrastructure is the key to isolating and diagnosing the root cause of the problem in the context of the end-user experience. Performance tools that provide this type of information are emerging, such as Quest PerformaSure and Wily Introscope.

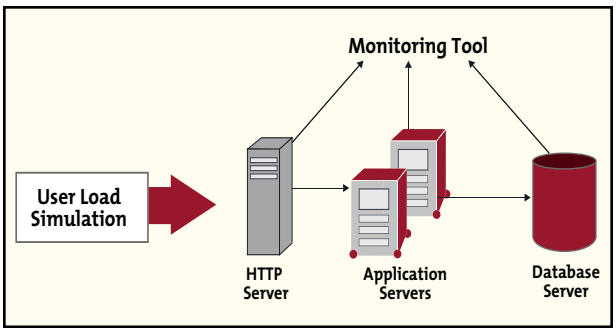



FIG. 1: LOAD TEST CONFIGURATION

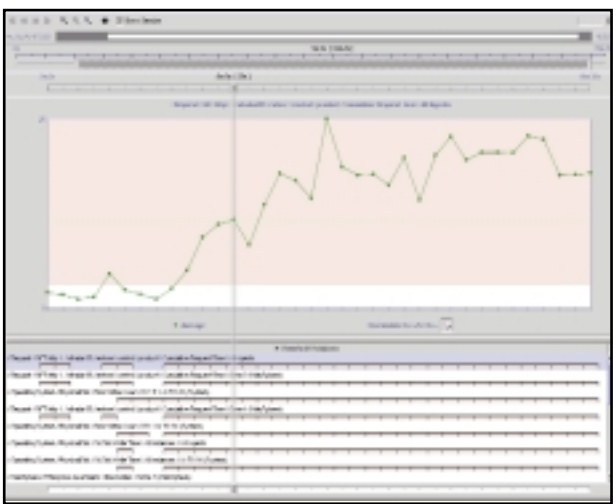


FIG. 2: EXAMPLE OF POOR RESPONSE TIME

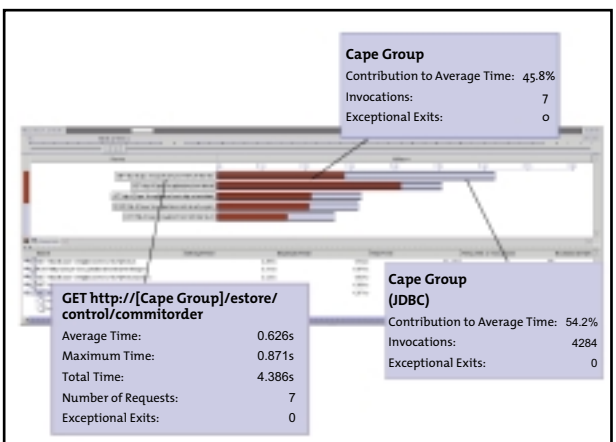



FIG. 3: EXAMPLE OF A TRANSACTION BREAKDOWN

In this article we will illustrate the diagnosis of a typical WebSphere-based J2EE application from an end-user transaction-centric perspective using PerformaSure. With the ability to capture transaction metrics and correlate them with metrics from system-level, network, database, and WebSphere application servers, we will be able to efficiently get to and repair the root causes of our performance problems.

## The Power of End-to-End Transaction Analysis

From the customer's perspective, your Web site is a black box. Customers don't know (or care) how many servers are involved, where they are, what hardware they employ, or which application server they use. They care only about how fast your Web page appears. Monitoring the end-user view tells you whether you have a publicly visible performance problem. Once you recognize that you have a performance problem, the challenge is to isolate the problem and be able to drill down to resolve it.

Following are two real-world examples of how you can diagnose J2EE performance problems by taking a transaction-centric view of the system.

### IMPROVING A SLOW TRANSACTION

In this particular example we examine the Sun PetStore reference application under a load test with the configuration shown in Figure 1.

#### *Detection*

First, identify slow transactions. Our first objective is to see how the system behaved over the length of the test run. By looking at the load driver response-time metrics, as well a set of indicating metrics, we get an idea of how our application performs.

As you can see in Figure 2, response time for the application is extremely slow throughout the entire run and gets progressively worse. In order to begin improving performance, the next step requires finding what components within the system are slow.

#### *Diagnosis*

Next, identify in which tier the problem transaction is spending its time. Once a problematic transaction has been located, we can identify which server(s) are responsible for the unacceptable response times. By breaking request response times down into their constituent server response times, a single performance team member can immediately identify which server to target to further their investigation into the root cause of the problem. For example, perhaps the customer login request is slow. But is it the Web server or one of the application servers? Or is it the database server?

In Figure 3, we see that the commitorder request is the most expensive transaction with respect to time. In fact, we can easily see how much time this request is taking in each tier, in this case 54.2% of its time is spent in JDBC calls. This warrants further investigation in order to improve this number.

#### *Root Cause*

Next, break the transaction into its logical Java components. By focusing on the problem transaction, we can

break the transaction into its logical and physical Java components that make up the request. A powerful visual map of the transaction's round-trip execution with timing information and a critical performance path aids in this process. Color-coded highlighting helps us narrow down the search to an offending component.

Even without any prior application knowledge, you can easily identify suspect components, right down to the method level, for further investigation by the appropriate functional expert. Which components and methods are slowest for the application's business-critical requests? Which methods or components are being called excessively? Are the EJBs interacting efficiently with the database, and using properly tuned SQL queries?

With the ability to view the component taking the greatest amount of time in the call graph, we can quickly focus our attention on that particular component. In this example, we know that the commitorder request was spending just over 54% of its time in a JDBC call. Figure 4 shows how we can use a transaction map to zoom in and narrow down to the specific area of code, in this case a JDBC query that's taking up this time.

By taking a closer look at the "hot" components, we see that both the parent getItem() and the JDBC executeQuery() call counts are low (7), but the ResultSet.getString() call count is high (672). ResultSet is called approximately 96 times per call to getItem().This is suspiciously high and results in the lengthy response time for this transaction. Now that we are able to isolate a single class as the root cause, a possible corrective measure would be to examine the method under question and optimize the SQL statement.

In this simplified example of focusing on the end-user transaction for performance root-cause analysis, we can see the benefits of tracing the request through a distributed J2EE system and provide some visibility into what once was a black box to engineers.

## ANALYZING INTERMITTENT PERFORMANCE PROBLEMS

Another common pattern is performance problems that occur partway into a test run, or after a server has been running in production for some period of time. Such problems are less likely to be solved by simple code-path improvements. A more likely cause is resource exhaustion somewhere in the system.

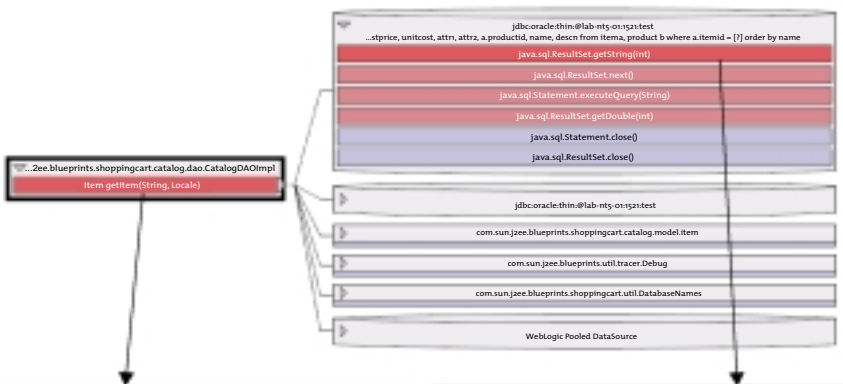
In addition to simply isolating application response times, this type of problem often requires correlating metric data from the application server or other server components. Let's look at an example, again using PetStore, of solving a performance problem that does not appear until later in a test run.

### *Identifying Problematic Time Periods for Key Metrics*

Again, our first objective is to try to get a picture of how the system behaved over the length of the test run. By observing the critical thresholds, we can quickly isolate the point at which the response time for our transactions started to increase.

### *Correlating other Metrics*

By viewing individual metrics in relation to time and to each other, we can start to isolate the root cause of our problem. As shown in Figure 5, we observe that about one-




**Method:**
com.sun.j2ee.blueprints.shoppingcart.catalog.dao.CatalogDAOImpl.getItem(String, Locale)

PetstoreCluster

| | Total | Avg. | Max. |
|---|---|---|---|
| Cumulative Time (s) | 0.457 | 0.065 | 0.093 |
| Exclusive Time (s) | 0.118 | 0.016 | |
| Parameter Bytes Serialized | 0 | 0 | 0 |
| Return Bytes Serialized | 0 | 0 | 0 |
| Call Count | 7 | | |
| Exceptional Exits | 0 | | |

**JDBC:**
jdbc:oracle:thin:@lab-nt5-01:1521:test
java.sql.ResultSet.getString(int):
select itemid, listprice, unitcost, attr1, attr2, a.productid, name, descn from item a, product b where a.itemid = [?] order by name

PetstoreCluster

| | Total | Avg. | Max. |
|---|---|---|---|
| Cumulative Time (s) | 0.148 | 0.000 | 0.016 |
| Exclusive Time (s) | 0.148 | 0.000 | |
| Parameter Bytes Serialized | 0 | 0 | 0 |
| Return Bytes Serialized | 0 | 0 | 0 |
| Call Count | 672 | | |
| Exceptional Exits | 0 | | |

**FIG. 4: ISOLATION OF A HOT SPOT**

quarter of the way through our session the response time starts to increase, as does disk-write activity on our WebSphere server. During normal operation, WebSphere Application Server should not be writing to disk because it is time consuming and it should not be necessary. This warrants further investigation.

We can see that the response time of transactions suddenly increases, which would indicate that perhaps this is not an algorithmic problem but one of resource contention. There is correlation with the disk-write activity that exceeded our thresholds, so there is something of a cause-and-effect relationship. Looking a bit more closely at passivation in WAS, we see an immediate increase in passivates at the same time we're running into response-time increases – which explains the increase in disk-write activity. To help confirm the impact on WAS, we see the throughput of the passivation request increase 100 times within the same time slice.

Remember, stateful session EJBs are maintained (in a cache or on disk) by WebSphere. Passivation is the process by which WebSphere Application Server removes an EJB from the cache while preserving the EJB's state on disk. Once the cache size is reached, WAS has to passivate EJBs to disk. Excessive EJB passivation can significantly slow down an otherwise well-performing system.

Now that we have isolated the problem to excessive EJB passivation, we should examine the application to ensure that the best practice of removing stateful sessions beans is being followed. We might also find it necessary to increase the cache size.

**FIG. 5: CORRELATING METRICS**

## Summary

In this article, we discussed the importance of establishing and monitoring performance goals. In order to meet your performance objectives, you need to be able to detect, diagnose, and resolve performance problems. Focusing on end-user transactions helped us put various key system and application metrics into the context of the user's experience with the application to help analyze and solve some WebSphere performance problems.

# TESTING WEB SERVICES

## Methods for ensuring server and client reliability

BY JIM CLUNE AND LUKE CHEN

**Solid testing techniques are essential for developing robust Web services because Web services' flexibility and connectivity provide an increased opportunity for errors. Problems can be introduced in any of a service's multiple layers, and even the slightest mistake can cause the entire service to fail.**

ABOUT THE AUTHOR

Jim Clune, Parasoft development manager, supervises the development of prevention and error detection software for Web and enterprise development. He dedicates much of his time to SOAPtest, Parasoft's new tool for testing Web services. Jim holds a BS in engineering from Harvey Mudd College and an MS in applied computer science and technology from Azusa Pacific University. He is pursuing a PhD at UCLA.

E-MAIL
jim@parasoft.com

In order for a complete Web service to deliver the promised functionality, both the client and the service must satisfy a number of requirements. Interfaces must be correctly described in a WSDL document. Messages must conform to both the transport protocol specification (such as HTTP 1.1) and the message protocol (such as SOAP 1.1). Messages must also conform to the contract specified in the WSDL describing the service, both in terms of the message content and the binding to the transport layer. Add to the mix security provisions, interoperability issues, UDDI registration requirements, and performance requirements under load, and it is easy to see why Web service testing is not a trivial matter.

This article explains general best practices that developers of Web service servers and/or clients can apply to ensure service functionality, interoperability, and security. For developers of Web services (producers), it explains potential problems and describes techniques for exposing those problems. For developers of Web service clients (consumers), it describes techniques for verifying that the client correctly connects to the server, sends a correct message, and gracefully handles fault conditions. In addition, it discusses interoperability, security, and UDDI registry issues that affect both Web service producers and consumers. The bulk of the discussion assumes the use of WSDL for describing the service, HTTP for the transport layer, and SOAP for the messaging layer.

### Server Testing

There are three main categories of Web service testing:

- ***Functional testing:*** Verifies that the service functions correctly
- ***Regression testing:*** Detects whether a regression is introduced
- ***Load testing:*** Verifies whether the service meets performance and functional requirements under load

We explore each type of testing in the sections that follow.

#### FUNCTIONAL TESTING

Functional testing is typically the first step in testing a Web service server. (If the server does not work correctly, its performance, security, interoperability, etc., are essentially irrelevant.) The goal of this testing is fairly straightforward: to ensure that the server delivers appropriate responses for the

given requests. However, due to the complexity of Web services, this task is far from simple. With most Web services, it is impossible to anticipate exactly what types of requests clients will send. Enumerating all possible requests is not feasible because the space of possible inputs is either unbounded or intractably large. As a result, it is important to verify whether the server can handle a wide range of request types and parameters.

There are two main steps to each functional test:

1. ***A test client sends a request to the server over an HTTP connection.*** This involves determining what types and ranges of requests need to be tested to determine whether the server will react appropriately to the wide variety of requests it might receive. Once you have determined what requests to send, tools such as those available in WebSphere Studio Application Developer (WSAD) can facilitate the creation and execution of test clients.
2. ***The response is analyzed for correctness (either by inspection or by running the response through a tool or script that verifies conformance to a specification).*** This analysis can be as simple as performing a text comparison with the expected response or as complex as extracting specific information from an XML document and performing application-specific checks.The simplest possible functional test involves sending a request and checking whether the server returns a response or an error message. For example, assume we have a sample employee Web service that allows queries by last name and returns the results in the form of an XML document. The most basic functional test would involve sending a valid input parameter (a last name entered into the system) and checking whether a response or an error message was returned.

Although these types of simple tests provide an adequate way to begin testing, they cannot verify the service's more complex functionality requirements. Fully testing even this simple service's functionality requires checking for all of the following notions of failure.

1. The attempt to open a socket to the URL of the Web service fails. This indicates a network problem or an incorrect URL or IP address.
2. The Web service returns a fault, such as

```
<Fault SOAP-ENV:Server: service
'urn:EmployeeQuery' unknown>
```

   This indicates an error caused by the server or by the client, depending on the type of fault.
3. The Web service responds and does not return a fault, but the responding message is not readable by the client because of an interoperability issue. For example, either the server or the client (or both) might not resolve XML namespaces in accordance with the standard.
4. A response is received, but not in the expected format. For example, the response is in an incorrect XML format or some other arbitrary text format. This type of error can be detected by validating XML with respect to an XML Schema.
5. A response is received in the format expected, but the data contained is incorrect (for example, when we request records for Fett but receive records for Kenobi).

The symptoms of the first three problems are independent of any particular Web service because they fail at the levels of the HTTP and SOAP protocols, which are consistent across Web services. The fourth and fifth problems can also arise in any Web service, but their details, and therefore their detection, are necessarily application specific. For example, a different Web service might accept a ticker symbol and return a stock quote. In this scenario, an expected response (SOAP envelope omitted for clarity) might be

```
16.87
```

or perhaps

```
<Quote symbol="AMZN" value="16.87"/>
```

This is an example of receiving a response in an unexpected format. The details of checking for this type of error depend on the specific service because different services can have different response types.

Each of the potential failure types exhibits different symptoms when encountered in the test client. The first three problems typically result in exceptions; ideally, the client will catch these exceptions, record them as test failures, and continue testing. The format can be verified by parsing the response with a validating XML parser, or in any other way that your testing infrastructure allows. Detecting incorrect results in the correct format is the most application-specific test. It generally requires using tools that allow you to make sophisticated assertions about the service's responses, or writing code that parses the XML and tests for constraints. In the case of the incorrect last name from the employee service, the test needs to verify that the lastName attribute of each Employee element matches the last name specified for that particular query. The best way to implement this verification depends on your test client and your verification capabilities. Although WSAD does not currently provide this level of verification, its functionality can be extended with third-party tools such as Parasoft SOAPtest. Another approach is to write an XSL file that outputs an error message when applied to nonconforming outputs.

While you are performing functional testing, remember that a Web service has multiple layers and that errors can be introduced at each layer. There are transport-level errors, such as an incorrect content length specified in an HTTP header, message-level errors, such as an invalid SOAP envelope, and application-level errors, such as a getStockPrice operation returning the price for the wrong ticker symbol. Keeping the layers in mind is helpful for generating adequate test coverage, as well as for debugging failures.

Also, be aware that the WSDL can be another source of errors. If the WSDL permits a wider class of inputs than the application, it is increasingly vulnerable to erroneous input at the application level. Ideally, a service's robustness would be tested by using the type definitions from the WSDL to generate all possible inputs and send each combination to the server. In practice, this is not feasible because the input space is usually much too large. A more pragmatic goal is to cover a representative portion of the input space.

After you confirm that the server handles expected requests correctly, perform fault checking to see how it handles unexpected input. The system will inevitably be faced with unexpected requests either as a result of mistakes (such as a bad WSDL) or from attempts to breach

ABOUT THE AUTHOR

Luke Chen is a software engineer currently working on SOAPtest, Parasoft's new tool for testing Web services. He also serves as Parasoft's technical representative to the WS-I. He holds an MS and PhD in nuclear physics from Michigan State University and a BS in physics from Harvey Mudd College. Luke joined the Parasoft software development team in 2000.

E-MAIL
lchen@parasoft.com

service security. (Hackers sometimes trick applications into behaving unexpectedly by sending invalid inputs.) Performing this fault checking involves sending the service requests with illegal and/or unexpected parameters, then verifying the response with assertions, custom code, or other tool-specific verification methods. The expected service behavior in these situations can depend on the stage of development as well as whether the Web service is intended for public versus internal use. For an internal service, it might make sense for the service to display its stack trace when a runtime error occurs, because the stack trace offers very valuable information for debugging. For a publicly exposed Web service, displaying the stack trace is arguably undesirable because it provides additional information about your implementation details (details that you would prefer hackers not know).

WSAD offers considerable flexibility for producing Web service clients. A standard client (shown in Figure 1) can be generated concurrently when Web services are generated and deployed.

If needed, this standard client can be customized graphically or programmatically in the built-in JSP editor (shown in Figure 2). The combination of these client-generation and customization options provides the opportunity to perform a broad range functionality testing.

## REGRESSION TESTING

After you have verified the server's functionality, rerun the functional test suite on a regular basis to ensure that modifications do not cause unexpected changes or failures. A common technique is to send various requests, manually confirm the responses, and then save them as a regression control. These regression tests can be incorporated into a regular automated build process. When regression tests are run frequently, regressions are easy to fix because they can be directly attributed to the few changes made since the last time the test was run. WSAD does not currently provide an explicit regression testing feature, but this capability can be added by extending WSAD with additional tools.

## LOAD TESTING

The next step in the server testing process is load testing. The goal of load testing is to verify the performance and functionality of the service under heavy load.

The best way to start load testing is to have multiple test clients run the complete functional test, including request submissions and response verifications. When load testing ignores the functionality verification process and focuses solely on load-rate metrics, it risks overlooking critical flaws (such as functionality problems that surface only under certain loads).

To thoroughly test the service's performance, run the functional test suite under a variety of different scenarios to check how the server handles different types of loads. For example, the test could check functionality and response time under different degrees of load increases (sudden surges versus gradual ramp-ups) or different combinations of valid and invalid requests. If the load tests reveal unacceptable performance or functionality under

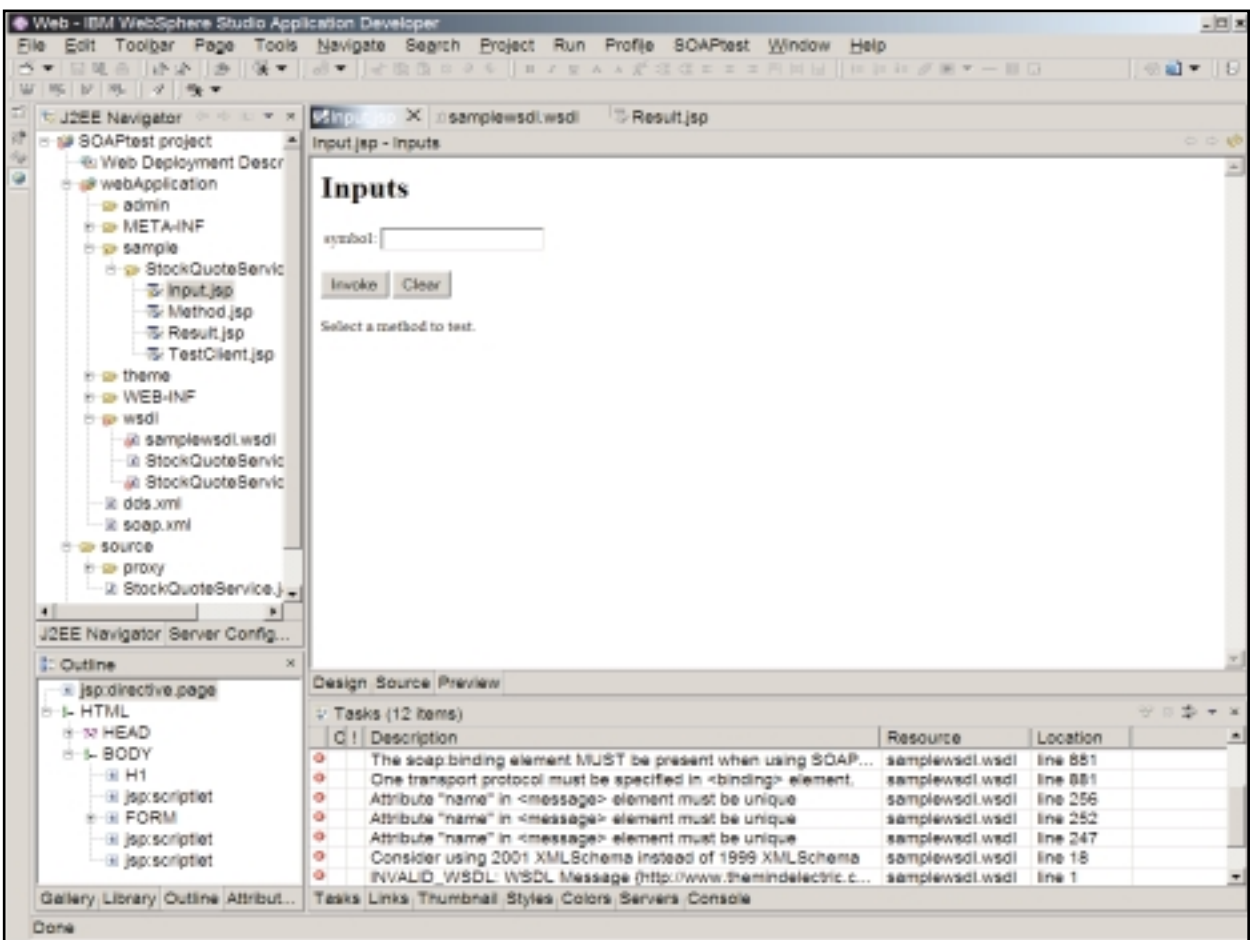



**FIG. 1: EXERCISING THE SERVICE WITH AN AUTOMATICALLY GENERATED TEST CLIENT**

load, the next step is to diagnose and repair the source of the bottleneck. Sometimes, the problem is caused by a fundamental algorithmic problem in the application, and the repair could require something as painful as an application redesign and rewrite. Other times, it is caused by some part of the infrastructure (the Web server, the SOAP library, the database, and so forth). In these cases, fixing the problem might be as simple as changing a configuration or as complex as changing the architecture. Because fixing performance problems sometimes demands significant application or system changes, it is best to start load testing as soon as possible. By starting early, you can diagnose and fix any fundamental problems before it is too late to do so without a major rewriting or rebuilding nightmare.

WSAD does not currently provide load testing functionality. It allows you to generate a large load by writing custom client code that uses a loop, but this is not the preferred approach. Load testers provide more control over how the load is generated because they allow you to control parameters such as test duration and load size.

## Client Testing

SOAP client developers are responsible for ensuring that the client sends requests properly. If a client sends invalid or improperly formed requests, the server usually cannot deliver the expected results. The process of testing clients is a little different from testing services because clients are the initiators of Web service interactions. This means that from a testing standpoint, there are two main things to verify: whether the client can correctly initiate an interaction by sending a request, and whether the client behaves correctly when it receives a response. Note that the second part requires inspection of the client application; it cannot generally be determined by merely observing wire traffic.

The best way to test a particular client depends on the nature of the application. If the client accesses a server that can accept "test" requests with no harmful side effects, it can directly access the live server during testing. If the server is not yet available or should not be sent test inputs, the client can access an emulated server or server stubs during testing.

No matter what type of server a client accesses, the same general principle applies: the client sends a request, the server responds, then client success or failure is determined by recording and verifying the request and/or by verifying the server response. (The same techniques and tools used to verify server functionality can be used for this purpose.) Of course, server bugs could mislead you: if the server is not operating correctly, correct client requests might result in incorrect responses, and incorrect requests might result in apparently correct responses. You can ensure that server functionality problems are not confusing your results by (1) verifying the request as well as the response, and (2) testing the simplest possible server implementations (server stubs) instead of – or in addition to – testing actual, complex servers.

After you verify that the client sends acceptable requests and can receive responses, shift to testing exceptional cases. For example, test that the client behaves properly when the server goes offline by sending the response to an invalid URL. Or use server stubs to simulate the server sending the client invalid data.

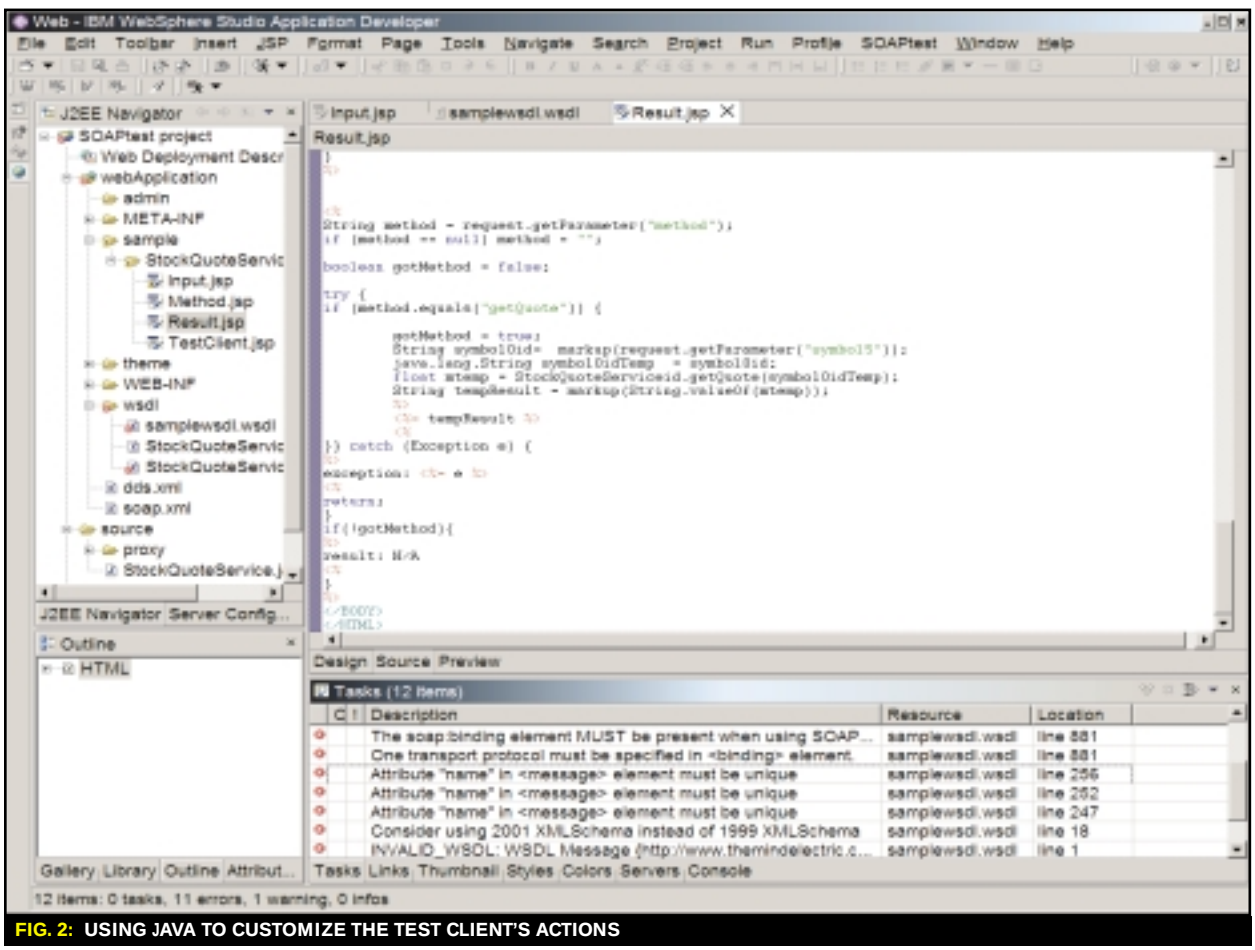



**FIG. 2: USING JAVA TO CUSTOMIZE THE TEST CLIENT'S ACTIONS**

Conference: January 21-24, 2003
Expo: January 22-24, 2003

The Javits Center
New York, NY

# LinuxWorld Conference & Expo

## Where OPEN Minds Meet

Linux is thriving! Financial services, telecommunications, government and retail are utilizing this amazing technology. Shouldn't you?

- Attend Keynotes and learn from inspiring visionaries who are devoted to Linux and Open Source.
  - Hector Ruiz, Ph. D., *President & CEO, AMD*
  - Jeffrey M. Birnbaum, *Managing Director & Global Head of Enterprise Computing in the Institutional Securities Division, Morgan Stanley*
  - Michael Tiemann, *Chief Technical Officer, Red Hat*
  - Steven A. Mills, *Senior Vice President & Group Executive, IBM Corporation*
  - Randy Mott, *Senior Vice President & Chief Information Officer, Dell Computer Corporation*

- LFS LINUX FINANCIAL SUMMIT

  All-new - Linux Financial Summit is an executive program created to address the growing demand for quality information and insight on Linux and Open Source solutions for the financial industry.

  *Sponsored by:* egenera, hp invent

- ENTERPRISE SOLUTIONS CENTER

  Check out the all-new Enterprise Solutions Center, where you'll experience multi-vendor Linux and Open Source enterprise solution demonstrations.

  *Presented by:* LINUX INTERNATIONAL, WiLD OPEN SOURCE

- Participate in LinuxWorld's world-class education program and learn how Linux has been implemented in financial services, communications and government, gain technical and enterprise know-how in areas such as sys admin, business and emerging technologies.
- Explore the newest technologies from leading Linux and Open Source suppliers and a growing array of start-ups.
- Take your seat at The Golden Penguin Bowl, the lively quiz show.

## www.linuxworldexpo.com

IDG WORLD EXPO

**Cornerstone Sponsor**

hp invent

**Platinum Sponsors**

AMD, IBM, intel, redhat, Sun microsystems — We make the net work.

**Silver Sponsors**

macrovision — Globetrotter Software Division

UNITEDLINUX

Although WSAD does not currently offer a direct client testing feature, it is possible to test a particular client by writing a test service that performs the desired analysis on the client request.

## Other Testing Considerations

Functional testing and load testing are the most fundamental types of testing for Web services. Depending on the type of service being tested and its requirements, it might be necessary to address additional issues during the testing process. Some issues that might further complicate many developers' testing are interoperability, security, and UDDI registry use.

### INTEROPERABILITY

A driving force behind Web services is the promise of seamless interoperability for disparate programming languages, operating systems, and various runtime environments. Unfortunately, the mere adoption of technologies that promote this idea (XML, SOAP, WSDL, UDDI) does not make the promise a reality.

Ideally, interoperability would be verified by checking that a service adheres to a comprehensive, universally implemented set of standards. However, the existing W3C recommendations are still evolving. Furthermore, the technologies are flexible enough to provide implementers with a myriad of options (document versus RPC-style, SOAP encoding versus literal encoding, different array representations, different versions of HTTP, SOAP, WSDL, UDDI, etc.). Flexibility is generally beneficial, but if everyone chooses a different way to do things, it does not serve the goal of interoperability. As options proliferate, it becomes increasingly unlikely that any given vendor solution will completely conform to all aspects or options allowed by the standard.

Given the reality that not all of the standards today are fully developed or consistently implemented, one of the most pragmatic approaches to interoperability is the one taken by the Web Services Interoperability Organization (WS-I). WS-I, though not itself a standards body, intends to serve as a standards integrator by developing a core collection of profiles that are a subset of the various Web service technologies. By restricting development to technologies specified in WS-I profiles, developers can increase the odds that their systems will interoperate with other systems. Development tool companies are already working with the WS-I to develop tools that automatically check compliance with these profiles and, in the event of noncompliance, pinpoint exactly what needs to be changed to ensure compliance. Expect to see tools that check compliance with these profiles soon after the profiles are officially released.

### SECURITY

Web services security is not a single problem, but rather a host of interrelated issues. For any given application, some of the issues will be critical, while others may be of lower priority or even irrelevant. Some facets of security worth considering when deploying Web services are:

- ***Privacy:*** For many services it is important that messages are not visible to anyone except the two parties involved. This means traffic will need to be encrypted so that machines in the middle cannot read the messages.
- ***Message Integrity:*** Provides assurance that the message received has not been tampered with during transit.
- ***Authentication:*** Provides assurance that the message actually originated at the source from which it claims to have originated. You may need to not only authenticate a message, but also prove the message origin to others.
- ***Authorization:*** Clients should only be allowed to access services they are authorized to access. Authorization requires authentication because without authentication, hostile parties can masquerade as users with the desired access.

Security impacts testing requirements in two important ways. First, any security requirements for a Web service naturally translate into testing requirements. If a service requires a certain level of privacy, or if it requires that messages be authenticated in a certain way, then specific tests are needed to ensure that these security requirements are met. The second way that security impacts testing is subtler. To some extent, security schemes will complicate the process of testing and debugging the basic functionality. For example, nonintrusive monitors can often aid in functional testing as well as load testing. Encrypted traffic presents an obvious complication to this approach to testing.

### UDDI

Thus far, we have not addressed how publishing and discovering services fits into testing. UDDI is not yet as mature a technology as some of the others discussed in this article, but it is evolving and gaining acceptance. Services registered in a UDDI registry that are discovered and bound dynamically have all the testing requirements that we have already discussed, plus the find and bind features require additional testing. It is helpful to consider UDDI testing in two pieces: the registry implementation and the entries within the registry. Most users will not be implementing their own UDDI registry, so the primary focus will be on testing the content of the registry. The most reliable way to test the content is to write test clients that perform inquiries on the registry, and then use the registry data to actually invoke the service. Functional testing of services can then be extended to include dynamic binding to endpoints specified in a UDDI registry. This ties together the functional testing of the registration with the service implementation. The current version of WSAD provides full UDDI support, which allows both querying and registration.

## Conclusion

By integrating the discussed testing practices into the Web service development process, you can ensure that a Web service server works well with the possible types and volumes of client requests, and that a Web service client correctly accesses and retrieves whatever data a service has to offer. You can start implementing the discussed practices at any point in the development process, but if you start testing early, you will maximize your ability to prevent errors as well as detect errors. Typically, the earlier you detect a problem, the easier it is to fix it, and the less chance you and your team members have to inadvertently worsen the problem by building code or components that interact with the problematic element, or by reusing the problematic element for other servers or clients. If you start your testing as early as possible, then continue using the related tests as a regression test suite throughout development, you will not only ensure the client's or server's continued reliability, but also streamline the development process.

# ATTN: Wireless Developers...

# WDJ JAN'03


**Coming in January:**

**SYS-CON Media's NEW, EXPANDED COVERAGE of wireless development…and developments in wireless**

► WDJ focuses on wireless solutions from the perspective of everyone from senior IT management to the individual wireless developer, both corporate and independent.

► WDJ features regular monthly columns by the world's top developer-writers on the **major** wireless platforms, the **major** wireless hardware, and the **major** wireless applications... with the accent firmly on drilling down into every aspect of Wireless That Works.

► Bill Ray, WDJ's Editor-in-Chief, has been developing handheld applications for 16 years – from a mobile data gathering system deployed in 1986 to Bluetooth games still in development. He has worked with every category of application, including the world's first Java-capable DTV set-top box, which he developed for Swiss Telecom.

**Bill Ray**
Editor-in-Chief

# **Wireless Solutions** That Work!

## *Exclusive Developer Coverage*

*Look for specific "developer" columns covering:*

| ***DEVICES:*** | ***PLATFORMS:*** | ***TECHNOLOGIES:*** |
|---|---|---|
| *Palm / Handhelds* | *Palm OS* | *Bluetooth* |
| *HP / iPAQ* | *Symbian OS* | *802.11* |
| *Sharp / Zaurus* | *WinCE* | *SMS / EMS / MMS* |
| *RIM / BlackBerry* | *J2ME* | *SIM Phase 2+* |
| *Dell / Axim X5* | *BREW* | *and more...* |
| *Nokia / Communicator* | *.NET Compact Framework* | |
| *and more...* | *Linux* | |
| | *Nokia Series 60* | |
| | *and more...* | |

***SUBSCRIBE TODAY!***

***1-888-303-5282***

***WWW.SYS-CON.COM***

***SPECIAL LIMITED TIME OFFER***

***ONLY $49.99 FOR 12 ISSUES***

**COMING JANUARY '03** ► wireless DEVELOPER'S JOURNAL

SYS-CON MEDIA


# Implementing EJB 2.0 container-managed relationships under WAS 5.0

BY JOSEPH K. KROZAK

One of the exciting new features of the WebSphere Application Server (WAS) 5.0 release is support for J2EE 1.3. With this release of the J2EE specification comes EJB 2.0, which contains a number of features that effectively make both session and entity beans far more flexible and scalable components of an application's architecture.

Major advancements in container-managed persistence (CMP), along with new capabilities such as local interfaces, combine to provide far greater consistency and efficiency for EJBs than was ever possible with earlier versions of the specification. Another important capability introduced in EJB 2.0 is the ability to specify and implement relationships between entity EJBs. These container-managed relationships (CMR) allow developers to declare relationships of various cardinality between entity beans, while deferring the actual implementation and enforcement of these relationships to the underlying container. Collectively, all of these EJB 2.0 features make it possible to implement complex associations between fine-grained entity beans – a capability that has been desired since the introduction of entity beans to the EJB specification.

This article will take a look at CMR, along with the CMP advancements and new local interface features that make up the EJB 2.0 specification and that will be supported by WAS 5.0.

## EJB 2.0 & WAS 5.0: Solid Underpinnings for CMR

Before delving into CMR, it is important to first address the other features of EJB 2.0 upon which CMR depends. The first feature is the advanced CMP model introduced with EJB 2.0. In this new version of the specification, the persistent state of an entity bean can no longer be explicitly defined in the entity bean class as attributes. Instead, developers must provide abstract accessor ("getter") and mutator ("setter") method declarations within an abstract bean implementation class. Upon compilation and subsequent deployment, the J2EE container will supply the appropriate implementations for these abstract methods, including the automatic storage and retrieval of bean attribute values to and from the underlying persistent store. The elimination of explicit bean state attributes means that such attributes can no longer be explicitly addressed in code. Instead, they can only be addressed via their respective getter and setter methods. An example of this change is illustrated in Listings 1 and 2, in which an extract of a Loan entity bean is shown under the EJB 1.1 (pre-WAS 5.0) and EJB 2.0 (WAS 5.0+) paradigms, respectively.

As a comparison of the listings shows, it will be necessary for applications employing EJB 1.1 entity beans to be refactored upon migration to WAS 5.0. These changes promote bean encapsulation and consistency of access, which were somewhat compromised under previous versions of the EJB specification. In addition, it will be shown that the consistent use of getter and setter methods to access bean attribute state under CMP will be extended to get- and set-related bean instances under CMR.

WAS 5.0 extends the power of EJB 2.0 CMP with its support for advanced data sources. Under EJB 2.0 it is possible to use CMP to map a bean's state to a nonrelational data source, such as an object-oriented or hierarchical legacy database. Most EJB 2.0–compliant application servers overlook this subtle point, relying solely on JDBC data sources. Instead, WAS 5.0 employs Java Connector Architecture (JCA)–based resource adapters to manage the physical connections beneath its data sources. In fact, WAS 5.0 developers who wish to use CMP with traditional relational databases must use a new JCA-based Relational Resource Adapter. This architecture makes it much simpler to provide access to nonrelational data sources. This concept is especially compelling when one considers the possibility of supporting CMR relationships between CMP

ABOUT THE AUTHOR

Joseph K. Krozak is vice president of technology development for Krozak Information Technologies, Inc. (www.krozak.com), a leading supplier of advanced software solutions to Fortune 500 and midmarket companies. Joe has over 15 years of experience in the information technology industry, including 10 years of extensive experience building large distributed-object and component-oriented systems.

E-MAIL
jkkrozak@krozak.com

beans spanning vastly different kinds of data stores, as illustrated in Figure 1.

Another EJB 2.0 feature that will be supported by WAS 5.0 – and that is crucial to CMR support – is the local interface. Local interfaces were, in part, created to address the networking overhead associated with using traditional remote interfaces. Local interfaces are essentially pure Java interfaces that do not extend java.rmi.Remote. As such, they incur none of the networking or serialization overhead of their remote counterparts. The introduction of local interfaces means that it is now possible, and entirely appropriate, to design EJB applications using fine-grained entity beans. CMR relies upon local interfaces to establish and maintain relationships between entities. In fact, any bean that does not provide a local interface can only participate on the source side of a CMR relationship. CMR's reliance on local interfaces makes the implementation and enforcement of interrelated entities extremely efficient. Hence, with EJB 2.0 and WAS 5.0 it is entirely appropriate to designate robust 1-to-1 (1:1), 1-to-many (1:m), or many-to-many (m:n) relationships between fine-grained entities. Such designs were expressly forbidden in earlier incarnations of the EJB specification due to the enormous performance implications resulting from networking overhead. Entire design patterns, such as the J2EE Composite Entity, were created to deal with this overhead by creating coarser-grained entities. Under WAS 5.0, such patterns should, for the most part, no longer be necessary. In fact, unless absolutely necessary, such patterns should be removed from J2EE applications when migrating to WAS 5.0.

## Exploring CMR

Implementing CMR under WAS 5.0 requires a combination of support structures defined at the bean, database, and deployment descriptor levels. At the bean level, CMR support requires extending the familiar CMP concept of providing abstract methods for attribute access to methods that provide container access to the relationships between entity beans. In CMR, such relationships can be classified by cardinality and navigability. For simplicity, let's assume the existence of two entity bean classes, A and B. CMR can support 1:1, 1:m, or m:n relationships between instances of A and B. At the bean level, abstract relationship methods must be defined to allow the container to enforce and manage the relationship.

To explore CMR in more concrete terms, let's discuss its use in a simplified financial application involving loans, borrowers, and addresses. To further simplify our example, we will assume that these entities support only local interfaces. In our application, a given borrower can have one address and potentially multiple loans. Hence, the relationships are 1:1 for Borrower_Address and 1:m for Borrower_Loan. At the bean level, the Borrower_Address relationship would require that abstract methods be defined in the source bean (borrower) implementation class to allow the container to manage (i.e., get/set) the target address instance automatically.

Listing 3 illustrates the method declarations that would be required in the borrower interface and implementation classes to support the aforementioned relationship.

Borrower also has a 1:m relationship with Loan, so the Borrower local interface and implementation class must provide abstract methods to support this relationship in addition to the 1:1 relationship with Address. Assuming the relationship is navigable from Borrower to Loan, Listing 4 illustrates the methods required on Borrower to support its 1:m relationship with Loan. The key difference between these methods (besides the target bean class used) – and the methods supporting the 1:1 relationship between Borrower and Address – is the use of java.util.Collection as the parameter and return types for the 1:m relationship in lieu of the target bean class. A subclass of java.util.Collection, such as java.util.Set, could have been used as well.

At the bean level, many-to-many relationships are not appreciably different from one-to-many relationships. However, the differences are more interesting at the database level.

The underlying database is responsible for the long-term durability and persistence of container-managed relationships. CMR would be useless without a permanent account of how various bean instances are related. CMR relationships are persisted in the underlying database using the combination of primary and foreign keys representing the entity beans that are interrelated. Specifically, for 1:1 and 1:m relationships, the target (or many-sided) bean database record must contain a foreign key (FK) that references the source (or one-sided) bean instance. For example, to support the Borrower_Address 1:1 mapping discussed earlier, the underlying table for Address, say Address_TBL, must contain a column(s) that references the primary key (PK) of the specific Borrower instance to which the Address belongs. So, if borrowers are uniquely identified by social security number and addresses are uniquely identified by a "one-up" object identifier or OID, then each Address_TBL record must contain a foreign key to the social security number PK identifying the source borrower. Similarly, each Loan_TBL record representing a Loan bean instance must contain a foreign key to the owning or source borrower. These table relationships are depicted graphically in Figure 2.

Many-to-many relationships are particularly interesting at the database level. For m:n relationships, a junction table containing foreign key columns to the primary keys of the relationship participants must be introduced. Suppose, for example, that we adjust our example to support m:n relationships between borrowers and addresses. In this case, the table mappings shown in Figure 3 would be needed in order to support the relationship.

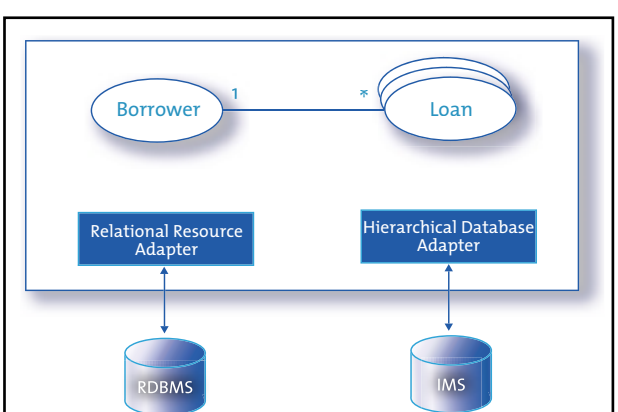



**FIG. 1: CMR ACROSS HETEROGENEOUS DATA SOURCES UNDER WAS 5.0**

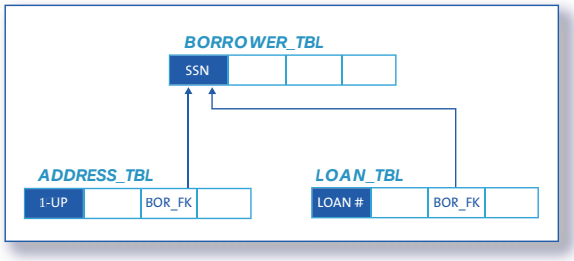



**FIG. 2: FK/PK MAPPINGS FOR 1:1 AND 1:M BEAN RELATIONSHIPS**

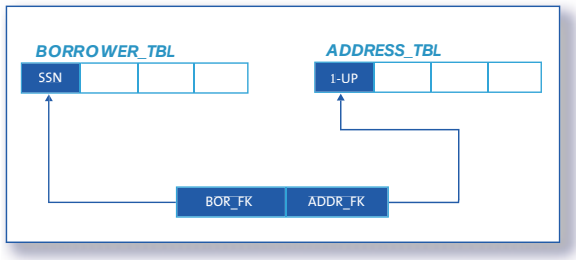



**FIG. 3: FK/PK MAPPINGS FOR M:N BEAN RELATIONSHIPS**

The last major step required to exploit CMR under WAS 5.0 is to make deployment descriptor–level declarations of the relationships. These descriptor definitions can either be created using the EJB Relations capabilities of the WAS Application Assembly Tool (AAT), or manually. Either way, the relationship definitions will ultimately exist within the EJB JAR's ejb-jar.xml descriptor file. A relationships XML stanza must be added after the enterprise beans XML stanza within ejb-jar.xml. Within the relationships stanza are one or more ejb-relation stanzas that define the specific semantics of a particular container-managed relationship between two beans. Each ejb-relation contains a name, along with the roles each bean plays within the relationship. The cardinality of the beans participating in the relationship is defined within these role definitions. Listing 5 provides an illustration of the CMR descriptor definitions required to model the Borrower_Loan 1:m CMR relationship.

Each major stanza can be optionally identified with an ID attribute, as shown in the listing. For the Borrower_Loan relationship, there are two roles – one for the borrower and one for the loan. The borrower is an owner of loans, and there is one borrower for many loans in the relationship. Hence, the borrower's multiplicity is specified as "one", and its bean is explicitly referenced within the ejb-name element of the relationship-role-source stanza. The subsequent cmr-field stanza declares the Borrower bean attribute that will be used to contain the loans associated with that borrower. In the spirit of CMP 2.0, the container will use the setLoans and getLoans accessor methods to manipulate this relationship, instead of explicitly accessing a declared attribute. The second relationship role covers loan relationships with borrowers. As such, its multiplicity is "many", and its "borrower" bean attribute will ultimately refer back to the owning borrower.

The CMR stanzas accommodate 1:1, 1:m, and m:n mappings between entity beans. Hence, there is considerable flexibility offered to cover the most demanding persistence relationships.

## Conclusion

WAS 5.0's support for EJB 2.0 and CMR will provide significant advantages to system architects and designers. Much of this functionality was previously deferred to bean-managed persistence (BMP) entities, external object-to-relational mapping products (like Oracle's TopLink), or simply not tackled for fear of the overwhelming complexity of implementing the solutions in code. With WAS 5.0, the power of declarative relationship management will allow architects to create simpler, yet more sophisticated and robust J2EE solutions.

## Reference

Alur, D., Crupi, J., Malks, D. (2001). *Core J2EE Patterns – Best Practices and Design Strategies, Java 2 Platform, Enterprise Edition Series*. Sun Microsystems Press/Prentice Hall.

### LISTING 1: LOANBEAN.JAVA

```
public class LoanBean implements EntityBean
{
…
     private String loanNumber;
     private String institution;
     private BigDecimal balance;
…

     public String someMethod()
     {
          String identification =
                this.loanNumber + …;
     }
…
}
```

### LISTING 2: LOANBEAN.JAVA REFACTORED

```
public abstract class LoanBean implements EntityBean
{
…
     private String loanNumber;
     private String institution;
     private BigDecimal balance;
…
     /* abstract getter/setter for each CMP field */

abstract public String getLoanNumber();

abstract public void setLoanNumber();
…
     public String someMethod()
     {
          String identification =
                getLoanNumber() + …;
     }
…
}
```

### LISTING 3: BORROWER AND ADDRESS ARTIFACTS

```
Supporting 1:1 Relationship
Borrower.java
import javax.ejb.EJBLocalObject;

public interface Borrower extends EJBLocalObject
{
  …
  public Address getAddress();
}

BorrowerBean.java
abstract public class BorrowerBean implements EntityBean
{
```

```
…
  /** container managed relationship with address */
  abstract public Address getAddress();
  abstract public void setAddress(Address address);
…
}

Address.java
public interface Address extends EJBLocalObject
{
…
  public Borrower getBorrower();
}

AddressBean.java
abstract public class AddressBean implements EntityBean
{
…
  /** container managed relationship back to Borrower */
  abstract public Borrower get Borrower ();
  abstract public void set Borrower (Borrower val);
}
```

LISTING 4: ABSTRACT BORROWER METHODS USE COLLECTIONS TO SUPPORT 1:M RELATIONSHIP WITH LOANS

```
BorrowerBean.java
abstract public class BorrowerBean implements EntityBean
{
…
  /** container managed relationship with loan  */
  abstract public Collection getLoans();
  abstract public void setLoans(Collection loans);
…
}
```

LISTING 5: CMR DESCRIPTOR DEFINITIONS

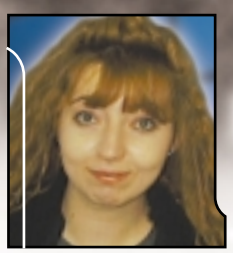
```
<relationships id="Relationships_1">
          <ejb-relation id="EJBRelation_1">
              <ejb-relation-name>BorrowerToLoan</ejb-relation-
              name>
              <ejb-relationship-role
              id="EJBRelationshipRole_1">
                 <ejb-relationship-role-
                 name>OwnerOfLoans</ejb-relationship-rolename>
                 <multiplicity>One</multiplicity>
                 <relationship-role-source id="RoleSource_1">
                    <ejb-name>Borrower</ejb-name>
                 </relationship-role-source>
                 <cmr-field id="CMRField_1">
                    <cmr-field-name>loans</cmr-field-name>
                    <cmr-fieldtype>java.util.Collection</cmr-
                    field-type>
                 </cmr-field>
              </ejb-relationship-role>
              <ejb-relationship-role id="EJBRelation
              shipRole_2">
                 <ejb-relationship-role-name>OwnedLoan</ejb-
                 relationship-role-name>
                 <multiplicity>Many</multiplicity>
                 <relationship-role-source id="RoleSource_2">
                    <ejb-name>Loan</ejb-name>
                 </relationship-role-source>
                 <cmr-field id="CMRField_1">
                    <cmr-field-name>borrower</cmr-field-name>
                 </cmr-field>
              </ejb-relationship-role>
          </ejb-relation>

 </relationships>
```

# UNDERSTANDING the GLOBAL PICTURE

## Managing WebSphere infrastructures and applications

BY MARINA GIL-SANTAMARIA



ABOUT THE AUTHOR

Marina Gil-Santamaria is a product manager for Unicenter Management for Web and Application Servers. During the past five years with Computer Associates, Marina has held various positions, including senior systems engineer and specialist across different divisions of CA. Marina holds an MS in electrical engineering from the Universidad Politecnica de Madrid, Spain.

E-MAIL

marina.gilsantamaria@ca.com

Application servers have transformed traditional commercial Web sites from simple publishing models to sophisticated environments and applications that are central to an organization's growth and operations. WebSphere must be understood in the context of an extended WebSphere environment.

Let's take a look at a typical WebSphere implementation. WebSphere environments include Web servers (WebSphere ships with IBM HTTP server powered by Apache, but it also integrates with Apache, IIS, Netscape, and IPlanet Web servers); caching servers; deployed applications; back-end systems, including databases such as DB2, Sybase, Oracle, and Informix; collaborative systems such as Lotus Domino; middleware like WebSphere MQ; and enterprise resource planning (ERP) systems (see Figure 1).

As we can see in Figure 1, applications are deployed across extended underlying infrastructures on both distributed and mainframe environments. This means that even though our main focus is WebSphere, we shouldn't forget that Web-based applications traverse a path of complex, interdependent entities, any of which is a potential bottleneck or failure point. And not only is every element of the chain responsible for presenting content to the end user, but problems can happen in the source code of a particular JavaBean, in a specific method's call to a back-end database, or even in a network router. Consequently, effective Web performance management requires the latitude to monitor heterogeneous systems and networks across the extended WebSphere infrastructure. It is important to use modular management tools with the same look and feel so complete infrastructures can be monitored the same way and from the same central location. We also need the depth to uncover potential causes of slowdowns at the lowest system and source-code levels.

### Focusing on WebSphere

Let's forget for now about the extended WebSphere environment scheme and focus on WebSphere and its underlying J2EE technologies. The J2EE application model divides enterprise applications into three fundamental parts: components, containers, and connectors.

- ***Components:*** These are the key focus of application developers, focusing on specifics such as business logic and user interfaces.
- ***Containers:*** These intercede between clients and components, providing services transparently to both, including transaction support and resource pooling.
- ***Connectors:*** These sit beneath the J2EE platform, defining a portable service API to plug into existing enterprise vendor offerings.

When building J2EE applications, organizations rely on services provided by vendors using a combination of standard, commercially available components in addition to their own custom components. This approach results in faster develop-

ment time, and portability across a range of enterprise platforms. But what happens from a management perspective? This model adds complexity to pinpointing where the performance bottlenecks or crashes are really happening. Are these problems due to the underlying services and therefore IBM's fault? Or does the fault lie with the commercially available component or the in-house–developed components? Finding the answer requires a management tool that will monitor both WebSphere infrastructures and deployed applications. Why? Because on the one hand you need to make sure that your WebSphere infrastructure is healthy and available to present content to your end users, but on the other hand you need to ensure that deployed applications are working properly and that return content is complete and accurate.

Keeping in mind WebSphere infrastructures, a management tool should monitor individual WebSphere containers and services performance, as well as system performance, to ensure that there are enough system resources to handle your traffic. For example, you should look for a management tool that will oversee things like JVM performance, JMS, garbage collection, disk space or threads, and memory usage metrics, to ensure that your WebSphere systems are not running out of resources, which could cause system instability, or fatal Out of Memory Exceptions. In addition, a management tool should provide monitoring of metrics associated with processors, file systems, and port availability. And of course, you really need a proactive management tool to monitor your WebSphere environments. It should allow you to configure warning and critical thresholds, along with automatic corrective actions, to automatically address critical problems as soon as they happen, without requiring IT intervention.

## Monitoring WebSphere Applications

As infrastructures are increasing in complexity, applications are becoming more and more complex as well. In fact, today's typical Web application can have more than 80 components distributed across different systems. Let's take a closer look at the J2EE application model shown in Figure 2. As you can see, it consists of a client-side presentation tier, server-side presentation tier, server-side business logic tier, and a data access layer within the enterprise information system tier.

## Monitoring the Front End of the Applications

As we can see in Figure 2, for server-side deployment of dynamic content, J2EE supports both Java servlets and JavaServer Pages (JSP) technology. JSP and servlets are part of the server-side presentation tier, forming the front end of an application, so it's crucial for WebSphere administrators to oversee the performance of these visible components. What if the average time taken to execute a servlet is increasing? What if it's starting to take longer for an end user to be authenticated on the login servlet? How can you detect those trends that are just beginning to appear in the application server before they develop into a critical situation? One of the things that you should look for in a management tool is the ability to help you monitor and manage the performance of the front end of your deployed applications by automatically discovering all JSP and servlets deployed on your environment, so you have a way of understanding and visualizing your complete front-end tier. For any deployed JSP or servlet, you should be able to monitor in real time performance parameters such as invocation total/second, longest execution time, and average execution time, so problems and bottlenecks can be easily identified for key servlets or JSP, and application problems can be tracked down to the offending servlet or JSP.

## Synthetic Versus Real-Time Monitoring

Vendors have been addressing J2EE management from two very different approaches: real-time versus synthetic monitoring. Real-time monitoring can help you detect application problems when they happen, as soon as they happen, and synthetic monitoring proactively monitors Web applications and allows prevention before the end user encounters application crashes or wrong data. An ideal management solution should deliver both approaches. For real-time monitoring capability, a management solution should deliver an "insider view" of the application to manage the transactional availability and performance of the application in real time, and have the capacity to pinpoint a performance or availability problem to an offending JavaBean home or remote interface, a servlet, or a database connection. In this context, crucial real-time transaction availability and performance metrics could be captured by dynamically instrumenting the target application with

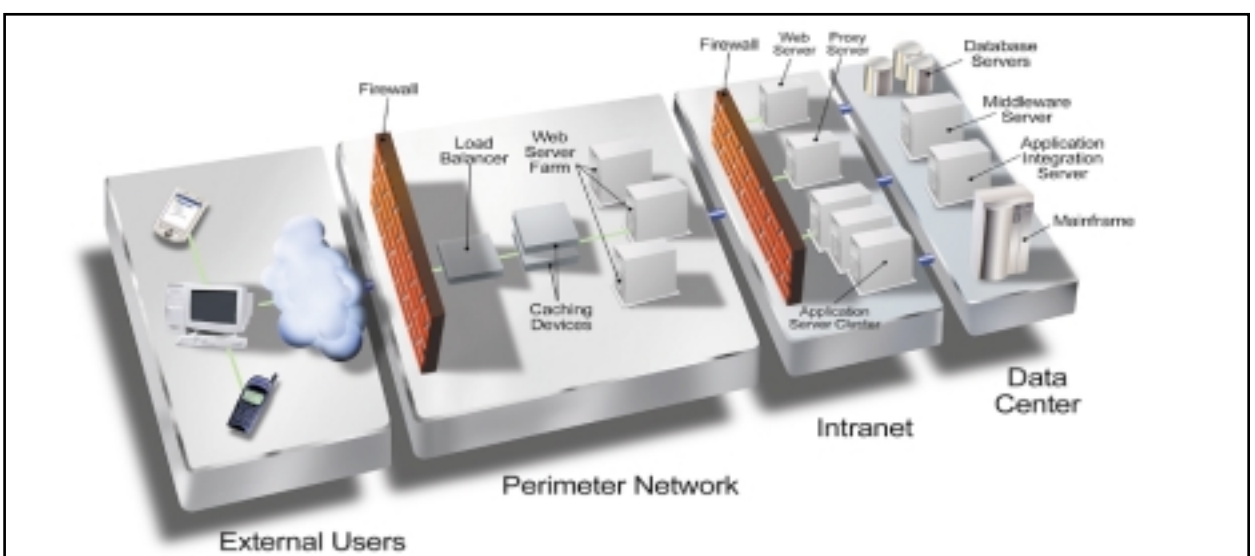



**FIG. 1: EXAMPLE OF AN EXTENDED WEB INFRASTRUCTURE ENVIRONMENT**

a watcher component. These dynamically inserted watchers do not make any persistent changes to the application code and do not require any foreknowledge of the application domain or business logic. At the same time, you should look for a management tool with abilities to periodically exercise your deployed applications, to help detect problems before end users encounter them. From a management perspective, you should start by identifying your real business objectives – for example, to get users to register for a technical seminar, or to sell an item from the inventory – to identify which Web transactions are critical in your environment. Each of these critical business scenarios or paths should be monitored and managed synthetically and in real time as a Web transaction.

## Monitoring JavaBeans

Monolithic applications of the past are now being replaced with JavaBean components. These components constitute the skeleton of the J2EE server-side business logic tier, and form the application's core (see Figure 2). EJB technology provides the ability to model a full range of objects by defining two distinct types of EJB components: session beans and entity beans. Session beans represent behaviors associated with client sessions, for example, a purchase transaction on an e-commerce site. Entity beans represent collections of data, such as columns in a relational database, and encapsulate operations on the data they represent. Entity beans are persistent; surviving as long as the data they're associated with remains viable. One of the requirements of an application server management tool should be real-time monitoring of individual JavaBean performance to ensure the overall health of the application. What if the number of rolled back transactions is reaching the maximum acceptance criteria for an EJB? Which EJB has the most requests? Specifically, you need a WebSphere management tool that will automatically discover and populate your GUI with the set of entity beans and session beans that have been deployed on your application server, and will automatically update the topology as new WebSphere applications are deployed and put into production. For each EJB, you should monitor key parameters such as: number of times accessed, cached beans, transactions committed/second, rolled-back transactions, transactions in process, cached beans in use, idle cached beans, and so on. Monitoring these parameters will help you detect performance problems. You should also take them into consideration for performance tuning, or when planning for WebSphere application server additions or configuration changes.

## Monitoring JDBC Connection Pools

Let's move next to the data access layer and JDBC connection pools. Each time a resource attempts to access a database, it must connect to that database. A database connection incurs overhead, as it requires resources to create the connection, maintain it, and then release it when it is no longer required. The total database overhead for an application is particularly high for Web-based applications. Web users connect and disconnect more frequently, and usually less effort is spent during the interactions themselves. WebSphere Application Server enables administrators to establish a pool of JDBC database connections that can be shared by applications to address these overhead problems and improve response time.

When a user makes a request over the Web to a resource, the resource accesses a data source, and when the request is satisfied and the response is returned to the user, the resource returns the connection to the connection pool for reuse. What if the number of requests waiting for connection is growing? What if the total number of available connections is decreasing? You need a management tool able to take corrective actions to ensure that there are enough resources in the pool to avoid poor application performance. Monitoring the status of every JDBC connection pool parameter – such as the number of active connections, the number of requests waiting for connections, the highest number of waiting connections, the longest waiting time, and the total number of available connections – is imperative to ensure the application's good health, to

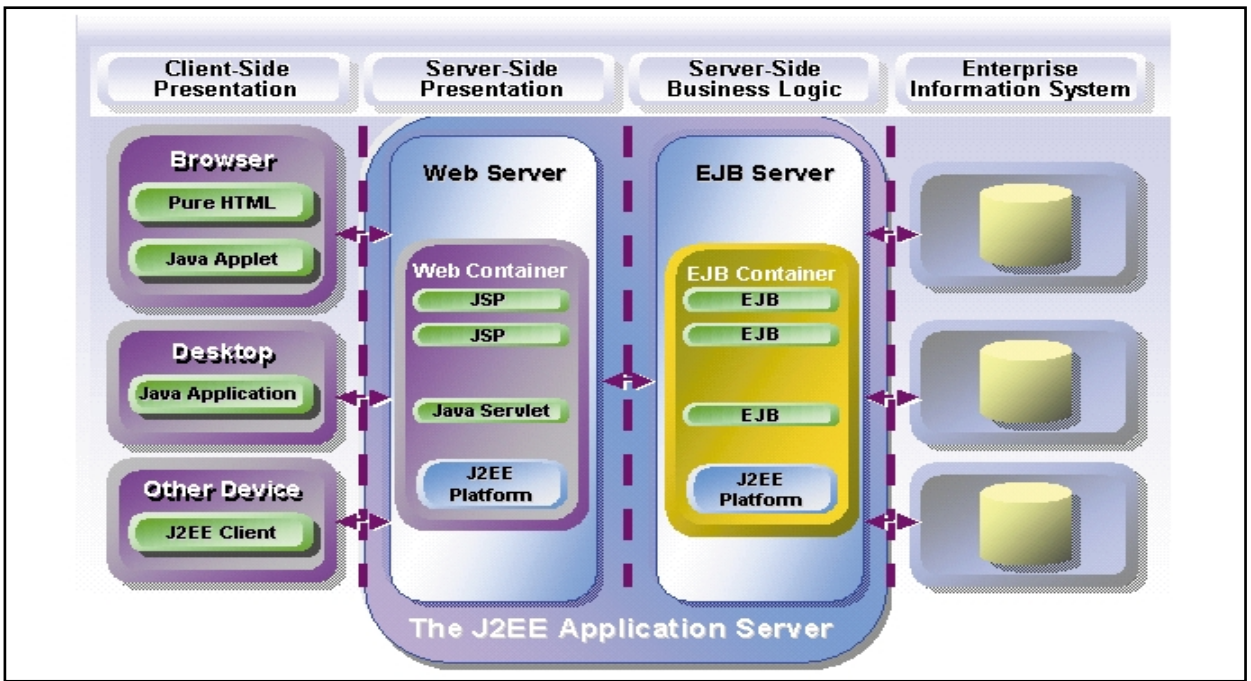



**FIG. 2:** J2EE APPLICATION MODEL

assure that transactions with back-end systems are executed in time, and to guarantee a good end-user experience.

## Monitoring Custom Business Logic

I like to define business logic as the term that describes "the action behind the button." When you are performing a transaction on a Web site and you click a button to see a list of available parts or pricing, you're interfacing with the application logic. A visually pleasing presentation layer is displayed to you, but behind the scenes the application is executing a number of predefined steps or logic, checking the inventory database for availability and checking the sales database for current pricing. The application presents a consolidated picture to you through the presentation layer (Web browser), but the action behind the button is carried out by the business logic developed into the application. Let me give you an example: consider a bank account EJB that provides deposit, withdraw, and balance-check methods.

A typical user interface, built using servlets, will use these methods to provide an ATM-like functionality through a standard Web browser. A customer could deposit money, withdraw money, and check the balance of the account (see Figure 3).

A management tool should really go down one more layer to monitor the custom business logic encapsulated in your deployed JavaBeans. It should allow creation of virtual user interactions to navigate through the J2EE infrastructure, "exercising" all your Java components and reporting back on application health as well as the performance of the actual business logic. A management tool should have a robust EJB autodiscovery functionality to provide a complete picture of your business logic as well as the methods and parameters that each method requires at execution time. Once this information is obtained, recorded interactions should be created to test critical transaction paths or flows (a set of steps) in a nonintrusive way. Recording each step should be as simple as selecting a JavaBean, picking the method to execute, and configuring the parameters associated with each method. Once the user has completed the wizard-based recording phase, a WebSphere management tool should periodically execute the complete flow at user-specified intervals, validating results against values previously stored in the recording phase, and monitoring parameters such as execution time, return value, and exception errors for each step. These synthetic monitoring capacities will provide you with a complete picture of the health and behavior of your business logic, and with the capabilities to correct and avoid application problems before end users run into exception errors or invalid return data.

## Managing the End-User Experience

Most end users have encountered errors such as "Error occurred while processing your request," or "The page cannot be displayed," or "Page not found," while shopping online, banking online, or surfing the Web. What is the result of these types of errors? If the Web application that is experiencing a problem is an internal one, employees are not able to do their job, so productivity decreases for them, as well as for IT, as they need to invest time to troubleshoot and resolve the problem. And if the Web application having the problem is an external one, what happens? Well, it is likely you will lose your customer's trust. If your Web site is not reliable then you might not be reliable either, the user may reason. You may even lose this customer to your competition. Both

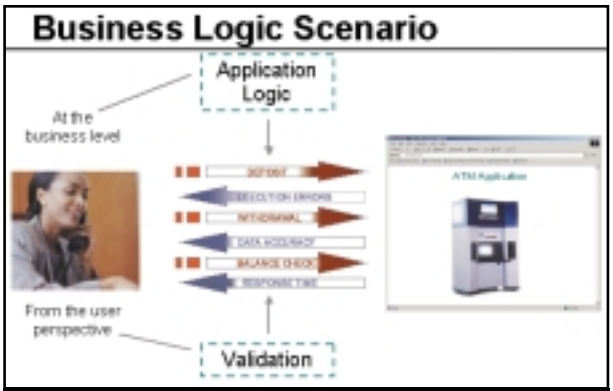



**FIG. 3: BUSINESS LOGIC EXAMPLE**

of these scenarios translate into lost revenue.

What does this mean from a management perspective? While managing WebSphere systems and the performance of individual application components both synthetically and in real time is essential, it doesn't give you the complete picture, as it doesn't tell you how well you're serving your end users. A management tool should also monitor the front end of your applications, both synthetically and in real time. For each monitored transaction you should check parameters such as response time and content (including patterns, size, and changes in the source code). Once you have response-time metrics of real-time/synthetic monitored transactions, and performance metrics across the different elements of your extended WebSphere infrastructure, you can feed these metrics into a common Web-based reporting tool that your management vendor should provide, to manage your service goals across the extended enterprise.

## Management for WebSphere

Computer Associates' solution for WebSphere environments, Unicenter Management for WebSphere, will help you maintain your critical Web applications and exceed user service levels by proactively managing the performance of your WebSphere servers and their underlying J2EE technologies, including the end user's experience and your application business logic. Unicenter Management for WebSphere is able to monitor performance from the top-level user experience down to individual Java components to identify the exact EJB, servlet, method, or external connection to back-end systems that is causing the problem, seamlessly integrating with the management of your total enterprise, to help you locate and automatically resolve problems in today's extended WebSphere Application Server infrastructure.

## Conclusion

Managing WebSphere infrastructures and world-class Web applications is a complex, difficult, and time-consuming job. I have discussed managing the WebSphere extended environments and the need to proactively monitor both WebSphere infrastructures and applications. I have discussed the philosophy of managing the containers and services, the contained (including the application business logic), and the end-user experience, and I have explained the need to monitor deployed applications both synthetically as well as in real time. It's important to keep all these factors in mind when selecting the appropriate WebSphere management tool to help ensure a positive end-user experience across your complex WebSphere environment.

CASE STUDY

# Global Holding & Investment Company


## WebSphere lays technical foundation for explosive potential

Global Holding & Investment Company is a retail consumer finance business founded 17 years ago in what was essentially a different technological era. President and CEO Joe Jones discusses how WebSphere has helped to transform his business from a paper-based company to a sleek automated firm poised for exponential growth over the next few years.

### The Company

GH&I specializes in the financing of consumer items, with a market niche in consumer goods typically priced below $4,000 and paid for in installment plans over a period of time ranging from 6–36 months. GH&I finances the purchase of household appliances, furniture, and other durable consumer goods that are bought with a small down payment and then paid for on a monthly basis.

"What we do is to step in to finance the transaction – we basically buy the paper or fund the paper to the merchant or retailer that's selling the merchandise so they can replace their inventory," says Jones. "We send a payment book out to the consumer and the consumer makes their monthly payments to us. Essentially we take out the middle man so the customer doesn't have to go to a credit card company or to a bank to borrow financing."

"Our subniche in the marketplace is actually buying credit risk paper – paper that's been priced at credit risk according to the previous credit history of each individual borrower," Jones says. "We buy paper – what we call loans – across the country. Over the past 17 years we've built a sizable portfolio with a well-diversified product base."

The average loan in the GH&I portfolio is about 24 months with a balance of about $2,000. The company services all of its loans in-house, including all of the collection work, loan-tracking and origination, and legal work. In addition to its consumer lending business, which serves 30 states, GH&I has a consumer retail mortgage department that handles first and second mortgages, home equity loans, and home improvement loans for homeowners in the New York and New Jersey areas. The third arm of the company is their collection department, which in addition to handling in-house loans, also takes on third-party collections for outside retailers that are having difficulty with their own collections.

### Technological Background

The technological end of the business has radically changed in the years since Jones founded GH&I. "When we opened Global originally, we used to draw a circle in the sand and we would put rocks in the middle to keep track of what loans we had or didn't have," he recalls.

"Essentially, everything was done on paper. Loan balances were kept on paper; account ledger sheets were all kept on paper – and tracking the loans was a monumental task, where piles of paper would move from one side of the desk to the other side of the desk and from one person's desk to another desk.

The purchase of the company's first computer was a major investment. "It was a 286 computer that was put out by some local company here and we thought it was going to automate our business and eliminate all of our paper and tracking problems," Jones says. "We were running Lotus, back in the days prior to Windows. We were running the DOS version of Lotus and everything was backslash file retrieve, and I remember we were excited because we had a half MB of RAM in the unit and we thought we were going to fly."

GH&I has increased its investment in technology quite substantially since then, adding a PC here and there and eventually networking all of its computers together. Today the company runs a series of networks in its office. One platform is running a Unix/Linux system and the other is running a Windows NT system supporting the entire staff. "Without the advancements in technology we could never be as efficient as we are today with respect to servicing our loans," Jones says. "We now have the ability to track loans live on our screen. Moment to moment, we're able to know exactly what our balances are, where we're going, how to adjust any type of collection needs that we have. Any type of action that needs to be done can be done live at this point."

## WebSphere's Role

Improved efficiency was the first noticeable impact of migrating to WebSphere, Jones says. "In the first obvious impact, it allowed us to tie applications together that were kind of scattered, so we were able to share data more efficiently. And sharing the data gives us the ability to analyze it from different perspectives, allowing us to make more efficient management decisions," Jones says.

WebSphere allows retailers to access GH&I's Web pages to transmit and access data in a secure live environment. "So if they have a consumer in the store, they can get to us live in a matter of seconds, have an approval, and documents print out right from their PC – instant approvals," Jones says.

"Workflow is the name of the game in our business, Jones says. "We receive literally hundreds and hundreds of applications a day from all across the country. When the application or the paperwork is sent to us, we have to kind of live with it from what we call 'womb to tomb.'"

When an application enters the office, it is entered into GH&I's main system and into a loan-tracking system that allows the company to give the merchants live access to the status of their loans. A merchant can come online at any given time and see the status of any of their loan applications.

"All of that is running on WebSphere," Jones says. "Once we've approved the loans we're then able to take all that data and move it to our loan servicing department, which runs on a Unix-based system that tracks all of the payments, due dates, the history of the account, the performance of the account, as well as storing crucial variables on each and every client so we can do certain types of regression analysis – looking to see who's paying, why certain activities are working better than others."

"The business logic is completely automated – we couldn't do it any other way," Jones says. "I don't believe that we can just sit down with automation and figure out mathematically who's going to pay a loan and who's not. What the system does for us is give us the data that we need in order to make intelligent strategic decisions," he adds. "Our business now, from the time a loan comes in until the time it's paid off, is all automated. Every step of the process is trackable and traceable. We can pull any account at any point in time based on any variable in the file and be able to compare it to other similar loans.

"The WebSphere product at this point has taken all of our small homegrown applications that we've developed over the years, that were kind of scattered in different programs, and has tied them all together into one manageable program so our programmers can now write everything in Java and they can move it from application to application and use is just about anywhere." GH&I is in the process of designing new systems with WebSphere products that are going to allow the company to integrate its different applications even more closely.

GH&I is investigating the possibility of adding a new WebSphere product to its arsenal – HATS (Host Access Transformation Server). "One of the big reasons we're looking at it is that in our collections system right now everybody works on the old Wyse terminals," Jones says. "We call them dummy screens. They're green screens, and we find that when we recruit new employees, they're not accustomed to it. Everybody has grown up in the Windows environment and we need to be able to move them into that environment. So they're going to go from the green screen to the GUI." Jones estimates that integrating HATS into GH&I's collection systems will allow the company to shorten the training process for a new employee by as much as 25–30%. "This obviously translates into a dollars-and-cents savings," he says.

GH&I plans to greatly expand its business over the next few years. "The industry is growing, but I think we are outpacing the industry," Jones says. "Our niche in the marketplace – and our brand name – is really established at this point. We are looking to increase our business by approximately tenfold over the next four years."

"We are presently streamlining our business for organizational efficiency and addressing the present capital structure to enable Global to attract new investors. The raising of new funds is going to allow us to increase the size of our portfolio substantially," Jones says. "Our challenge here is being able to manage that growth with the right type of technology. We're looking at WebSphere for products and innovation that are going to allow us to take the uniqueness that we've established here and build it out and expand out on it, to extrapolate it into a larger portfolio."

## Conclusion

"Without WebSphere's help over the past several years we probably would not be in the position we are in today – prepared for the growth that we're about to experience," Jones states. "Gathering new business is easy; managing the business when it comes through the door is a whole different ball game. And WebSphere has helped us manage the business once it comes through the door. After that it's a matter of maintaining it, and that's where the future growth comes from."

ABOUT THE AUTHOR

Patti Martin is cofounder of Simplex Knowledge Company, where she is vice president of Creative Services. She manages the company's Web servers and oversees Web content and creation. Patti received her education at the New School in New York City and has taken continuing education classes at NYU and the School of Visual Arts.

E-MAIL
pat@skc.com

*Evolving into a more agile development organization*

# Managing Change in WebSphere Studio Application Development

BY MELISSA BORZA

Web services initiatives are expected to change the very nature of computing and of application development. While Web services projects may not be fully realized today, many organizations are beginning to build the infrastructure to support this important development venture. According to analysts, WebSphere Studio will be one of the primary tools used to develop Web services.

ABOUT THE AUTHOR

Melissa Borza is the business manager for Computer Associates' AllFusion Change Management Suite. Specializing in application life cycle management, Melissa has written numerous articles and presented seminars and lectures throughout the world. Melissa is currently responsible for combining her strategic business expertise with her knowledge of technology to position and promote CA's end-to-end application life cycle management strategy.

E-MAIL

melissa.borza@ca.com

Web services is a set of standards used to create an open environment in which employees, partners, and customers of services can locate and work with service providers through Web-based heterogeneous systems. Web services also refers to the network-based applications that can interact automatically with other such applications.

Adding to the business drive to build Web services are intense competitive pressures and increasing market demands, which are forcing development organizations in the new economy to become more agile. It is critical that development teams be able to quickly create high-performance, flexible, content-rich Web applications while increasing efficiency and reducing development costs. In pursuit of these goals, developers must address a continuous flow of complex changes to application source code, documentation, test scripts, and other development components, with the ultimate goal of delivering top-quality, error-free, reliable, and scalable applications to meet business objectives.

As today's application development environments grow increasingly complex, the need for sophisticated, enterprise-caliber change and configuration management solutions intensifies. Specifically, in the WebSphere Studio and J2EE development realm, tying everyday development to change and configuration management activities – and to the process of deployment into operations – is the key to success. Maintaining control over the entire software development life cycle – from analysis to coding to testing to production – enables IT to ensure high-quality deliverables in the J2EE world, where extensive coding and testing is required to produce even simple applications.

Many WebSphere applications are composed of multiple traditional and Web applications. To manage this complexity, manufacturers recommend that WebSphere users have administrator, development, test, and production servers to help organize the development workflow (see Figure 1). Since J2EE applications are often built in an IDE in a PC environment and run in a variety of environments, the challenge is to build and assure the applications in the test production environments. This type of development requires a solution not only to manage that workflow, but also to automate the build process. This requires a comprehensive life cycle management tool.

Build management is often considered a small part of configuration management and is often overlooked by development teams. Building an application entails combining components to develop a complete executable program. This sounds rather simple, but Web services and many other J2EE applications may be composed of hundreds of individual components with many variables. Automating the build process is a key factor in successful application development and deployment.

A build management solution ensures a consistent, reliable, and repeatable process for application building. Build management tools use configuration information that indicates what goes into the application, along with specific rules that define how to integrate these components into a running application. For example, if a developer is creating a program to run on an HP Unix machine, it requires rules that define what C compiler to run, what options to use, and what machine to use for the build. Building the same application the same way every time improves application quality.

In an agile environment, changes to applications, Web content, documentation, and design specifications

are continually requested by end users as well as by analysts and QA resources. If these changes are not properly managed across platforms and throughout each stage of the application development life cycle, the organization will face costly production failures and system downtime.

As with other IT initiatives, process control and team collaboration are critical to the success of Web services development. Organizations can significantly improve productivity through managed collaboration, providing they have the technology needed to keep everyone – including business analysts, developers, testers, and end users – synchronized with one another. Organizations need to define and enforce software development processes in the form of a development life cycle to tie IT development to IT operations and back to line-of-business requirements, to reduce risk and to improve the time-to-market associated with developing and deploying new systems.

Most Web services projects are supported by diverse teams in which multiple developers work on the same source code at the same time in order to meet the heavy demands placed upon their organization to quickly deliver these new applications. To successfully implement this kind of concurrent development, users must automate the process of merging multiple versions of source code and identifying conflicts for resolution (see Figure 2). Without an integrated change and configuration management solution, IT cannot manage this kind of collaboration.

The drive to accelerate time-to-market while maintaining control over the application development life cycle requires the seamless integration of world-class change management capabilities and effective development processes. To ensure optimal organizational efficiency and application quality while attaining maximum business value from change management tools, organizations must ensure that they can support all of their strategic development environments. Integrating a complete change and configuration management solution into the development environment provides centralized control and helps ensure that your systems are up and running and meeting the service levels defined by the business. Such integration ensures that mission-critical systems are developed quickly and maintained according to exact standards.

Organizations need a single, enterprise-wide change management solution that will enable them to track software changes and manage the application development process – regardless of the platforms or environments these applications are developed on or deployed to. Such a solution should have an

easy-to-use, intuitive interface and a multitier architecture to synchronize development teams anywhere in the enterprise. It should offer simple point-and-click and drag-and-drop operations to enable developers to meet demanding delivery schedules and to realize maximized productivity, reduced development costs, and improved application quality.

Through the Eclipse program (www.eclipse.org) software vendors are working together to create an open environment that allows the open source community to create additional plug-ins for WebSphere Studio to streamline the development process and ensure successful projects. Several change management vendors have taken advantage of this standard to provide WebSphere Studio developers with seamless access to their software change and configuration management and life cycle management solutions.

Life cycle management solutions provide the ability to manage, control, and improve the process of building, changing, and operating WebSphere applications, with the primary objective of reducing and managing risk.

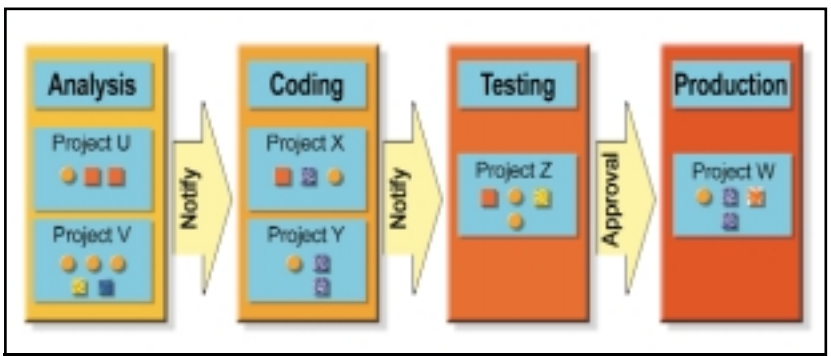



**FIG. 1: MULTIPLE SERVERS HELP MANAGE THE DEVELOPMENT WORKFLOW OF BUILDING AND DEPLOYING WEB SERVICES AND J2EE APPLICATIONS.**

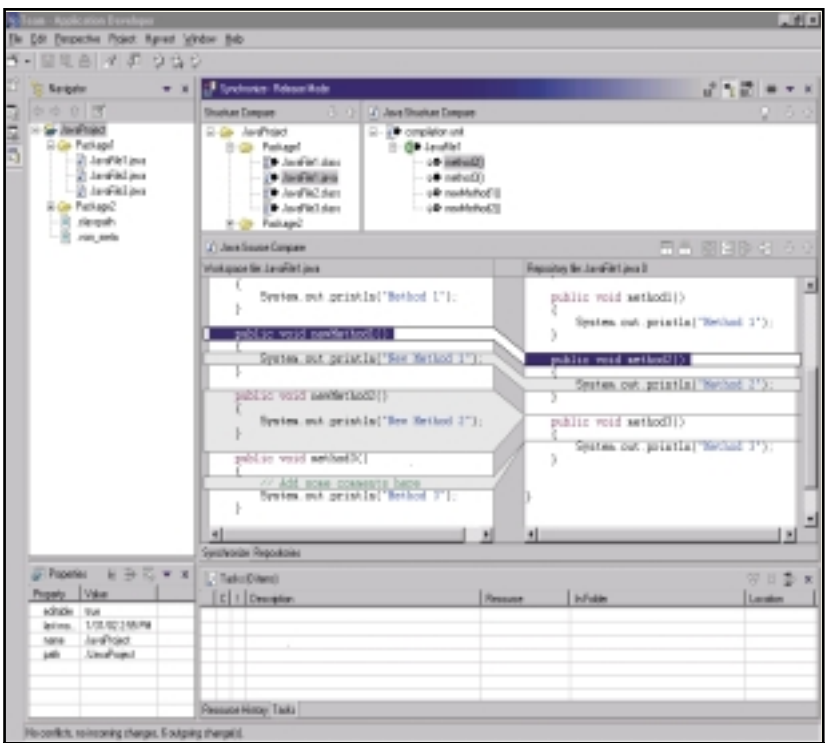


**FIG. 2: AN INTERACTIVE MERGE FACILITY ALLOWS A TEAM OF USERS TO DECIDE WHICH VERSION OF A CONFLICTING LINE TO INCLUDE IN THE FINAL VERSION OF THE APPLICATION.**

Because Web services projects are so diverse – ranging from Web enablement to high-powered, cross-platform applications written with multiple development tools and in several languages by teams transcending geographic boundaries – the disciplines surrounding this process are rapidly evolving. Specifically, software change and configuration management is expanding beyond traditional application development. These solutions provide a management focal point for integration initiatives that traditionally focus on system management and automation tools such as alert management, software distribution, and service desk management.

Even though new development is taking off in so many directions, application development managers must maintain service levels. Everyone on the team has to do more with less – less time, less money, and fewer resources. They must exceed their previous best efforts and constantly develop better applications. In addition, many organizations have quality efforts and continual improvement projects in place. Whether organizations are trying to implement the Software Engineering Institute's Capability Maturity Model or the ISO 9002 standard, enterprise-wide change management must take center stage.

To increase the speed and effectiveness of delivering Web services and J2EE applications to the market, WebSphere developers must employ powerful life cycle management solutions that are integrated with the IDE and provide much more than simple workgroup-oriented software change management. Only then will organizations be empowered to increase the speed of development, ensure software accuracy and team effectiveness, reduce deployment risk, lower training costs, and truly evolve into an agile organization.

# 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


**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

Boston call for papers opens November 11, 2002.

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

# SUBSCRIBE TODAY TO MULTI

## Go To www.sys-con.com/suboffer.cfm


## and receive your **FREE** CD Gift Package VIA Priority Mail

Each CD is an invaluable developer resource packed with important articles and useful source code!


More than 1,400 Web services and Java articles on one CD! Edited by well-known editors-in-chief Sean Rhody and Alan Williamson, these articles are organized into more than 50 chapters on UDDI, distributed computing, e-business, applets, SOAP, and many other topics. Plus, editorials, interviews, tips and techniques!

**LIST PRICE $198**


The most complete library of exclusive *JDJ* articles compiled on one CD! Assembled by *JDJ* Editor-in-Chief Alan Williamson, more than 1,400 exclusive articles are organized into over 50 chapters, including fundamentals, applets, advanced Java topics, Swing, security, wireless Java,... and much more!

**LIST PRICE $198**


The most up-to-date collection of exclusive *WSDJ* articles! More than 200 articles offer insights into all areas of WebSphere, including Portal, components, integration, tools, hardware, management, sites, wireless programming, best practices, migration...

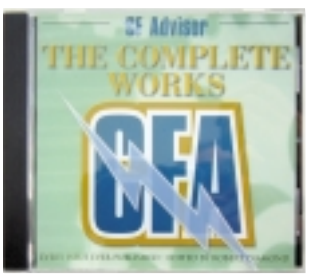
**LIST PRICE $198**


The most complete library of exclusive *CFDJ* articles on one CD! This CD, edited by *CFDJ* editor-in-chief Robert Diamond, is organized into more than 30 chapters with more than 400 exclusive articles on CF applications, custom tags, database, e-commerce, Spectra, enterprise CF, error handling, WDDX... and more!

**LIST PRICE $198**


The largest and most complete library of exclusive *XML-Journal* articles compiled on one CD! Edited by well-known editors-in-chief Ajit Sagar and John Evdemon, these articles are organized into more than 30 chapters containing more than 1,150 articles on Java & XML, XML & XSLT, <e-BizML>, data transition... and more!

**LIST PRICE $198**




The complete works of *CF Advisor* are now on one CD! Check out over 200 exclusive articles on topics such as e-commerce, custom tags, Fusebox, databases, object-oriented CF, WDS, upgrading CF, wireless, Verity, and more. Plus, find interviews, editorials, and source code!

**LIST PRICE $198**

## Pick the CDs to go with your Multi-Pack order

- Pick one CD with your 3-Pack order
- Pick two CDs with your 6-Pack order
- Pick three CDs with your 9-Pack order

- ☐ Web Services Resource CD
- ☐ Java Resource CD
- ☐ ColdFusion Resource CD
- ☐ XML Resource CD
- ☐ WebSphere Resource CD
- ☐ CF Advisor Complete Works CD

**Your order will be processed the same day!**

WebSphere DEVELOPER'S JOURNAL
PRESENTS...

# SHARE

## *Volunteer organization offers user-focused education, professional networking, and industry influence*

In an ongoing effort to provide access to the most useful resources for our readers, ***WebSphere Developer's Journal*** presents SHARE – the original IBM user group – a nonprofit organization founded in 1955 whose member organizations use IBM enterprise-class system technologies. An extensive network of more than 400 volunteers drives its visions and initiatives.

SHARE provides the valuable resources today's IT professionals need to successfully manage IT infrastructures. With its large and diverse member base, comprehensive educational forums, networking opportunities, industry influence, and wide range of communications tools, SHARE gives its members a solid foundation for industry, personal, and professional success.

### Education

SHARE is renowned for its educational excellence delivered through a variety of channels. Semiannual user conferences, real-time Webcasts, downloadable conference presentations, and online resource and code repositories combine to distinguish SHARE from other user groups and training providers. SHARE's programming is guaranteed timely and relevant because much of the education is delivered by front-line IT practitioners.

### Networking

SHARE counts more than 2,200 organizations among its active membership, including a majority of the Fortune 500, leading colleges and universities, independent consultancies, and federal and municipal government agencies.

SHARE facilitates peer networking through several online tools in addition to providing opportunities during its semiannual user conferences.

### Industry Influence

SHARE influences the IT industry through an annual survey of its membership, along with the organization's online enhancements application, SHARE Requirements 2.0. Requirements 2.0 enables SHARE members to submit, discuss, and vote on requests that are then sent to respective IBM product development teams for consideration and remarks, thereby having a direct impact on product evolution.

### SHARE in Dallas

SHARE's next event, SHARE in Dallas, takes place February 23–28, 2003, in Dallas, Texas. No other technical conference, symposium, or seminar offers SHARE's breadth of programming, volume of user experiences, tremendous return on training investment, and participants' willingness to share in technical problem solving. Complementing the robust technical program are an expansive trade show, ample opportunity to network with peers, hands-on labs, unique access to influential developers from IBM as well as other leading software providers, plus an applications development special program event.

### Applications Development Special Program Event

This special program event will run the length of the weeklong conference with an agenda that revolves around saving – saving budget, saving time, saving headaches. These sessions will include the latest development news, along with direct-from-the-trenches user experiences. Four special application program tracks highlight savings for managers, Web developers, traditional developers, and software change management specialists:

- ***Save a Million Management Misunderstandings:*** Focusing on J2EE, .NET, XML, SOAP, Unicode, and other current topics. Recommended for applications management, architects, and strategists.
- ***Save a Million on Web Development:*** Focusing on learning and using the new tools and languages for Web-enabled applications. Recommended for applications developers, project managers, and architects.
- ***Save a Million on Traditional Development and Support:*** Focusing on application analysis, debugging, testing, migration, renovation, and modernization. Recommended for applications developers, tools support, and management.
- ***Save a Million on Software Change Management:*** Focusing on new solutions for cross-platform software change management (SCM) and deployment. Recommended for applications software

change management specialists, managers, and applications staff.

## WebSphere Offerings at SHARE

WebSphere-focused sessions at SHARE in Dallas begin on Sunday, February 23, with a seminar, "Developing JMS Applications Using WebSphere Application Server and WebSphere MQ," for an additional participation fee. This seminar, presented by Gary Ward of Information Design, Inc., and Andrew Wilkinson of IBM Hursley Lab, will cover the development of JMS applications for WebSphere MQ and WebSphere Application Server using WebSphere Studio Application Developer.

The "WebSphere Opening and Keynote" session will be led by Al Hechler, program director, WebSphere for z/OS and OS/390. This presentation will give an overview of the WebSphere family, rollout for future function, and a discussion of customers who have used WebSphere for z/OS successfully.

A sampling of other WebSphere sessions to be presented at SHARE in Dallas include:

- WebSphere Studio Application Developer and CICS
- XML in a Production CICS Environment
- Understanding the WebSphere MQ Object-Oriented Model for COBOL Programmers
- WebSphere MQ with Java and JMS
- Introduction to the XML and MQSeries Extenders for IBM's DB2
- WebSphere V4, EJB Security Options & Config
- WAS Performance and Tuning – What an Experienced Mainframer Wants to See
- WebSphere for Dummies
- Deploying Web Applications in WebSphere V4 for z/OS and OS/390 Hands-On Lab
- Java and XML: Servlets and JSPs
- Developing Web Services with WebSphere Studio Application Developer
- Web Services Advanced Topics: Beyond SOAP, WSDL, and UDDI
- High Volume e-Business, Designing for Success
- Java Lab: WebSphere Application Developer for Java
- ISPW: Enterprise Software Change Management for WebSphere
- WSED: WebSphere Studio Enterprise Developer
- WebSphere Studio Asset Analyzer – Web Enabling Core Business Applications
- Java: User Experience – Managing High-Performance Java Applications in Large-Scale WebSphere Environments

For more information on SHARE in Dallas, including registration and the technical program, visit www.share.org/dallas.

*From Trilog*

# FlowBuilder XML Edition: XML Super Glue

REVIEWED BY JAY JOHNSON

About five years ago, I worked with a pure XML-embedded database. The advantage of such a database is unmatched flexibility. New definitions of everything can be added or changed on the fly. It was lightweight and useful in tight spots, but of course lacked many of the features of more rigid relational databases. One advantage it had was that it was compatible with everything. At the time it was obvious that an XML-based component integration strategy could be just the thing to quickly glue a system together.

ABOUT THE AUTHOR

Jay Johnson is an independent J2EE architect.

E-MAIL

aguy4ejbs@yahoo.com

Nowadays enterprise server-side technologies abound, but between the J2EE, EJB, Web services, .NET, Domino, zOS, CORBA, etc., paradigms, no grand unified solution has emerged. The EJB paradigm, for example, solves a number of problems common to enterprise applications, but it's a compromise that is built to best fit distributed enterprise applications. It is by no means flexible enough to satisfy all possible customer requirements, and its complexity requires an uncommon level of Java expertise. Often, the challenge is to quickly build Web applications that can be deployed in any IT environment, while leveraging existing developer skills.

It should, therefore, come as no surprise that a commercially available, XML-centric development environment has emerged that capitalizes on XML as the greatest common factor among all existing server paradigms.

Trilog designed its Visual XSP Studio as a means to develop sophisticated Java applications in an intuitive manner that suits developers with HTML, XML, and JavaScript skills. Experienced Notes and Domino developers will find that FlowBuilder follows much the same paradigm as the Lotus environment.

Welcome to a new view of server technology. Instead of EJBs, or in addition to them, FlowBuilder provides XML adapters that act as EJBs. This tool can represent the hierarchy of a database easily in XML or create XML from a schema of related tables, and HTML from the SQL or XML (see Figure 1).

FlowBuilder XML Edition provides a robust workflow engine with intuitive drag-and-drop design tools for modeling processes and defining workflows. The workflow component includes a unique utility called the Organizational Planner for defining the organizational structure, including groups, roles, and skills for individual users. This data can be included directly in a workflow definition, allowing the application to dynamically determine the best recipient among the available resources in the organization.

Overall, Trilog's FlowBuilder XML Edition provides a robust set of functionality across a diverse range of business needs, delivering value on a number of levels while also complementing existing systems and skills that organizations may already have in place. Thus, FlowBuilder XML Edition is at the intersection of standards-based application development, process automation, and integration. Other vendors provide independent tools to handle these functional areas, but none are offering them in a package designed for Lotus shops.

FlowBuilder's XML Server Page (XSP) technology is similar to JSP and ASP and provides full separation of the business logic layer and the presentation layer – allowing developers to independently change the HTML presentation logic without impacting the business logic. The product's Visual XSP Studio does not use proprietary tags within HTML layouts, and its visual environment for adding components (XSP Beans) to HTML layouts can make the development process more intuitive than other alternatives.

Since XML is the greatest common factor in many of the systems in an enterprise, FlowBuilder XML Edition is compatible with any IT environment. FlowBuilder uses standards-based XML to ensure openness. A solution created using FlowBuilder stores all its definitions

and data in XML, and uses XML for all information exchange. The solution can integrate with other systems and use their data within its applications, leveraging existing IT investments.

Obviously, FlowBuilder XML Edition aligns well with emerging Web services strategies. The system's components are natively built as Web services that use true SOAP messaging to communicate internally, with the FlowBuilder server acting as the Web services container.

Using FlowBuilder, users can define databases and/or Lotus Notes (Domino server) as resources, and can auto-convert Domino applications (Lotus Script, forms, and all design elements) into Java, XML, or HTML, and create an XML database for the application. Flowbuilder also enables non-Domino applications to connect to a Domino resource. Flowbuilder provides one of the best (if not the only) migration paths from Lotus Notes to J2EE.

## Product Components

The major functional components of FlowBuilder XML Edition are:

- ***Visual XSP Studio:*** Client application for building Web applications based on XSPs. This Web development environment allows developers to create applications that can manage and present XML content as part of a defined business process; all application metadata and objects are stored as XML.

- ***Visual Integrator:*** A generic XML adapter that allows developers to access and merge data from multiple third-party systems for use in Web applications and workflows. The Visual Integrator converts this external data into standard XML on the fly, and automatically constructs the schema – without requiring developers to build connectors, schemas, or data definitions.

- ***FlowBuilder client interface:*** Java-based browser interface that interacts with all of FlowBuilder's servers (BizDAP Server, Design Server, XSP Server, and Workflow Server).

- ***Visual Designer:*** Browser-based workflow design utility that provides an intuitive interface for creating process definitions in a drag-and-drop fashion. The Visual Designer runs against FlowBuilder's Process Server, and stores all process definitions and application components as XML in a relational database.

- ***Visual Organization Planner:*** Browser-based administration utility that allows organizations to define internal and external organizations (e.g., divisions, departments, suppliers, customers, etc.), the hierarchy of users (e.g., VP, manager, employee), and roles and subroles. The Organizational Planner also defines the relationships between all of these elements. All definitions are stored in FlowBuilder's BizDAP Server, which can also include information from LDAP directories. At runtime, FlowBuilder's Matrix Organization Search Engine (MORSE) uses the Organization Planner database to evaluate workflow recipients based on role assignment and availability, making dynamic determinations of the best recipients.

- ***Workflow Monitor:*** Server that tracks all system actions and audit trail data, giving administrators access to detailed statistics and other administrative features.

I found it easy to use FlowBuilder to rapidly develop a simple database application that mapped to several DB2 tables. FlowBuilder autogenerated a Web-enabled user interface, including the code necessary to access the database from the UI. The tool also generates field validation on the form, and since everything in FlowBuilder is XML, I could add new fields on the fly directly into XML from the form definition.

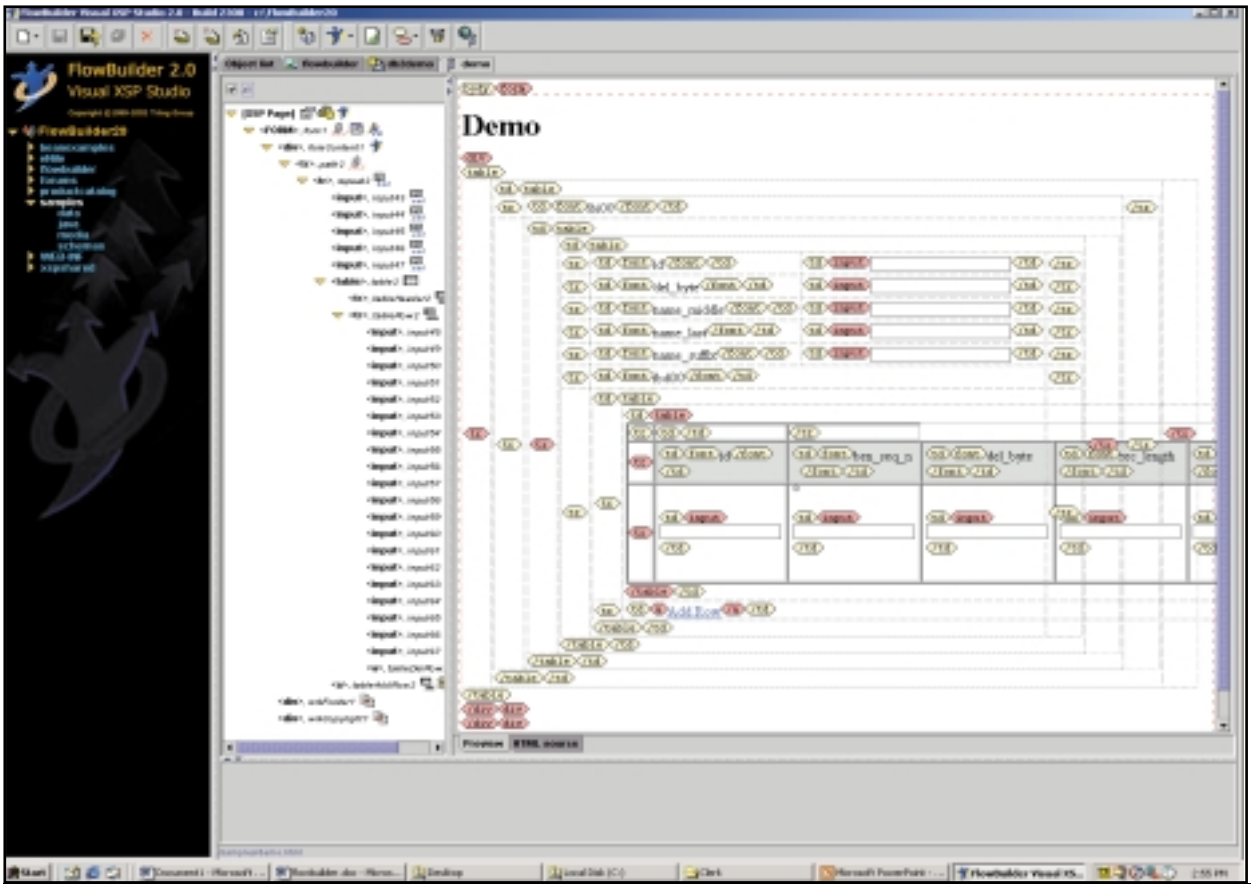



**FIG. 1: FLOWBUILDER VISUAL XSP STUDIO**

One of the neat features of the system is that user privileges to view and edit Web pages are automatically implemented in the application based on the roles defined using the Visual Organization Planner.

The autogenerated Web pages have all the plumbing to connect with the database, but of course must be completed with an HTML editor to make them aesthetically pleasing.

In the world of component-oriented programming, FlowBuilder is a pure Java tool that can add value to a project in a variety of ways. It fits neatly in between IDEs, components, and databases, and produces components that execute on almost any server. While it works as a sort of universal component glue, it also provides the ability to split the page layout from any processing on the server.

The presentation layer contains only presentation logic, and the presentation logic is loosely coupled with the display since it uses XML tag IDs, not taglibs embedded in HTML.This loose coupling means each tag is disassociated from the logic, allowing the logic to be coupled in later. This allows developers to generate a layout, which can be interpreted in any way needed depending on the client type.

Unlike most relational databases, FlowBuilder's XML database can include a schema in another schema, or one schema can be included in many schemas. In addition, the tool supports component polymorphism by applying different transforms to same tag. This goes way beyond JSPs and ASPs!

Three major enhancements are planned for future releases.

- The ability to build your own XSP Beans without writing Java.
- Visual SOAP – which will provide the ability to visually build Web services by putting beans on XML tags as you build an XSP page.
- Copy/Paste text does not work when Visual Process Designer is run as an Applet in a browser, but this will be fixed in the next release.

### Trilog Group Inc.

**COMPANY INFO:**
54 Cummings Park
Suite 308
Woburn, MA 01801
E-mail: sales@trilog group.com
Web: www.triloggroup.com
Telephone: 800-818-2199

**SUPPORTED PLATFORMS:**
Windows NT/2000/XP, Linux, OS/400, OS/390, Solaris, AIX, HP-UX, and MacOS X

**MINIMUM SYSTEM REQUIREMENTS:**
Pentium III 500 MHz,
128 MB RAM,
300 MB disk space

**PRICING:**
Visual XSP Studio is $5,000 per developer. The FlowBuilderWorkflow engine is priced independently based on server configuration. For additional information, call 800 818-2199.

# One Size Doesn't Fit All

BY KEN GREENLEE

Choosing a technology vendor today may seem like a game of chance. With many "dot-coms" turned into "dot-bombs," laid-off employees have turned to consulting as a way to earn a living. Now, I do not begrudge anyone earning a living. I consider myself to be a consummate capitalist. But as the number of vendors claiming to be Java-WebSphere specialists has multiplied, sorting through various proposals may feel like spinning the wheel of some game of chance. It should not. So, what should IT managers look for when hiring a consultant or training provider? Apply this simple test to determine if your prospect is a good choice.

## Qualifications

Anyone setting foot in your office should have the proper qualifications. Like passing a bar exam, industry certifications confirm knowledge and proficiency. For the Java platform, look for recent Sun Java certification; for J2EE application development, insist on WebSphere Studio Application Server and OOAD (object-oriented analysis and design) certifications. It also wouldn't hurt to have a WebSphere administration certification thrown in for good measure. The jCert Initiative, Inc., (found at www.jcert.org) can help you sort out the various technical certifications. Dedicated consultants usually have multiple certifications and are happy to provide a detailed resume.

Which brings me to the next point: experience. Require that a vendor already have the experience needed to make your project a success. Ask for a list of references, and be sure to do a little detective work. Spending 15 minutes on the phone may save thousands of dollars in project expenses. Granted, the vendor won't give you the name of someone who won't give a good reference, but at least it is something to check.

What about working outside the box? In other words, what will the vendor do if one of your employees asks an off-topic question? Does the vendor have the experience and acumen to answer the question, or is the vendor merely a talking-head newscaster who knows how to pronounce technical jargon? Clearly, you will want a vendor qualified to work outside the box while keeping the project on target.

## Flexibility

If you are hiring a training vendor, how willing is the vendor to make changes to the curriculum? No two snowflakes are alike, and neither are two projects. Training vendors should be willing to take existing courses, add new content when necessary, and create a unique syllabus.

What about scheduling and location? What about cancellations? A vendor should realize that your existing business takes precedence over any new training. Choose a vendor that is willing to make scheduling changes – even the week before a scheduled class.

We all know that one size doesn't fit all, and this is where you may eliminate many vendors.

## Focus

What are the vendor's true motives? Is the vendor using the engagement to sell some other unrelated product or service? For example, when training, does the vendor use class time to sell hardware? Time is too valuable to be wasted on commercials. The vendor should focus on your project. Any additional marketing should be done outside of class. If the vendor attempts to market products during class, insist that it be done during lunch. Oh, and ask for free pizza!

## Value

Low cost is not necessarily value. Of course, competitive pricing will help narrow the field, but would you feel comfortable buying a piece of life-saving equipment produced by the lowest bidder? Value is really about the intangibles that make choosing a vendor a success. Expect a guarantee. What happens if you are not satisfied with the vendor, for any reason? Do you get either a partial or full refund? Choose a vendor that offers a guarantee with the fewest exclusions. Ideally, the guarantee would be as simple as "if you are not satisfied, you do not pay."

If you are purchasing training, how are the courses conducted? Does the vendor minimize breaks and lunch to give you a full day of training? Does each student receive a professional-looking kit of materials? What happens if students have questions after the vendor has left?

These are the intangibles that can greatly increase a vendor's value, making your choice much easier.

So how do you apply this test? Probably, the best approach is to rank each of these by relative importance. Create the ranking before beginning your vendor search. If a potential vendor does not meet your criteria, ask for a change or move on to the next vendor. With a little perseverance, you will succeed in your vendor search, leaving the games of chance to Las Vegas.

ABOUT THE AUTHOR... Ken Greenlee is chief technology officer of Kenetiks, Inc., a worldwide provider of WebSphere training and mentoring. Ken has 14 years of professional software development experience, holds a B.S. in computer science from Indiana University, and has several WebSphere and Java certifications.

E-MAIL
kengreenlee@hotmail.com