SYS-CON

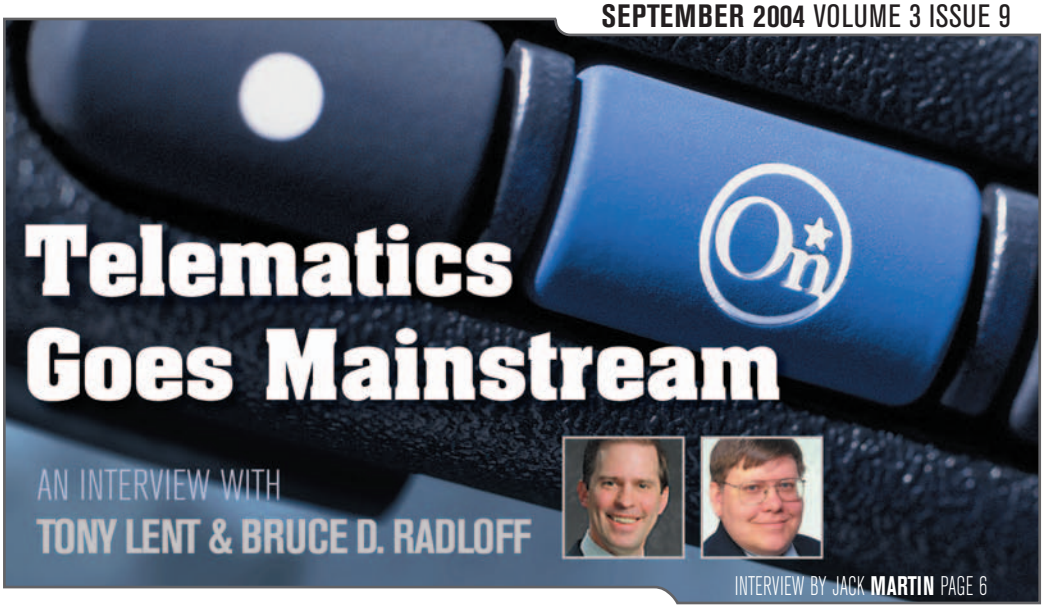
PERFORMANCE TESTING • MIGRATION • STRUTS

WebSphere® JOURNAL

www.SYS-CON.com/WebSphere

The World's Leading Independent WebSphere Resource

SEPTEMBER 2004 VOLUME 3 ISSUE 9



DISPLAY UNTIL NOVEMBER 30, 2004

$8.99US $9.99CAN

09

0 09281 03422 3

What//: are your Web applications saying to your customers?

diagnosys

Have you survived a Web App from Hell? Tell us how you lived through the delays, confusion, and pain, and H&W will put your name in the hat for an HP iPAQ H5500 Pocket PC.

Share your story today at:
www.hwcs.com/wsj14.html

You might as well be closed for business if your Web applications aren't running at top speed 24 hours a day. With poor Web performance costing businesses $25 billion a year, you need to find problems, and you need to find them fast.

You need DiagnoSys.

DiagnoSys is the first intelligent performance management solution that analyzes your J2EE and .NET applications from end to end. DiagnoSys starts analyzing quickly, gathers data directly from all application resources, and proactively finds and helps fix the real cause of problems – before they cost your business big time.

So when it's absolutely essential that your Web applications say "Open 24 hours," trust DiagnoSys.

H&W
www.hwcs.com | 1.800.338.6692

Bridging the Enterprise Computing Gap For 25 Years

You May Have Noticed

BY JACK MARTIN

As I'm sure you've noticed, over the last few months I've been asking you whether you think BEA is DOA. The response has been intense and interesting. At some points some readers started with personal attacks on me and, in some cases, the publisher of this magazine. I think those people have watched way too much cable television news over the last couple of years.

It was textbook, biased opinion based on no facts and seemed in most cases to be people with obvious second agendas. My goal was not to pick on BEA or say BEA had bad technology; as a matter of fact, many WebSphere users seem to respect and appreciate what BEA has to offer from a technology perspective.

What I was trying to find out was why BEA, which was first to market and way ahead of everyone for a long time, lost its lead to IBM WebSphere and continues to fall behind. Obviously, all is not well over at BEA. After their market share started to drop, many of their key employees left BEA for other jobs in the industry. BEA is now also suffering from a massive brain drain.

Over the summer BEA's top technology guys, Adam Bosworth and Scott Dietzen, quit, as did Scott Edgington, vice president of partner relations. BEA's top marketing guys, Rick Jackson, vice president marketing, and Erik Frieberg, vice director product marketing, also pulled their own plugs.

I have spoken with many system integrators and software developers in the past few months and asked each one why they thought BEA was in decline. Many of them shared a similar view: that while BEA has great technology they do not have the business savvy to grow and compete with IBM on all of the fronts they have to just to stay even no less regain the lead.

As far back as May 31, 2002, Joanne Correia, a Gartner Group analyst, went on the record, stating, "They're in the rearview mirror," she said. "It's no longer a technology game, it's a brand-marketing game, and IBM's the king of that."

The Gartner Group recently told their clients to stop deploying any new BEA projects until things became clearer or we know where things are going. These are not good signs for customers who have purchased BEA's products in the past, or people who recommended that their companies go with BEA.

I recently spoke with Jim Brett and Paul Valenti of Cartesis and asked them why they went with WebSphere. They told me, "There are several reasons why Cartesis chose WebSphere as the backbone of the Web component of their BPM solution, Cartesis Magnitude. First, Cartesis seeks to integrate best-of-breed components into its solutions and our market research overwhelmingly pointed to IBM's WebSphere. Also, a multitude of our clients desire to run their application on a respected and proven product like WebSphere. Additionally, we also have an active partnership with IBM, which enabled us to integrate this proven servlet engine into our solutions with the developmental support provided by IBM. In short, Cartesis continuously seeks to architect and develop its solutions with proven vendors, components, and partners; utilizing IBM WebSphere was a no brainer and has shown to be a profitable decision."

The answer I got from Jim and Paul is the same one I got over and over again from everyone I've spoken with. Everyone says it a little differently but it boils down to the same basic message: great technology, the customers like it, and a strong partnership with IBM.

With all of the time we have spent on BEA in the last few months you would have thought that BEA would have called me back by now but they haven't. If they ever do call I'm going to ask them if they know any of the people who have written to the Web forum of this magazine; in my opinion, some of them watch way too much cable news.

Jack Martin, editor-in-chief of ***WebSphere 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. Jack is coauthor of ***Understanding WebSphere***, from Prentice Hall. jack@sys-con.com

WebSphere® JOURNAL

ADVISORY BOARD
Richard Arone, David Caddis, Mike Curwen, Devi Gupta, Lloyd Hagemo, Barbara Johnson, Shannon Lynd, James Martin, Doug Stephen, Victor Valle

EDITOR-IN-CHIEF JACK MARTIN
EXECUTIVE EDITOR GAIL SCHULTZ
CONTRIBUTING EDITOR PATTI MARTIN
PRODUCT REVIEW EDITOR JAY JOHNSON
SPECIAL PROJECTS EDITOR RICHARD ARONE
EDITOR NANCY VALENTINE
ASSOCIATE EDITORS JAMIE MATUSOW

WRITERS IN THIS ISSUE
Roland Barcia, Sergei Baronov, Louis Cirne, Jonathan Forck, David Lapp, Joseph R. Marques, Jack Martin, Tilak Mitra, Michael Pallos, Asim Saddal

SUBSCRIPTIONS
For subscriptions and requests for bulk orders, please send your letters to Subscription Department.
888-303-5282
201-802-3012
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 KIRCAALI
PRESIDENT, SYS-CON EVENTS GRISHA DAVIDA
GROUP PUBLISHER JEREMY GEELAN
SENIOR VP, SALES & MARKETING CARMEN GONZALEZ
VP, INFORMATION SYSTEMS ROBERT DIAMOND
VP, SALES & MARKETING MILES SILVERMAN
PRODUCTION CONSULTANT JIM MORGAN
FINANCIAL ANALYST JOAN LAROSE
ACCOUNTS PAYABLE BETTY WHITE
ADVERTISING DIRECTOR ROBYN FORMA
ADVERTISING MANAGER MEGAN MUSSA
ASSOCIATE SALES MANAGERS KRISTIN KUHNLE, BETH JONES, DOROTHY GIL
LEAD DESIGNER LOUIS F. CUFFARI
ART DIRECTOR ALEX BOTERO
ASSOCIATE ART DIRECTORS RICHARD SILVERBERG, TAMI BEATTY, ANDREA BODEN
WEB DESIGNERS STEPHEN KILMURRAY, MATTHEW POLLOTTA
CIRCULATION SERVICE COORDINATORS EDNA EARLE RUSSELL, LINDA LIPTON

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® Journal (ISSN# 1535-6914) is published monthly (12 times a year).
Postmaster send address changes to:
WebSphere Journal, SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645

WORLDWIDE NEWSSTAND DISTRIBUTION
CURTIS CIRCULATION COMPANY, NEW MILFORD, NJ

FOR LIST RENTAL INFORMATION:
KEVIN COLLOPY: 845 731-2684, KEVIN.COLLOPY@EDITHROMAN.COM
FRANK CIPOLLA: 845 731-3832, FRANK.CIPOLLA@EPOSTDIRECT.COM

Why is it that some Java guys are more relaxed than others?

These days, with everything from customer service to sales and distribution running on Java, keeping your enterprise applications available can be pretty stressful.

Unless of course, you've discovered the power of Wily. No other software offers our level of insight. Which means you'll be able to monitor transactions and collaborate with colleagues in real-time. Even more important, you'll be able to optimize performance across the entire application—end-to-end.

So put Wily to work. And keep performance problems from pushing you over the edge.

Get Wily.™

1 888 GET WILY | wilytech.com

wily technology

Can Your Car Talk?

And do you know what it's saying?

Jack Martin, editor-in-chief of ***WebSphere Journal***, recently sat down with Bruce Radloff, CTO of OnStar; and Tony Lent, vice president of OnStar's Wireless Strategic Business Unit, to talk about how they use WebSphere.

TONY LENT

BRUCE D. RADLOFF

WebSphere Journal: Bruce, what does the chief technology officer of OnStar do?
Bruce Radloff: I manage the advanced systems development group here at OnStar.

WJ: What does the advance systems development group focus on?
BR: It maintains an end-to-end overview of what OnStar is all about, with a focus on new product innovation. That overview includes three primary components: the vehicles themselves, and the system integration that goes along with them; the network that connects the vehicles to our back office; and the back office, which comprises several different platforms, including our call centers, Web interaction and support, and OnStar Virtual Advisor, which is how we provide voice-activated, Web-based automatic services to the driver.

WJ: Give me a primer on what OnStar is?
BR: That's easy. OnStar is the industry-leading provider of safety and security in-vehicle services. We use GPS technology and an embedded cell phone to link the vehicle and driver to a 24-hour call center where operators, whom we call advisors, are available to respond to all kinds of service requests, from emergencies to stolen vehicle locations to simple routing support to remote door unlocks.

WJ: OnStar services include automatic airbag notification. How does that work?
BR: Going back to our early days, airbag notification has been a cornerstone of our safety and security offerings. In the event the airbags go off, there's a trigger in the electronics within the vehicle that automatically generates an outbound call to the 24-hour OnStar call center. That call transmits vehicle location information, then toggles over to a voice call so the advisor can speak to the occupants in the vehicle.

WJ: How many subscribers do you have?
BR: Right now we're sitting at about 2.7 million moving towards 3 million by the end of 2004.

WJ: Recently, someone told me that OnStar has been instrumental in recovering a lot of cars that have been stolen. Can you talk a little bit about that?
BR: Right. In the event your vehicle is stolen, you can report it to the police or you can call OnStar and we'll help you report it to local police authorities. Once we have the police involved and get their approval, we can locate the vehicle – from a data perspective, we call into the vehicle to retrieve latitude/longitude information – and share that information with the police so they can try and recover the stolen vehicle.

WJ: It kind of takes the fun out of stealing cars!
BR: That's the idea.

WJ: Another thing I find very interesting is being able to ask OnStar how a car or truck is running.
BR: That's our GM Goodwrench remote diagnostics service. I think, as we move forward, our abilities to help our customers better manage their vehicle by understanding the data that the vehicle produces and translating that into actionable additions for our subscribers is going to be huge. All of us at one point or another have probably experienced the "check engine" light coming on. What OnStar can do, at the customer's request, is actually go and do what we call a diagnostic probe and pull out the diagnostic trouble code and then give the customer some advice in terms of what actions can be taken. It goes back to the whole peace of mind issue with our subscribers. When the light comes on, you can hit the OnStar button and we can tell you a range of options: pull off to the side of the road and help is on its way; or that there doesn't appear to be a big issue and just make sure you have a

dealer look at it the next time you're in for service. And in some instances, the issue will resolve itself with an ignition recycling, so you basically don't have to do anything.

I think it's a great way for us to drive some connection and build that bridge back to the dealership. Obviously, the GM Goodwrench brand is a big one for us and I think we believe that GM dealer service people know our vehicles better than anybody else so they can solve the problem quickly and efficiently.

WJ: I recently experienced the OnStar Virtual Advisor firsthand while test-driving a 2004 Silverado and it was incredible. Could you explain to me exactly how that worked?

BR: The virtual advisor is an automated interactive voice content delivery system.We have integrated a set of the virtual advisor components so you can go out to a "my OnStar" Web site to establish information preferences. We've tied those preferences to the customer care records that we maintain in our database, which is basically the vehicle, the vehicle site, those kinds of things and then we put the partnerships together in area content providers. Let me give you an example: after connecting with the Virtual Advisor, you ask for traffic. We have the ability to deliver traffic delays or construction information for a metro area or for a radius around the exact vehicle location at the time you call in. You can go out to the Web and put in some specified routes, and then receive your traffic information by route as well. Basically what we do is update your profile with the vehicle data, so we know what type of vehicle, where the vehicle is, and then the content that we get from our partners.

WJ: It completely changed the concept of driving a car by having that information available. How are your subscribers taking to the service?

BR: One of the big challenges we've had since we started the business is getting people up-to-speed with all of its capabilities. I think everybody recognizes the OnStar safety and security messaging we've been doing on the radio and our advertising, but they can't be familiar with everything we have. A lot of the other services, like Virtual Advisor, are extensions that offer additional value. I think the users feel very similar to the way you do, but we're still trying to get that message out.

WJ: It seems to me that you're shifting into a total service mode. You're taking service to the next level by leveraging technology.

BR: I think we have a set of capabilities now that has enabled us to take service to a different level. That's really what it's all about: helping the people that buy our product in more creative ways that actually minimize the impact on them.

WJ: I understand that your messaging is done exclusively with JavaBeans. How does that system work?

BR: Early in our development process we made a conscious decision to adhere to the J2EE specification. As a result, we have selected products that serve as containers for our Enterprise JavaBeans (EJBs).

Messaging is handled in one of two ways:

- Our synchronous communications leverage RMI as the messaging mechanism between our session EJBs in the container and the client, or Web services (SOAP/HTTP) as the messaging mechanism between objects at the enterprise services layer.
- Asynchronous communications are handled through the Java Message Service (JMS) utilizing either a message driven EJB or JMS libraries implemented directly in the client.

We have found that the EJB construct has served us well in managing the large number of simultaneous client sessions we support and has facilitated the structuring and management of our business logic and persistent data objects.

WJ: I understand that General Motors is using IBM's voice technology for all of the voice interaction that goes with OnStar, which is an extraordinarily cool thing. How does that work?

BR: The technology is in the client device that is installed in the vehicle. The IBM product will be embedded into every vehicle that will be manufactured with our new digital hardware and it will govern all the interactions that take place between our unit and the subscriber. I think the big capability for us is what we refer to as continuous recognition enabling continuous digit dialing. With our older hardware, our Personal Calling users had to dial one digit at a time, and wait for the system to repeat the number. That's eliminated with our new continuous digit dialing. A couple of other functions, such as number storage, have also been streamlined.

WJ: Is that the same technology when you talk to the virtual advisor and they give you directions?

BR: It depends on what actions you're taking about. We're in the process right now of actually migrating some of the voice rec technology in our back office to newer platforms. Our back-end offering right now is around OnStar Personal Calling minute replenishment and automated demonstration/education. Currently all of our IVR capabilities are hosted on IBM products.

WJ: How do you keep it up and running?

BR: We have a 24x7 ops staff with access to a suite of tools that perform continuous monitoring. These tools enable us to keep track of the system performance, capacities, and status. Our call centers are geographically dispersed so that if we have a huge ice storm or snowstorm here in Detroit our systems are not impacted.

WJ: Now we're talking with Tony Lent, vice president of OnStar's Wireless Strategic Business Unit. Tony, what do you do at OnStar?

Tony Lent was named vice president of OnStar wireless in October, 2002 and is responsible for development and execution of OnStar's wireless business strategies, including OnStar Personal Calling and Virtual Advisor. Prior to joining OnStar, he held leadership positions with Sprint PCS, and most recently was area vice president of the Great Lakes consumer area in Sprint PCS' North Region. In this role, he was responsible for directing all sales, marketing, and consumer relations activities in the area. He also assumed the full profit and loss responsibility for the area, which was the second largest area in Sprint PCS' 11 areas.

Bruce D. Radloff is chief technology officer of OnStar, and is responsible for the oversight and development of current and future technologies associated with the OnStar service. Bruce joined General Motors Information Systems and Services in 1997 as director of systems engineering. He was then reassigned as chief information officer of OnStar and held that role until August 2001. Prior to joining GM, Bruce was director of market development systems for Bell Atlantic in Arlington, Va.

Tony Lent: It's a combination of sales, servicing, and marketing. There is also an operations component – I'm here to make sure that our services meet the needs of our customers.

WJ: Do you have anything to do with OnStar's Personal Calling program?

TL: Absolutely. Personal Calling is a fast-growing business that started in 2000 and has grown significantly over the last couple of years. In fact, we recently sold our 400 millionth minute. We estimate that our subscribers make and receive about 6 million calls in their vehicles each month.

WJ: That's a lot of minutes.

TL: It sure is. What we're finding is that our messaging really works. Eighty-five percent of our customers have a wireless phone already and we talk about using the OnStar phone "just in case" – OnStar Personal Calling is for those who might not be able to get service from a handheld portable or for customers who prefer to talk hands-free instead of using a handheld portable. It's also great for someone who forgot his or her phone, or battery life is low. Our OnStar minutes are good for up to 12 months. Customers can buy a bundle for as little as $14 for a 12-month bundle and there is no monthly access charge, no additional long distance or roaming charges. And the number of minutes sold is a good indication that our subscribers find increasing value in OnStar Personal Calling.

WJ: I test drove one of your Silverados recently, and what I found interesting is that the phone is literally built into the car. There is no phone; but there is a phone.

TL: It's an embedded phone and is a safer option because it is hands free and voice activated. Customers don't have to deal with a keypad; they can keep their eyes on the road; and they can even program our version of speed-dial numbers – we call them name tags – customers can program up to 20 of those for the people they call most often.

WJ: You have a very different type of handset than what the traditional cell phone user would see. Can you tell me a little bit about that?

TL: It's not a traditional handset at all. OnStar has three simple buttons, built either in the mirror or in the dashboard, that you use to make emergency calls, access OnStar safety and security services, or dial the phone. For OnStar Personal Calling you just press the white phone icon once, say "Call," then you can provide a name tag if you've already loaded that number in. In the event that you haven't loaded the nametag, dialing is as simple as saying, "Dial". Then it asks you to say the phone number.

WJ: The last thing to talk about is Generation 6 hardware, which is just about to come to market. I understand there are significant changes from what the public has experienced so far. Tell me about them.

TL: We just launched our Gen 6 hardware. A couple of key things about that are that it's our sixth generation of hardware in eight years and that's pretty phenomenal, especially for an automotive manufacturer. It is digital hardware – effectively dual mode, so it's a combination of analog and digital. That's important for us because it provides our customers with greater reliability in that they have access to a larger service area. Because it is digital, the call quality overall has improved. It's also much more user friendly especially in how you dial. Right now, when you dial a phone number for personal calling you have to do one digit at a time. With the Generation 6 hardware you can just say the number all in one stream. We find that people prefer continuous digit dialing. We've had a chance to test it out with our GM dealers, with our employees, and with customers, and they really appreciate the enhancements. So, we're really state of the art as far as voice recognition technology.

WJ: Did you develop that technology yourself?

TL: No. We did that with our partners at IBM.

WJ: Are there any other hardware changes for the vehicles?

TL: Recently we announced that early in 2005 our hardware will be modified to maintain or contain resident languages capabilities that will allow OnStar customers to have three languages built in to our hardware The hardware advancement provides a customer-focused service that will enable Spanish-speaking subscribers to use OnStar Personal Calling, in their preferred language.

WJ: I hear you've gotten some interesting feedback on your Generation 6 offering from people who buy fleets of cars and trucks.

TL: Yes. We have heard from our fleet customers, who are very interested in the capability to use a portable phone in conjunction with the OnStar service. They love the idea of the hands-free capability in the vehicle. They like the fact that there is a three-watt embedded wireless phone and an externally mounted antenna. And the hands-free aspect is consistent with most companies' safe driving policies. We see this as a big opportunity in the future.

WJ: Are you doing that integration with Bluetooth?

TL: No. We just announced a partnership with Verizon called the America's Choice with OnStar plan, which gives customers option to use Verizon wireless minutes with handset or OnStar-embedded phone and the convenience of one bill. Customers can use their Verizon Wireless plan minutes to make and receive calls on their handheld Verizon Wireless phones and OnStar in-vehicle phones and drivers can simply transfer incoming calls from their Verizon 4Wireless handset to the embedded phone in their OnStar-equipped General Motors vehicle.

The World of Telematics

An interview with Jim Ruthven

JIM RUTHVEN

WebSphere Journal: I'm talking now to Jim Ruthven, program director for IBM Telematic Solutions. Jim, what do you do?

Jim Ruthven: I'm responsible for the sales and development activities for IBM Telematics in the North American region. We have a team of salespeople who work with clients to use telematics to create value for themselves and stakeholders – in particular, automotive OEMs, as well as Tier One suppliers.

WJ: In case some of our readers don't know what telematics is, can you explain it?

JR: Sure. The simple explanation is that telematics is the notion of connecting vehicles to back-end systems to deliver telemetry and content. That is, information about vehicle performance, location, and time to back-end systems where they can be used. It also includes the delivery of content to vehicles for things such as Internet access, e-mail capability, information exchange – those kinds of applications and services.

WJ: I understand that you have some very interesting stuff going on right now with General Motors' OnStar Group. Tell me about that.

JR: We're very excited about that relationship. As you may know, OnStar is the world's largest telematics service provider (TSP), with more than 2.5 million subscribers. Hopefully, by the end of 2004, that will have grown to more than 3 million subscribers, which will make OnStar 5 times as large as the second largest TSP. Recently, OnStar and IBM announced that OnStar had chosen the IBM Embedded ViaVoice product as the foundation technology for voice capability in the next generation device – the sixth device iteration that they have installed in General Motors vehicles since their inception in 1996.

We're excited to be working with OnStar from a couple of perspectives. First, OnStar is working with IBM to help us continue to enhance our voice products so that they can be used with a variety of other customers and in other industries. It really strengthens our product because of the rigorous requirements associated with an automotive environment. As you may know, it's very different from speaking into a microphone in an office because of the noise characteristics and the environmental characteristics associated with the acoustics in an automotive cockpit. Working with OnStar and their Tier One suppliers, in particular Motorola and LG Electronics, we have been able to continue to enhance and improve the voice-recognition capability of our voice products

WJ: It sounds like, and correct me if I'm wrong, General Motors is coming out with a new version of OnStar. That's one generation every 12–18 months.

JR: That's correct.

WJ: Are the people at OnStar, from your experience, technology people?

JR: They really are. If you look at OnStar's history and how it started, they've done an excellent job of matching and mirroring the electronic industry with the automotive industry. My experience with the folks at OnStar is that they consider themselves to be information technologists – providing the kinds of safety and security applications that their customers want and need by marrying information technology with automotive technology.

WJ: That's really aggressive to have your technology switching every 16 months when you consider that we're talking about a car. It's not like a new function being added to, say, a computer operating system.

JR: In many ways, OnStar has set the bar in the industry from a usability perspective. For those who aren't familiar with the OnStar system, it basically consists of three simple buttons, which are typically located somewhere around the rearview mirror, that allow customers to access a variety of applications and services. So, OnStar really has paced the industry in

terms of HMI, which is why having them as a voice technology customer is so important.

WJ: What is HMI?

JR: Human-machine interface. The way that customers interact with the OnStar system is via voice (no pun intended). It doesn't have a screen or push buttons. General Motors firmly believes that telematics technology should enhance safety and security – keeping your hands on the wheel and your eyes on the road. So, if you had a screen and you forced the customer to interact with the telematic system by pushing buttons or looking at the screen, it takes away from the safety and security aspect. Given that, voice technology is truly a foundation technology for the OnStar system and being able to provide that foundation technology is really a feather in our cap.

WJ: How are this voice technology and WebSphere tied together? What would somebody have to do to embed voice technology?

JR: It really depends. One of the hallmarks of IBM's voice strategy or voice technology is that we really have two sides to it. On one hand, we've got the embedded voice side and in this case, Embedded ViaVoice is the technology that OnStar has adopted. It is a stand-alone technology.

WJ: Is that the same ViaVoice that you can run on your PC?

JR: It's based on that technology and, in fact, that was its original design point. It obviously has been enhanced to take into consideration computing platforms that are less powerful than a laptop or a desktop type of a system. That's one of the great things about working with companies like General Motors and Honda on their navigation systems. We had to take that base ViaVoice engine and migrate it to a much lower power of form factors.

So we have the embedded voice side, which is pretty much stand-alone. It operates inside the cockpit, and if you think about vehicle scenarios or use cases, there are some things that you want to have happen that never leave the car's cockpit. For example, a command along the lines of, "Let's roll the windows up or down" or "Change the radio station."

You don't want that voice command to have to go back to the server to be processed and then send the command back to the vehicle; you want all of that to be self-contained inside the vehicle. Embedded ViaVoice accomplishes that. So the continuous digit dialing function, say "Dial 123-456-7890," never leaves the vehicle.

On the other hand, there are some applications that require significant processing power on the back end. For example, "Find me the best route to the nearest McDonald's." You want the voice command to interact with systems in the back end to take advantage of the power of the back end both from a cost and from an upgrade perspective inside the vehicle. One of the hallmarks of IBM's strategy in the voice arena is that we've got both sides covered and we can help customers make intelligent choices on which technology is appropriate for the application. In OnStar's case, not only do they use our embedded voice engines for embedded applications, but for their virtual advisor application and interactive voice recognition capability, they are using IBM's WebSphere product on the back end.

WJ: Where on the back end?

JR: Actually, in the OnStar service center. That virtual advisor capability is based on IBM's IVR [Interactive Voice Recognition].

WJ: Is that part of the WebSphere Voice Server?

JR: It is, in fact, WebSphere Voice Server.

WJ: Can you expand on that?

JR: A hallmark of IBM's IVR capability is to take voice commands and allow them to interact with contact delivery systems on the back end. That's the function that your voice server is providing for the OnStar team.

WJ: What I think is interesting, and it's testimony to what OnStar is and what IBM has done for OnStar, is that you have Acura, Audi, Isuzu, Subaru, and Volkswagen all using the same technology as well as General Motors. Lexus Link is also based on OnStar.

General Motors lent me a 2004 Silverado with OnStar in it. It had a Corvette engine, all-wheel drive, and it was painted Fire Engine Red. But once you got past all the car things, what really struck me about driving that truck was this whole OnStar thing, how it changed the driving experience so I could always know where I was. I could get it to find hotels and places to eat. It could tell me if the truck was running well, which it always did, but I kept asking regularly.

There are about 150 million cars in the United States right now and OnStar has about 2 million customers. Moving forward, this telematics and OnStar thing will become a very mainstream product. I'm curious how you see telematics changing our life experience.

JR: That's an excellent question and fairly wide ranging; it really gets to IBM's point of view as it relates to the telematic industry. For your readers who aren't aware, OnStar has been around since about 1996, which is when IBM actually got into the telematics marketplace by working with Delphi to create a concept car around some telematics use case scenarios. In 1996, the belief was that consumers would pay billions of dollars to have personalized news, weather, stocks, and sports delivered to their vehicle. It was during the Internet bubble. People were experimenting with new business models and believed that others would actually spend money, on a monthly subscription fee, for this kind of information. Over time, we learned that it wasn't the case. In fact, IBM came out of the chute saying, "You know, there are other devices out there." Namely, that includes cell phones, PDAs, and other portable handheld devices that customers and clients carry with them that aren't tethered to the automobile and that provide the kinds of functions in which the consum-

ers are interested, in a much more elegant and useful way.

The real value of telematics was in enhancing the vehicle experience with vehicle-centric applications and services. For example, we believe that OEMs can use telematics technology and the capability to deliver diagnostic information to dealers. It allows them to ensure that they have the right technician with the right part on-site, positively impacting customer satisfaction.

In addition, when the yellow "check engine" light comes on, we know that it can mean any one of 3,000 different things. Just looking at the light, you really don't know exactly what's wrong. When that light comes on now, you can call the OnStar service center and the customer support representative can tell you what the problem is. We're envisioning a time when that information can be sent to a dealer, in case a significant problem arises and requires service. So, it's those kinds of applications and services on which IBM has really been focused in terms of the automotive industry.

Then, they can take that very same information and get it back into the hands of automotive company engineering and manufacturing department. That way, they can understand the problems that are happening in the vehicles more quickly and put engineering plans in place to fix them – without running the risk of manufacturing those same issues or problems into subsequent vehicles. So there's a real, tangible impact here – telemetry being used to create value, not only for consumers, but also for the automotive value chain.

WJ: Telematics is interesting. It's different than a lot of other technologies when I think about it. It changes the way people live and many times, it's transparent, where the user doesn't even realize that there is some type of technology that's somehow interceding to serve them.

JR: Exactly. If you think about the vehicle experience, the thing that drives people crazy is having to take their vehicles to the dealer for service.

But a societal impact of this kind of technology is that automakers will be able to understand, perhaps even before the customer understands, that there's a problem with the vehicle. By the nature of how vehicles are developed and produced today, mainly with increasing amounts of electronics and software, automakers probably have the ability to fix vehicles without the customer even knowing that there was a problem.

WJ: Like predictive maintenance and prognostics. What prognostics?

JR: Prognostics is the idea of being able to take diagnostic information from the vehicle and predict what's going to happen.

For example, when electronic control units fail in vehicles, they don't fail all at once. They fail over time and they begin to display diagnostic trouble codes well before they fail. So, the ability to understand those codes, together with automotive knowledge and experience, can enable companies to predict whether or not a certain electronic control unit is going to fail and be able to take corrective action before it does.

There are other interesting aspects to this capability. Combining telematics with in-vehicle electronics and software architectures will allow automotive companies to create capabilities around horsepower on demand.

Let's say you've bought a vehicle and you run it in economical mode while driving in the city. And you're getting ready to go to the mountains and tow your boat. In the future, you'll actually be able to have the capability to download horsepower. It may not run as efficiently, but you'll have more effective performance.

WJ: You're suggesting they pay for the horsepower as they would for their vacation and then when the time period ends, the horsepower goes away?

JR: That's correct. The societal impacts of this are profound. You can even have better gas mileage with more economical driving most of the time, using only the maximum capability of the engine when you need it.

WJ: When do you see that actually being a product?

JR: Lots of things have to happen before that actually occurs, but we're thinking 7–10 years from now.

WJ: What else do you see on the horizon for telematics over the course of, say, the next 3–5 years?

JR: Well, I think you're going to see some of the things that I described earlier. You'll certainly see enhanced integration between vehicle telemetry delivered by telematic systems, dealer systems, and back-end automotive systems.

The other hallmark that's key to IBM's thoughts is that there is an ecosystem developing around this telemetry and this vehicle data. Other industries are beginning to pick up on the value and the capabilities provided by this data. For example, the insurance industry is keenly interested in information about vehicle performance, vehicle location, and the times it's being driven, vehicle speed, and insurance notification in the event of a crash. All of this data is available today – while not necessarily being delivered wirelessly yet – but as that capability in telematics becomes more prevalent, you'll see insurance companies become interested in creating specific policies for individuals. In fact, IBM is working with Norwich Union Insurance in the UK on just such a pilot project. 5,000 people have volunteered to put devices in their vehicle's and exchange some privacy for improved policy rates because the insurance company can reduce their risk by knowing how they're driving.

WJ: That would be like on-demand liability insurance?

JR: They call it "pay as you drive." We would call it on-demand insurance capabilities.

WJ: Very interesting. So, that will be a much fairer system if it comes to pass for a conservative driver who obviously pays for the non-conservative people that drive on our highways.

JR: I'd like to say that my mom would be an excellent candidate for "pay as you drive" insurance because she has a spotless driving record; she never speeds.

WJ: And she probably doesn't drive to high-risk areas.

JR: She doesn't drive to high-risk areas. She drives during normal business hours. She's not out late at night, but she's lumped in from an insurance rating perspective with folks who aren't necessarily as conservative. So, she's a primary candidate for that.

WJ: On the flip side, consider young men under 25 who pay the highest insurance. If you get a kid who's primarily driving at the speed limit during daylight hours and who doesn't go to high-risk areas, you could, in theory, charge him less than the young man who's out at 4:00 AM on Saturday night.

JR: In fact, young drivers are one of the targets of the UK pilot.

WJ: I hear that telematics with WebSphere is starting to be applied to the company that makes international trucks and international harvesters for people that use fleets.

JR: It's actually International Truck and Engine Corporation. We've been working with them for over 2 years now to help them develop their telematic strategy, implementation, and execution.

WJ: Are they actually doing that now or is that going to happen in the future?

JR: They're in pilot mode now. In fact, they just announced that they're launching their production telematic service in July. So they're in pilot mode now with several customers testing out the telematic system that we've built with them.

It's interesting if you start to think about the commercial vehicle aspect. Many of the issues associated with passenger cars disappear on the commercial fleet side. For example, privacy issues are much less stringent on the commercial fleet side than on the passenger car side.

WJ: Because everybody wants to know.

JR: There are a couple of aspects to it. On the passenger car side, there is normal fear of Big Brother watching. IBM's position is that there will be many ways for an opt-in kind of service and capability. On the commercial fleet side, fleet managers and owners actually own the truck. They allow their employees to use them and therefore, the issue of privacy is much less stringent. They have a right to know how their vehicles are being driven, where they are being driven, and how they are being used. So, many of the private citizen issues go away. But the return on investment of the commercial fleet side is also much easier to see because you can begin to get into areas such as driver productivity and asset utilization – real dollar savings and dollar values to the fleet manager's side of the business. In many ways, we see a more rapid acceleration of telematics in the commercial vehicle side than in the passenger car side.

The benefits and the notion of the ecosystem that I talked about earlier are still there. We know, for example, that there are companies, such as Fleet Risk Advisor, taking telematic data and combining it with data about drivers and their performance to help insurance companies work with fleets to improve the accident profile and reduce the amount and the severity of accidents that fleets have.

We know that asset utilization (truck utilization) is an important part of the promotional vehicle side of the business and telematics can certainly help fleet managers in their service organization understand the performance characteristics of vehicles and be able to perform preventative maintenance before they fail. Not only are they improving their asset utilization, but they are increasing customer satisfaction because they work at deliveries and they can make and meet their commitments.

WJ: You're talking about some pretty powerful stuff here. Do you see what you're doing as giving the next generation of market leaders the technological advantage that they need to win?

JR: Absolutely. If you look at the International Truck example that I just used, they fundamentally believe that they will be able to use this kind of technology to differentiate their vehicles in the marketplace to help them increase market share in a couple different ways. One way to help them to build better vehicles is by enabling them to know how their vehicles are performing in the field. A second way is by allowing them to consult with their customers to help choose the features and functions that will be most effective.

For example, in one of their pilots they're working with a company that has a number of tow trucks and they found that, although the tow truck company bought each vehicle with a winch on the front, only 5% of tow truck operators use that winch and obviously, every one of them uses the tow hook on the back. So, the company can counsel its customers to buy only 5% of the vehicles with the winch on the front. That way, they can track the trucks and dispatch them to places where the winch on the front was required. In addition, they found that the boom on the back required additional capability in the electrical system of the vehicle. They were then able to feed that information back to the engineering department to beef up the electrical side of the vehicle architecture. As a result, they could deliver a vehicle that performs better in the field.

To me, this is an excellent example of how International will be able to enhance its position in the marketplace, because it is the only company in the industry today that has the capability. They can consult with their customers and can improve their product at the same time using that technology. It really is transformation from that perspective. It's also transformation from the perspective of helping them to become more efficient and effective as a company. This data forces manufacturing and engineering as well as the retail side of the business to work together. It's through communication and conversation that they are able to improve the effectiveness of the vehicle and their ultimate products.

Find your portlets quickly and easily

Using the Search Capabilities of WebSphere Portal V5 - Part I

BY TILAK MITRA

Tilak Mitra is a Certified Senior IT Architect at IBM. He specializes in mid- to large-range enterprise and application architectures based on J2EE, MQ and other EAI technologies. tmitra@us.ibm.com

A typical portal application usually requires some form of search capability. Portal designers usually look into search engine products like Lotus Extended Search or open source implementations like Lucene in order to satisfy the search requirements. Although these search engines provide sophisticated search capabilities, the Native Search feature of the WebSphere Portal Server 5.x (hereafter called WebSphere Portal) has a fairly rich set of search engine capabilities that could satisfy the search requirements for many portal based applications.

In this two-part series I will introduce you to the basic capabilities of WebSphere Portal's native search. While Part 1 will take a very simple search scenario and walk you through its implementation and configuration, Part 2 will introduce more advanced search capabilities.

Example Prerequisites

To understand the scenario in this first article, you need to have only a very basic understanding of WebSphere Portal's configuration options – primarily the fundamental administration options. That said, this article is targeted not only towards the portal administrator, but also to those who want a better understanding of WebSphere Portal's native searching capabilities.

I used IBM WebSphere Portal Enable for Multiplatforms v5.0.2 for Windows environment for this example. If you want to run through this example, your machine should have WebSphere Portal installed on a machine with at least 1GB RAM. You also need basic Internet connectivity because the example involves accessing an external IBM Web site (www.ibm.com).

The Portal Search Engine

WebSphere Portal v5.0.2 provides a search engine that can crawl Web sites (external and internal), index aggregated content, and categorize documents. The categorization can be implemented by either a predefined set of categories or using a user-based custom categorization process. The categorization facility of WebSphere Portal Native Search includes an extensive list of predefined categories that are grouped into high-level business industry areas (e.g., finance, transportation, etc.). You can use these predefined categories for your portal applications and the Categorization Engine will automatically categorize content for you. Alternatively, the user-defined categories provide the flexibility of creating custom category trees that may be used to categorize the incoming search results.

The Portal Search engine collects documents from multiple sites into a single collection. The engine can be configured to automate the process of crawling the sites periodically and updating the search content. You can also manually trigger the collection update process.

When a collection is defined and activated, one of the main functions performed by WebSphere Portal Native Search runtime is the creation of indexes. An index is formatted data that is used by the search engine in to store, read, and match queries against it. Indexes provide a way of searching content in a more efficient manner. These indexes are stored in the file system in a location that should be accessible to the portal runtime. With this architecture, it is a simple matter of extending the portal native search capabilities to a clustered environment. All that has to be set up is a mounted file system (where the indexes are stored) that is then made accessible to each of the portal nodes in the cluster.

WebSphere Portal includes a Document Search portlet that is included in the list of portlets that is installed by default. The Document Search Portlet has the ability to crawl and index Web content sources and attachments. It can also create, schedule, and maintain search

indexes, thus providing search functionality that is comparable to other Web search engines. This Document Search portlet can be used in production-ready portal applications. We will be using this portlet in this example.

These are only some of the features of the Portal Search Engine. For a complete description, see the Content search section of the WebSphere Portal Infocenter or WebSphere Portal Administration Guide (see Resources).

The remainder of this article walks you through a simple search scenario in which you:

1. Set up a collection, using a predefined static taxonomy (a.k.a category).
2. Install the Document Search portlet onto a page.
3. Run the search to see the result.

Setting Up the Collection

Now let's start the scenario. First you create a collection of document URLs from a single Web site. These documents are the ones you want indexed and searched.

To create the collection:

1. Log on to the portal as an administrator (e.g., wpsadmin).
2. Go to Administration->Portal Settings->Search Administration. Select Create Collection (see Figure 1).
3. Supply the following information in the "New Collection" page:
   - *Location of Collection:* "IBMCrawlerPredefined". (Note that there is nothing special in this name.)
   - *Specify Collection Language:* "English"
   - *Specify Categorizer "Pre-Defined":* We are using predefined categories provided by Native Search in this example.)
   - *Select Summarizer:* "Automatic". (Summarizer is a feature used by the Portal Search Engine to form a summary of the entire document based on the most important sentences of the original document. Setting it to Automatic implies that we are going to use the summarizer functionality that is provided out-of-the-box
   - *Check the checkbox* for "Remove common words from queries...". (This removes common words like "on", "and", etc.)

   Click OK. The page should look like Figure 2.
4. An empty collection is created. We need to add one or more sites that will be a part of the newly created collection. In this example we will use the external IBM site (www.ibm.com). Note that multiple sites can be added to this collection, although for our example we are going to use a single site. Also note that a folder by the same name (IBMCrawlerPredefined) is created in the filesystem under InstallDir\WebSphere\AppServer directory (where InstallDir is the directory where WebSphere Portal is installed in your machine). WebSphere Portal stores the indexes (for this collection) in this folder (see Figure 4). Click the Add Site link (see Figure 3).
5. Choose the options shown in Figure 4 for the new site that is being added to the collection. A brief explanation of some of the attributes shown in Figure 4 is given here:

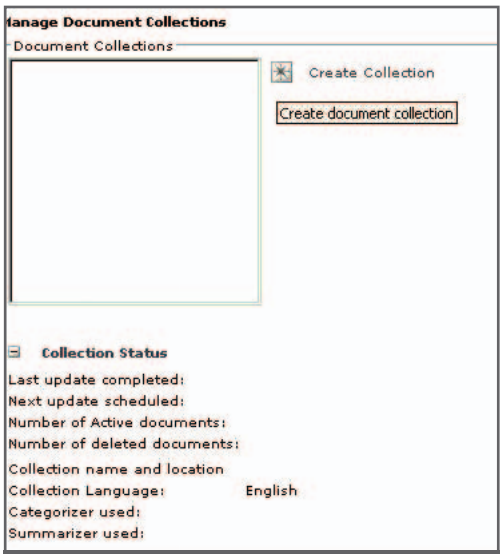


FIG 1: CREATING THE COLLECTION

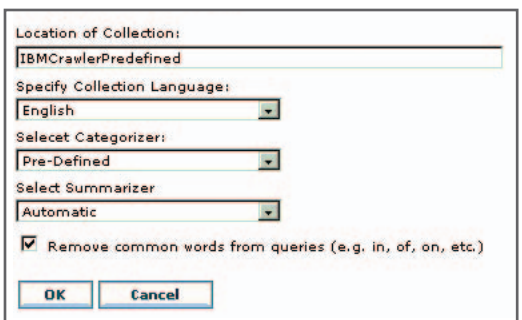


FIG 2: NEW COLLECTION" PAGE

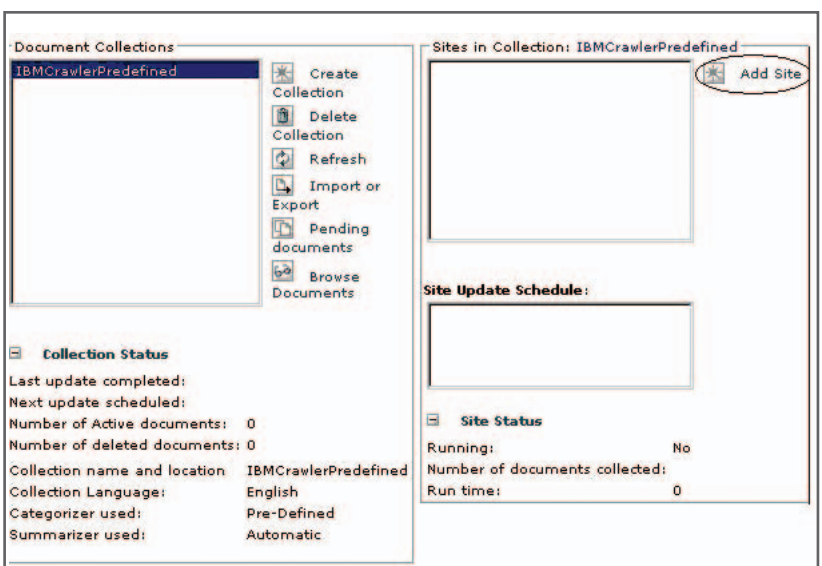


FIG 3: "ADDING A SITE" TO THE COLLECTION

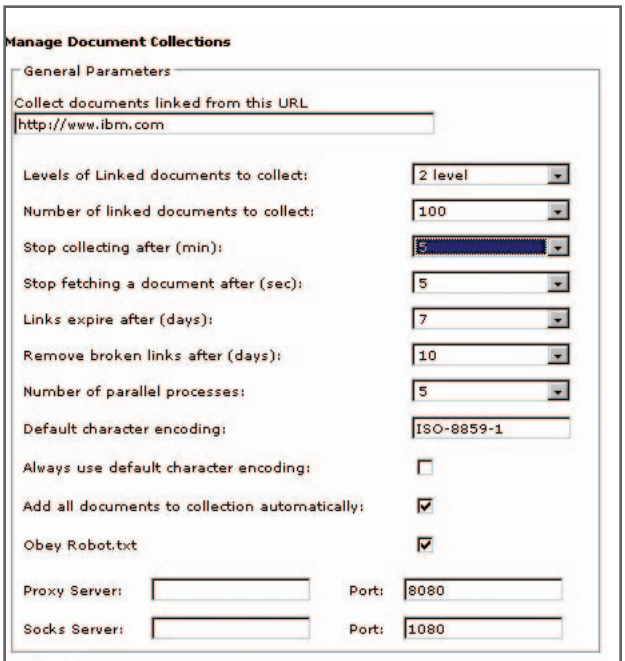


FIG 4: INDEXES FOR THE COLLECTION

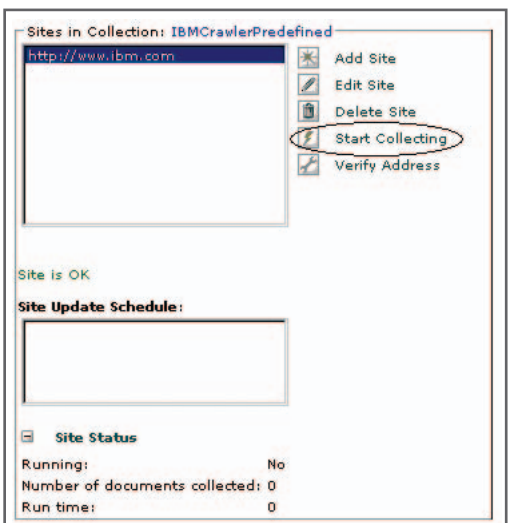


FIG 5: START COLLECTING LINK

- The "Collect documents linked from this URLs" attribute is set to www.ibm.com. This denotes the site where documents are to be collected
- The "Levels of Linked documents to collect" attribute is set to 2. This implies that two levels of URL redirections (from the main URL) will be navigated and searched for content.
- The "Number of linked documents to collect" attribute is set to 100. This implies that at most 100 documents (that match the search criteria) may be collected from the main URL.
- The "Number of parallel processes" attribute is set to 5. This denotes the number of parallel crawlers that go out to the site specified to fetch Web pages. The larger the site, the more parallel crawlers could be used to retrieve all pages from the site faster.
- The "Always use default character encoding" attribute is not checked. If this is not checked/used, then the encoding information provided with the HTML content is used for encoding. If this is used it provides the administrator with a means of overriding the encoding of the incoming HTML content.
- The "Add all documents to collection automatically" checkbox is checked. This implies that the incoming documents (from the URL) do not need any manual editing and may be directly added to the collection by the portal search engine.
- The Obey Robot.txt checkbox is checked. This is allows for Webmasters to provide directives to crawlers (robots) as to what pages the robot may or may not include in scans. This file is located in the same virtual directory as that from which the portal's content is served.

For an explanation of the other attributes please refer to the WebSphere Portal Server InfoCenter's Content Search Section (see Resources).

6. Click the Create button. Now you are ready to start collecting links off the site that you just defined.
7. The actual collection (of documents from the site) may be initiated by clicking the Start Collecting link in the "Sites in Collection:IBMCrawlerPredefined" panel (see Figure 5).
8. Click the twistie on the Site Status section to watch the progress of the document-collection process. This operation takes a few minutes to complete. It is advisable that nothing be done until the status is updated with a completion timestamp.

Now that you have created and configured the collection, it's time to prepare to test the search. WebSphere Portal provides a portlet – Document Search – that you can configure to use the collection you just created.

Configuring the Search Portlet

With the collection configured we now need to configure and subsequently use the Document Search Portlet. WebSphere Portal comes with a set of installed portlets out of the box ready to be customized and used. The Document Search Portlet is one such portlet. This portlet can be customized and used in a production-ready portal application.

1. From the Administration Tab (on the top right-hand corner) we need to get to the list of installed portlets. Click on the "Portlets->Manage Portlets" link from the left navigation bar. This displays the list of portlets. Find out the Document Search portlet and make a copy of it.
2. A copy of the original Document Search Portlet is created. Note that in most portal installs, the initial status of this newly created (copied) portlet is set to Inactive. Highlight the cloned portlet and click on the Modify Parameters link. The IndexName attribute is set to the value of InstallDir\WebSphere\AppServer\IBMCrawlerPredefined (where InstallDir is the directory where WebSphere Application Server is installed and IBMCrawler-

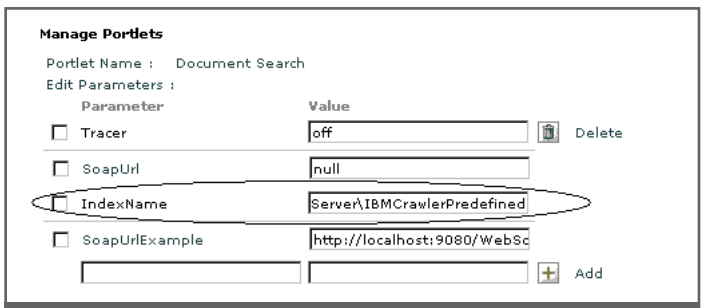


FIG 6: SAVED INDEXNAME ATTRIBUTE VALUE

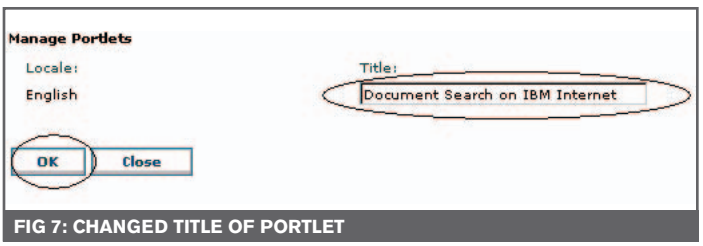


FIG 7: CHANGED TITLE OF PORTLET

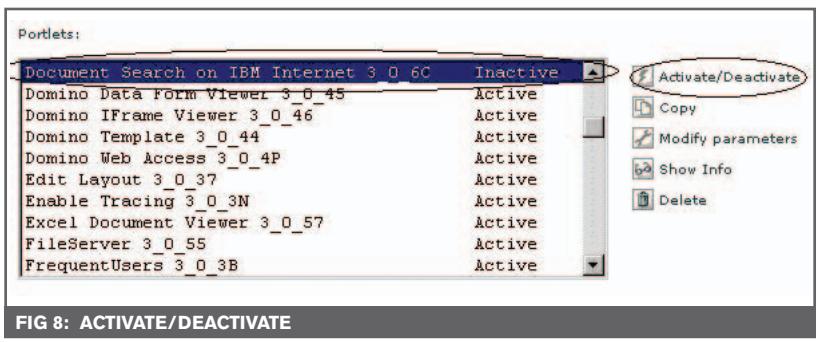


FIG 8: ACTIVATE/DEACTIVATE

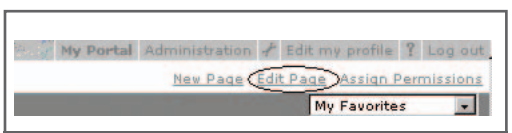


FIG 9: EDIT PAGE

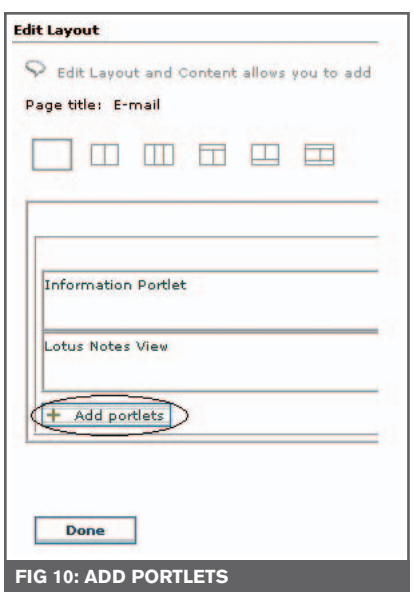


FIG 10: ADD PORTLETS

Predefined is the foler under which the WebSphere Portal stores the index and other metadata for the collection). Click the Save button (not shown in Figure 6. Scroll down to locate the same). The Indexname attribute value is saved.

3. In the same page the Language attribute can also be set. In our example we chose the default language (English). We want to change the Title of our portlet to give it a better name. Click on the link labeled "Set title for selected locale". In the subsequent Manage Portlets page we can change the title. For this example I changed it to "Document Search on IBM Internet". Click OK (see Figure 7).

   Pressing OK brings us back to the portlet's property modification page. Once back there, click the Save button again (as in step 2 earlier). This will save all the parameters that we modified for our portlet. Click on the Cancel button immediately to the right of the Save button. This will take us out of the parameter modification page altogether.
4. The portlet (renamed) will be highlighted. If the portlet is in Inactive mode we need to activate it. This is required since only activated portlets can be used to customize a page in the portal (see Figure 8).
5. All set!!! All that is left to do is to add the new customized document search portlet into one of the top-level portal pages.

Adding Our Custom Portlet

The portal is made up of top-level pages, which can be thought of as tabs in the most common UIs that we come across in our daily life. The top-level pages are meant to separate out content. They organize content according to a logical grouping of functionality. Quite a few top-level pages are provided out of the box (e.g., Content Publishing, Welcome, Documents, My Work, etc). For this example, I was searching for the page that had the least amount of content where I could put our customized portlet and make it conspicuous by its presence (a feeling that we achieved something!!!). I decided the "My Work" page to add to our portlet.

We are now going to add this new portlet to the existing real estate area on this page. To do this, click on the Edit Page link on the top right-hand corner of the page (see Figure 9).

6. On the Edit Layout page (see Figure 10, click the Add Portlets button.
7. A search screen will appear where we need to search for our portlet from the ones that are already installed. Key in the search keywords as "Document Search on IBM" and hit the Search button, you will see that a match was returned with our portlet (see Figure 11). Click on the checkbox by our portlet and click OK. This will bring us back to the original page on which we started

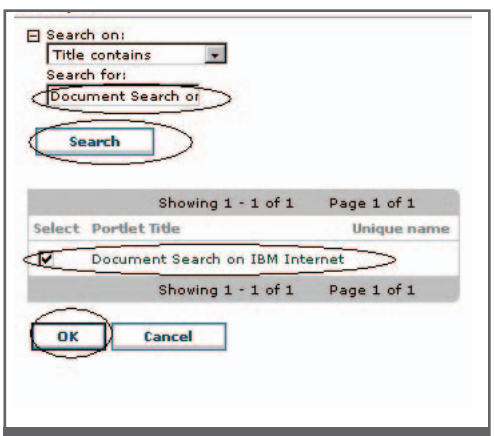


FIG 11: DOCUMENT SEARCH

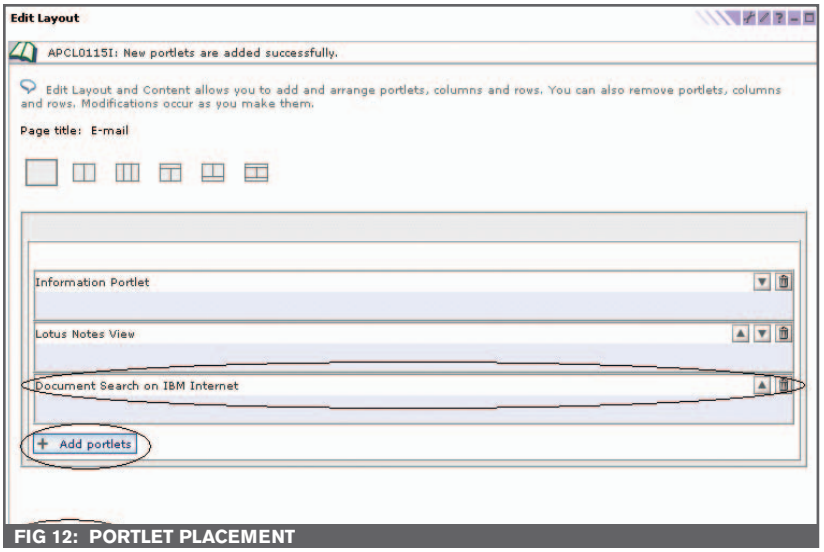


FIG 12: PORTLET PLACEMENT

to add our portlet to the My Work top-level page. At this point you can play around with the page and place the portal vertically, horizontally, in a T layout, etc. I accepted the default layout and click on the Done button (see Figure 12).

8. This completes the customization and setup process. We're ready to run!

Running the Search

Back in the "My Work" page we can see that our portlet has arrived, ready to be executed. In the search textfield, I keyed in the word websphere and hit the Search button to the right. An excerpt of the output of the search results are shown in Figure 13.

Notice that each search result is categorized using the Portal Search Engine's built-in, predefined categories. There are two interesting things to notice there:

1. Every search result is followed by a summary. This summary is created by the Portal Search Engine's Summarizer function. The Summarizer creates a summary for each of the documents that have a certain narrative quality
2. Each search result is designated as part of a category. The categories used here are the predefined categories that come with the WebSphere Portal's Native Search. The incoming documents are categorized by the native search engine's Categorizer function. The Categorizer places the documents in a category based on certain internal algorithms. Pages that do not qualify for any of the provided categories are dropped into the "Uncategorized" category.

The top-level node in the predefined category tree is called "root". The first search result is categorized under a main category, called "Business & Commerce," followed by several levels of subcategorization. The second search result is one of those that the Categorizer could not qualify under one of the predefined categories. Hence, it is left as "Uncategorized" under the top-level ("root") node.

Conclusion

In this article we saw how quickly we can set up a basic search portlet using the out-of the-box Document Search Portlet, and how we can do some simple customizations on the existing portlet.

While in this article I demonstrated a search function using the predefined categories within WebSphere Portal, in my next article I will take you through a more flexible and advanced feature in which users can define their own custom category hierarchy for incoming documents to be managed and presented to the user through the portal.

Resources

- The InfoCenter is the most comprehensive place to find information on WebSphere Portal Server V5.x: http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/index.html
- IBM Redbook, *IBM WebSphere Portal V5 A Guide for Portlet Application Development.* is an excellent book on portlet development: http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246076.html?Open
- WebSphere Portal Zone: www-106.ibm.com/developerworks/websphere/zones/portal/
- Web crawler robots provides references to Web robots: www.robotstxt.org/wc/robots.html

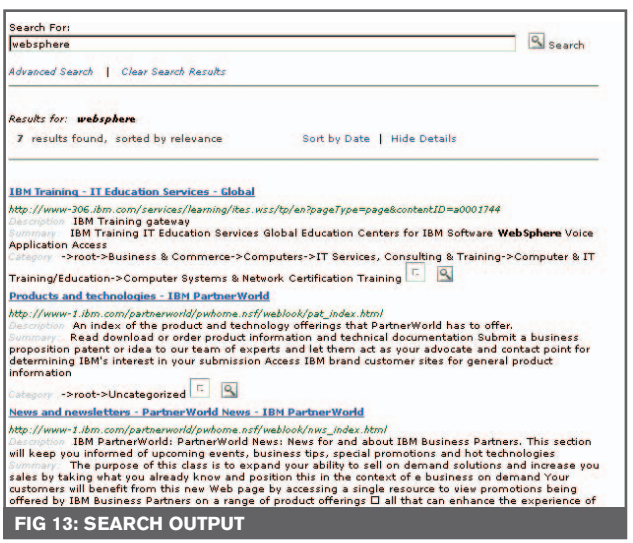
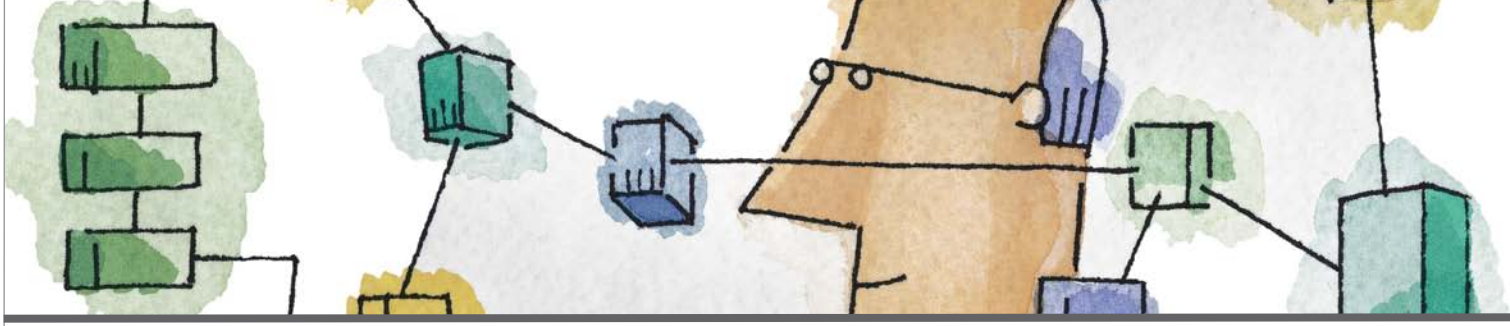


FIG 13: SEARCH OUTPUT



Performance Testing Web Services

Strategies and best practices

BY SERGEI BARANOV

Sergei Baranov is a software engineer for Parasoft. He's a load test technical lead for SOAPtest – Parasoft's tool for testing Web services. His interests include distributed and concurrent computing as well as software performance issues. He holds a degree in electrical engineering (MSEE) from the Moscow State Institute of Radio Engineering, Electronics, and Automation. The author would like to thank Jim Clune, Cynthia Dunlop, and Marvin Gapultos of Parasoft for their reviews of this article.

The successful development of scalable Web services on WebSphere (or any other application server) requires thorough performance testing. Applying a well-designed, consistent performance testing methodology throughout the development life cycle is key to satisfying a Web services application's performance requirements.

This article provides strategies for creating load-test scenarios that reflect tendencies of the real world environment. To help you apply these strategies, it introduces best practices for organizing and executing load tests, and suggests how these best practices fit into a Web services application's development life cycle. It also demonstrates how to utilize performance monitors offered by WebSphere and the operating system, as well as how to analyze load-testing results. The bulk of the discussion assumes that WSDL is used to describe the Web service's interface, and that SOAP and HTTP are used for the messaging and transport layers, respectively.

Creating a Test Plan

INCORPORATING PERFORMANCE TESTING INTO THE WEB SERVICE DEVELOPMENT LIFE CYCLE

Although performance and scalability are critical parts of any non-trivial Web service, performance testing is often left for the later stages of development. Introducing performance testing at the early stages of development, or even prototyping, brings the following benefits:

- Performance-related bugs (such as concurrency problems, memory leaks, and unreleased resources) are caught immediately after they are introduced, which helps isolate the culprit to the recently checked-in code or recent infrastructure changes.
- New features' performance costs can be evaluated and quantified.
- Project management has constant awareness of the application performance and has more confidence in meeting the release dates. This greatly reduces the risk of missed deadlines and overspending on expensive enterprise application profiling tools and consultants.

CREATING LOAD TEST SCENARIOS

Performance tests should be started after the functional tests execute successfully. The best way to perform load testing is to have the load test application's virtual users run the complete functional test suite. Load tests should identify and report all transport, message, and application errors.

The usefulness of the performance test results is directly related to how closely the tests emulate request streams that the Web services application will be handling once it is deployed in the production environment. To design load test scenarios with real-world value mixes, request types, sequences, and intensities, it is helpful to consider the load test scenario categories shown in Figure 1.

Type of Use

Depending on the type of deployment, your Web services can be exposed to various types of SOAP clients. These clients may produce unexpected, erroneous, and even malicious requests. Your load-test scenarios should include profiles that simulate this user. The more your Web service is exposed to the outside ("uncontrolled") world, the greater the probability of a non-regular usage. The "misuse" and "malicious use" categories may include invalid SOAP requests as well as valid requests with unusual or unexpected values or request sizes.

For example, if your service uses arrays or complex types, examine your WSDL and create load test scenarios that emulate requests with expected average and maximum possible element counts, as well as element counts that exceed the allowed maximum.

```
<xsd:complexType name="IntArray">
  <xsd:sequence>
    <xsd:element name="arg" type="xsd:int" maxOc-
curs="100"/>
  </xsd:sequence>
</xsd:complexType>
```

Measure service performance with various sizes of client requests and server responses. If the expected request sizes and their probabilities are known (for example, based on log analysis), then create the request mix accordingly. If such data is unavailable, test with the best-, average-, and worst-case scenarios to cover the full performance spectrum.

Emulation Mode

A Web service may or may not support the notion of a user. More generically, it may be stateful or stateless. Your decision to use either "virtual user" or "request per second" request emulation mode should be based on this criteria. For example, the load of a stateless search engine exposed as a Web service is best expressed in terms of a number of requests per second, because the notion of a virtual user is not well-defined in this case. A counter example of a stateful Web service is one that supports customer login, such as a ticket reservation service. In this context, it makes more sense to use virtual user emulation mode.

If your Web service is stateless and you have chosen the request-per-second approach, make sure that you select a load test tool that supports this mode. If a load test tool can sustain only the scheduled number of users, the effective request injection rate may vary substantially based on the server response times. Such a tool will not be able to accurately emulate the desired request sequence. If the number of users is constant, the request injection rate will be inversely proportionate to the server processing time and will be likely to fluctuate, sometimes dramatically, during the test. Web services load test tools offer both "virtual user" and "request per second" modes.

When load testing stateful Web services, such as services that support the notion of a user, make sure that you are applying appropriate intensity and concurrency loads. Load intensity can be expressed in request arrival rate; it affects system resources required to transfer and process client requests, such as CPU and network resources. Load concurrency, on the other hand, affects system resources required to keep the data associated with logged-in users or other stateful entities – session objects in memory, open connections, or used disk space. A concurrent load of appropriate intensity could expose synchronization errors in your Web services application. You can control the ratio between load intensity and concurrency by changing the virtual user think time in your load test tool.

Content Type

When load testing Web services, it is easy to overlook the fact that SOAP clients may periodically refresh the WSDL, which describes the service, to get updates on the service parameters it is about to invoke. The probability of such updates may vary depending on the circumstances. The test team can analyze access logs or make reasonable predictions based on the nature of the service. If the WSDL access factor (the probability of WSDL access per service invocation) is high and WSDL size is compatible with the combined average size of request and response, then network utilization will be noticeably higher in this scenario, as compared to the one without WSDL refresh. If your Web service's WSDLs are generated dynamically, the high WSDL access factor will affect server utilization as well. On the other hand, if your WSDLs are static, you can offload your application server by moving the WSDL files to a separate Web server that is optimized for serving static pages. Such a move creates increased capacity for processing Web service requests.

Type of Load

To ensure that your Web services application can handle the challenges it will face once it is deployed in production, you test its performance with various load intensities and durations. Performance requirement specifications should include metrics for both expected average and peak loads. After you have run average and peak load scenarios, conduct a stress test. A stress test should reveal the Web services application's behavior under extreme circumstances, which would cause your application to start running out of resources, such as database connections or disk space. Your application should not experience an unrecoverable crash under this stress.

Performance degradation – even dramatic degradation – is acceptable in this context, but the application should return to normal after the load has been reduced to the average. If the application does not crash under stress, verify that the resources utilized during the stress have been released. A comprehensive performance-testing plan will also include an endurance test that verifies the application's ability to run for an extended period of time. Endurance tests are run for hours or days, and could reveal slow resource leaks that are not noticeable during regular tests. Slow memory leaks are among the most common – if they are present in a Java environment,

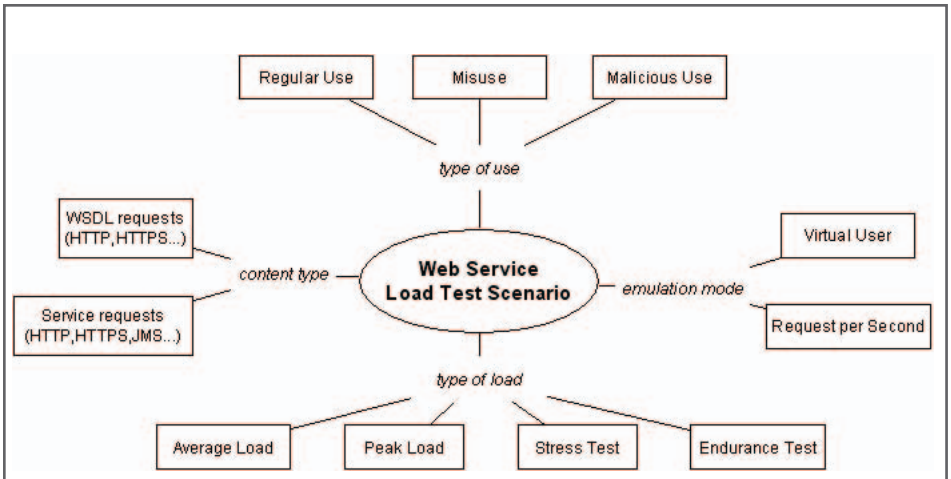


FIG 1: LOAD TEST SCENARIO CATEGORIES

these leaks could lead to a java.lang.OutOfMemoryError and the crash of the application server instance.

CREATING A REAL WORLD VALUE MIX

To better verify the robustness of your Web service, you should use your load test tool to generate a wide variety of values inside SOAP requests. This mix can be achieved, for example, by using multiple value data sources (such as spreadsheets or databases), or by having the values of the desired range dynamically generated (scripted) and then passed to virtual users that simulate SOAP clients. By using this approach in load tests of sufficient duration and intensity, you can test your Web service with an extended range and mix of argument values that will augment your functional testing. Depending on the circumstances, it may be advisable to run the mixed request load test after all known concurrency issues have been resolved. If errors start occurring after the variable request mix has been introduced, inspect error details and create functional tests using the values that caused your Web service to fail during load testing. These newly created functional tests should become part of your functional test suite.

WEB SERVICES DESIGN CONSIDERATIONS

The load test scenario creation practices previously mentioned help reveal performance problems that have already been introduced into a Web services application. However, it's equally important to prevent potential performance problems at the design and development stages. By focusing on prevention as well as detection, you reduce the number and severity of performance problems that you would have to identify and repair later, which in turn streamlines the development process and conserves development/testing resources. This section offers some tips on designing Web services in a way that prevents performance problems.

MINIMIZE THE NUMBER OF SERVICE REQUESTS AND ACCUMULATED TRAFFIC SIZE

Many current Web services development software packages, including WebSphere Studio Application Developer, offer tools that greatly automate the exposure of Java code via Web services. Although these tools are effective development instruments, indiscriminate use of such tools can cause performance problems. For instance, assume that we want to use the following Java interface for customer registration:

```
public interface Customer {
  void setName(String first,String last);
  void setAddress(Address address);
  void setTelephone(String phone);
  void setAccountNo(String account);
}
```

If the Customer interface is exposed as a Web service, it will take four SOAP calls to complete the customer registration operation. We can reduce the number of remote calls and accumulated request processing time and traffic size by introducing a single register operation as follows:

```
void register(String firstName,String
lastName,Address address,String phone,String
account);
```

An even better approach would be to introduce a CustomerRegistrationData type that will hold all of the parameters necessary for registration:

```
void register(CustomerRegistrationData data);
```

When designing a Web services interface, it is advisable to define a set of high-level actions that the Web service user would want to perform. Then, create a set of SOAP calls that perform the desired high-level actions with the minimal request quantity and traffic volume between the client and the server.

MINIMIZE POTENTIAL PERFORMANCE PROBLEMS

As previously discussed, your service can be exposed to erroneous or malicious requests. It's often possible to reduce the performance degradation caused by such requests. For example, assume that your Web service's WSDL was autogenerated and has complex types that allow an unlimited number of elements as a result of a default conversion of a Java array, or a vector.

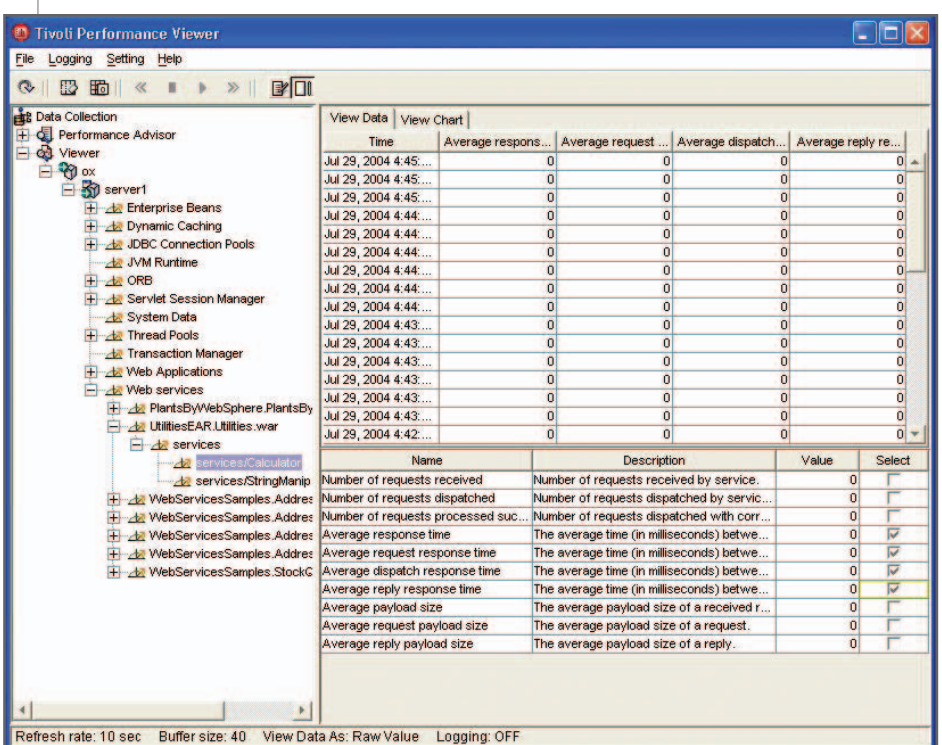


FIG 2: WEB SERVICE STATISTICS EXPOSED BY WEBSPHERE

```
<xsd:complexType name="stringArray">
  <xsd:sequence>
    <xsd:element name="id" type="xsd:string"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

Your application business logic, however, may have a well-defined limit of the number of elements in the complex type:

```
<xsd:element name="id" type="xsd:string"
maxOccurs="20"/>
```

Setting a limit will cause erroneous or malicious messages to be rejected during the early validation stages of request processing, thereby minimizing the performance penalty for servicing such requests.

What Are We Measuring?

MAKING SENSE OF RESPONSE-TIME MEASUREMENTS

Response time is one of the major parameters that characterize Web service quality. Although measuring a time interval may seem straightforward, it may be trivial to decide where to place this measurement, how to interpret it, and how to estimate its accuracy. Most load-testing tools measure the elapsed time between sending a request and receiving a response from the server. Although measuring response time on the client side better reflects simulated user experience, it might depend heavily on the network environment, where the particular load-testing experiment is performed and can yield different results with different network configurations. For this reason, it is beneficial to monitor the server processing time as well. WebSphere Application Server offers such monitors (see Figure 2). Comparing server processing times with the virtual-user recorded response times gives a good estimate of the delays caused by the network.

ELIMINATING SOURCES OF INCORRECT RESULTS ON THE TEST CLIENT

Load-testing tools that emulate hundreds to thousands or more virtual users may cause heavy resource utilization – or even resource starvation – on test client machines. This can not only cause measurement errors, but also distort the stream of requests generated by the load test tool. For example, CPU saturation, paging, and long JVM garbage collection cycles (if the load test client is a Java application) may cause bursts of requests instead of the desired uniform – Poisson – or another request distribution that the load test tool is supposed to generate. To prevent anomalies and measurement distortions caused by resource overutilization on load-generating machines, you can monitor major performance parameters, such as CPU, memory, and network utilization. If your monitoring efforts reveal that resource utilization goes higher than 75%, add more hardware to avoid those anomalies.

Overutilization of load test client resources can also result in test failures, which can be erroneously interpreted as server faults. Figure 3 presents a load-test run that has errors caused by socket starvation on the client, which ran on a Windows XP machine. The load test ran without errors for approximately 140 seconds, then errors started occurring in bursts of up to 7 errors per second. Error details indicate network problems:

```
java.net.BindException: Address already in use:
connect
```

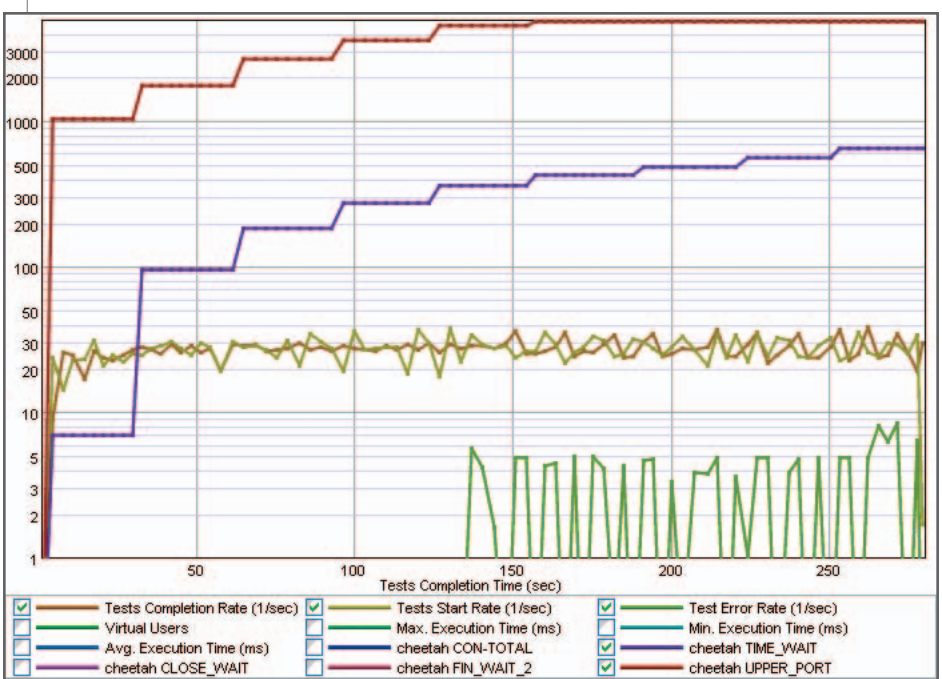


FIG 3: SOCKET STARVATION ON THE TEST CLIENT MACHINE

Adding TCP monitors on the test client revealed that the upper port on the client machine grows for the first half of the test, reaches the value of 5000, then stays at that value for the remainder of the test. The graph also shows a correlation between the upper TCP port reaching the maximum value and the beginning of errors. Further investigation revealed that, for Windows 2000 and higher, the operating system configuration restricts attempts to set up TCP connections from port 5000 and above. The upper port value is controlled by a Registry Key setting. After the Key value was increased to the maximum of 65534, the load test ran without errors.

Conclusion

By following performance-testing best practices, you can achieve more comprehensive coverage of possible performance issues that could arise during the production deployment of your Web service and increase its overall quality.

Best practices

Using the Thread Pool Funnel to Optimize WAS Performance

BY MICHAEL PALLOS

Michael S. Pallos is a senior solution architect at IBM. He has 19 years of software development, architecting, and business consulting experience. Michael holds two undergraduate degrees, an M.B.A., and is currently a PhD candidate at Walden University, focusing his research on computer security. MPallos@us.ibm.com

Organizations are increasingly using thread pools to enhance WebSphere Application Server performance by providing users with required information quickly without monopolizing resources required for other commands.

Regardless of the complexity of many J2EE applications, all that's important from the end user perspective, other than functionality, is processing time. The end user simply wants the requested HTML page to load quickly. Similarly, a business executive wants simple assurances that his or her organization is not losing customers due to slow Web site responses.

In the event that the response time is unacceptably slow, our competition is just a click away. As the old saying goes, "it is much easier to keep the customers you have than to get new ones." Successful Web site performance is key to a successful business and the Internet delivery channel.

The Pieces of a Successful J2EE Application

There are many J2EE application transaction components, some of which afford little or no control. Organizations, for example, have little control over the quality of customers' Internet connections to their Web sites or how the network path is routed in the Internet cloud. Fortunately, there are also many pieces of the Internet puzzle that organizations can manage and optimize, such as application servers, back-end integration technologies, Java applications, and hardware. A successful IT "ecosystem" requires that each infrastructure component be tuned to optimize the entire IT system and support business performance. Organizations, for example, must ensure that Java application execution streams (threads) provide users with required information quickly and without monopolizing resources required for other commands.

This article addresses best practices for harnessing thread pools in the application server environment, including the Hypertext Transfer Protocol (HTTP) listener, Web container (application container), the Object Request Broker (ORB) container (also referred to as the Enterprise JavaBean container), and the data source connection pool. It will also explore the notion of a thread pool funnel.

Thread Pool Overview

As users request Web site information, threads are required to process these requests. When focusing on the WebSphere Application Server (WAS) environment, applications requesting a thread are acquired from either the HTTP listener, Web container thread pool, ORB container thread pool, or the data source connection pool. The threads contained in these pools are the actual resources that service user requests. Therefore, tuning thread pools properly will enhance WAS performance and optimize the users' experiences.

The WAS administrative console is used to tune thread pools. Each thread pool is assigned a minimum and a maximum thread pool number. When WAS is launched, it will create the number of threads identified in the minimum thread pool number parameter. As application processing occurs, the system will generate additional threads as required, up to the maximum-number-of-threads parameter. It is critical to set the maximum number of threads for any specific queue or

"Successful Web site performance is key to a successful business and the Internet delivery channel."

command properly. If the system is constantly creating and destroying threads from the pool, system thrashing will occur, wasting valuable resources. When configuring a thread pool, it is important to remember that the "more is better" rule does not apply. Threads require a memory commitment and system resources. If the thread pool is configured to produce more threads than the system requires, valuable system resources are being denied to other resources. This type of configuration will burden the system and slow the application. Therefore, configuring thread pools accurately and harmoniously with each other is critical to optimal WebSphere performance.

Figure 1 displays an application request flow that requires back-end processing and illustrates the relationship among the thread pools as the user request is processed.

HTTP Listener

The HTTP listener is responsible for thread creation at the HTTP server level. Most of the processing that occurs here is static page serving, or HTTP post/GET pass commands to the back end. This is the first level of thread configuration that must be considered.

Web Container

The Web container is responsible for thread pool creation at the application server level. Most of the processing at this level includes servlet, JSP, EJB, dynamic page creation, and back-end pass-through processing. The Web container is the second level of thread pool configuration that must be configured.

ORB Container

The ORB container is responsible for thread pool creation at the object level. Most of the processing that occurs here includes the processing of non-Web–based clients. The ORB container is the third level of the thread pool configuration that must be configured.

Data Source

The data source level is responsible for creating the connection threads that are accessed from the database or "legacy" systems. These threads are the fourth level of configuration that must be addressed.

Thread Pool Funnel

When considering requests that are processed by the application, developers should consider Web architecture design and associated business processes. In many instances, more processing is required at the site's front-end systems than at the site's back-end systems. Because the number of requests requiring threads is greater at the front end of a Web site, the higher number of threads is required at layer one, and is then reduced for each subsequent layer. This configuration produces a funnel effect in which thread requirements are reduced as the administrator configures each of the four layers. If the thread pool amounts were equal, the system could overload at each pool location, thereby saturating the entire WAS environment.

Therefore, when setting thread pool minimum and maximum parameters for specific thread pools, developers are often able to create a thread pool funnel that will produce the best results for a site. The thread pool funnel is displayed in Figure 2.

Some of the requests to a Web site only require static page returns. These requests are merely services at the HTTP listener layer, and do not require Web container, ORB container, or data source threads. These requests simply make it to the HTTP listener layer (see layer one in Figure 2), and then provide the user with the static response.

In another scenario, the application may only have to process at the Web container layer, perhaps hitting a JSP and getting returned dynamic page content. When this type of processing occurs, the user request requires threads for processing at the HTTP listener and Web container layers. However, neither the ORB

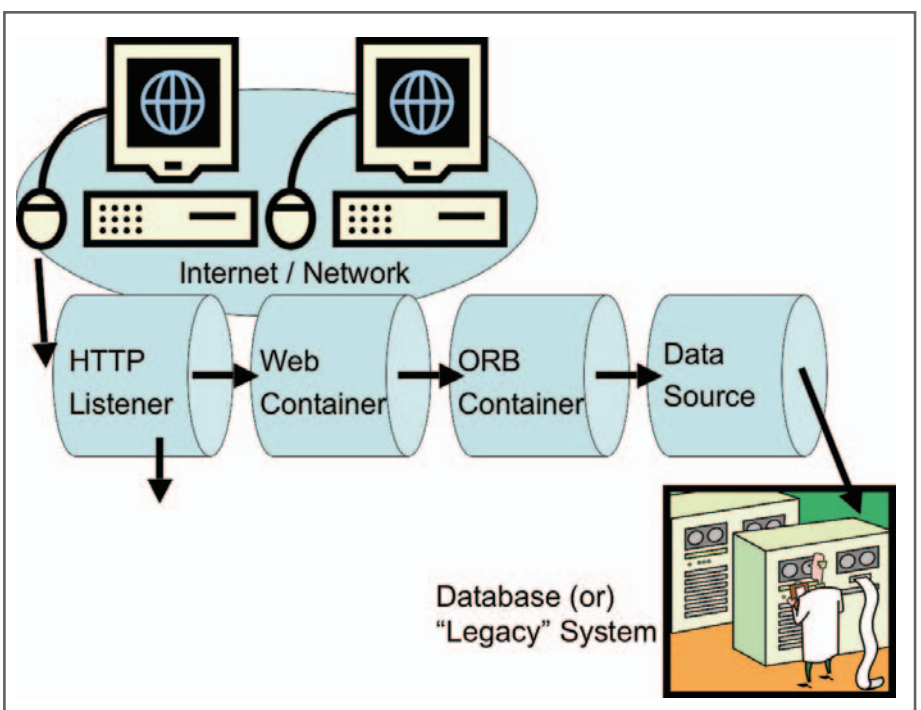


FIG 1: THREAD POOL QUEUES

"There are also many pieces of the Internet puzzle that organizations can manage and optimize"

container nor data source processing is required.

The third scenario is based on yet another type of inquiry, such as a non-Web Java application that would require processing. To service this request, a thread is acquired at the HTTP listener, Web container, and ORB container layers. In this case, processing at the data source layer was not required.

The final example, one which demands thread acquirement at all levels, includes a service rendered from the data source. A CICS transaction, for example, must be executed for the users' requests to be serviced via an EJB. In this final scenario, the users' requests require threads to be acquired at the HTTP listener, Web container, ORB container, and data source layers.

It is important to note that when a flood of requests occurs in a thread pool funnel, each request will wait its turn at the appropriate queue rather than overloading all queues. The thread pool funnel provides a mechanism to minimize the potential for requests to overload the application server, significantly slowing performance or even crashing the system.

Conclusion

The thread pool funnel concept was first introduced in the *Performances Analysis for Java* report last year by Joines, Willenborg, and Hygh. Although the thread pool funnel is not pertinent in all J2EE application architectures, it is an effective design pattern that can increase application performance significantly. I have seen many customers who have implemented this concept to reduce the end user's wait time, decrease the potential for application server saturation, and increase business productivity.

References

- Gunther, N. (2004). *How to Get Unbelievable Load Test Results*. Retrieved November 10, 2003, from Performance Dynamics Company Web site: www.teamquest.com/resources/gunther/load-test.shtml
- Joines, S.; Willenborg, R.; & Hygh, K. (Eds.). (2003). *Performance Analysis for Java*. Pearson Education, Inc.
- Willenborg, R. (2002). *Monitoring Performance with WebSphere*. Retrieved October 11, 2003, from the *e-Pro Magazine* Web site: www.e-promag.com/eparchive/index.cfm?fuseaction=viewarticle&ContentID=1492&websiteid=
- Zhang, Y.; Liu, A.; & Qu, W. (2003). *Software Architecture Design of an Autonomic System*. Retrieved November 10, 2003, from the University of Sydney Web Site: http://mercury.it.swin.edu.au/ctg/AWSA04/Papers/zhang.pdf

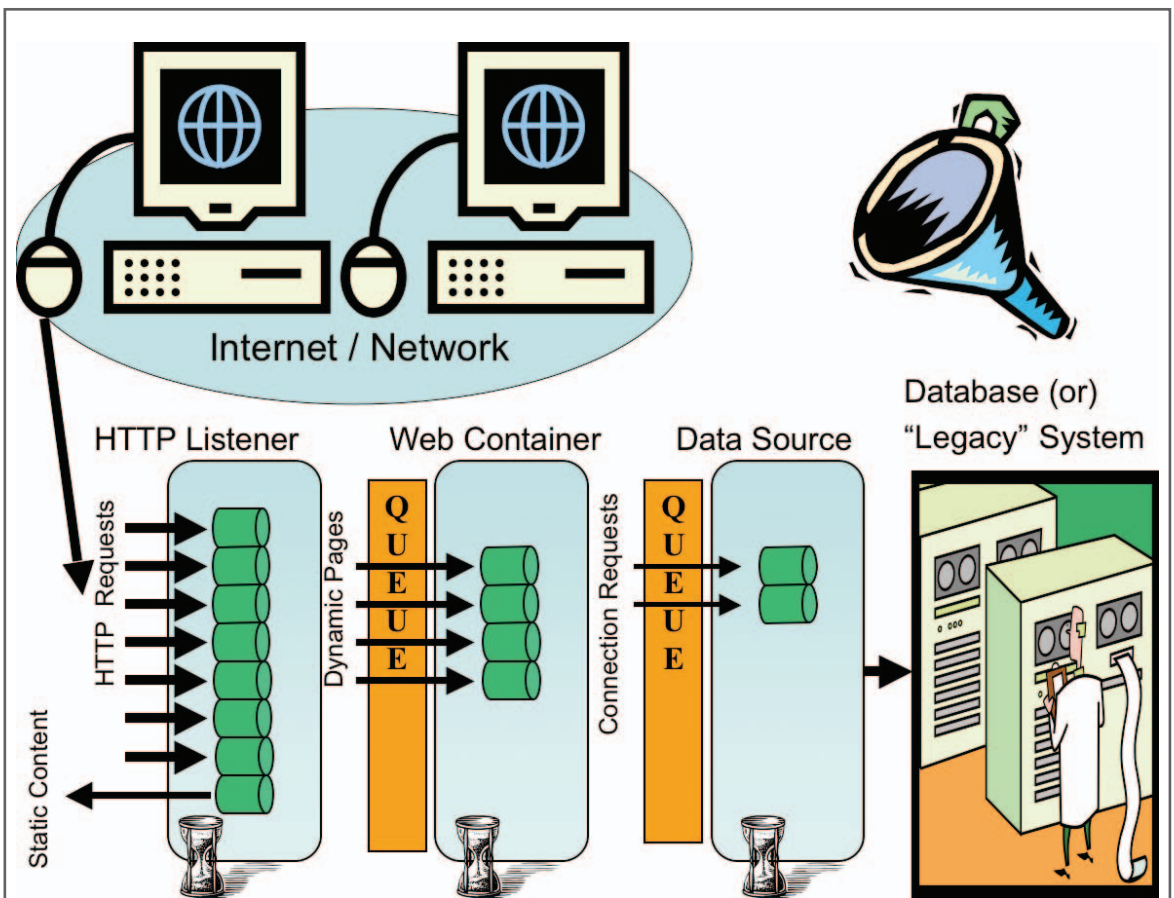


FIG 2: THREAD POOL FUNNEL

The power of data analysis in a portal architecture

A New View on Business Intelligence

BY JOSEPH R. MARQUES

Joseph R. Marques is a member of Prolifics' WebSphere Consulting Division – a specialized team of experts on which IBM calls to deliver training, mentoring, and development services, as well as to solve the toughest of their customer's challenges. Specializing and certified in Java and WebSphere architecture and best practices, application development and deployment, and portal development, Joseph is responsible for delivering distributed, J2EE, WebSphere solutions to clients worldwide. jmarques@prolifics.com

With better technologies, new architectures, and innovative ways of thinking about old problems, there are new applications for business intelligence and data analysis. I am talking about the power of the intelligent portal.

But what do portals have to do with data analysis? Everything. A portal, by bringing together such a wide range of applications, services, and even entire businesses, not only provides a solid integration platform, but by doing so, enables business intelligence. Data that was once separate, awkwardly related, and difficult to format can now be brought together under a common umbrella to communicate and be interpreted in new and innovative ways. A remote Web service, a legacy mainframe application, an old COM project – all can now talk to each other within a portal, which makes one think: what are they talking about? My suggestion: ask the logs.

Integration and Its Effect on Data Analysis

In the days of independent and uncommunicative applications, it was possible to have a complete view into the usage of an individual system – you could track that a user logged in and clicked here, waited for 15 seconds, then went to this page, then checked so-and-so data, and finally logged out.

However, to improve interoperability, companies adopted custom solutions to glue together disparate applications – where one program would speak directly to another. The problem with point-to-point integration is that it becomes increasingly difficult to understand how a single user is sharing the data across these applications. If each application has its own form of logging, it could be a veritable nightmare to merge those two pieces of data, much less analyze them accurately. Even if both programs were assumed to use best practices and logged events in universal time (UTC), slight differences in the clocks between machines could throw the interpretation of the data way off. Did the user click on this link or download that spreadsheet first?

Newer integration strategies, such as service-oriented architecture (SOA), provide for applications of any language on any platform to talk to each other over a central bus. With information from different applications flowing through the same central repository, it becomes easier to see and understand system usage.

Portal technologies too integrate applications through a central location – but at the presentation layer – offering a view into the user's actions. This ensures that events for a user are given a well-defined ordering to eliminate any confusion. A small application that interprets those records containing events collected at the presentation layer would help to cleanly tie together and accurately portray site activity and usage. Analysts and planners would be able to bring data back together and understand how it is being used, by whom, and toward what meaningful end.

Creating a Superior Customer Experience

When companies roll out a general portal solution, whether or not it was built on an SOA, the initial navigation is often awkward. Poor design or careless placement of functionality on pages can force a user to spend too much time on one page, while missing better services offered on another. If a robust logging solution was in place, it would enable organizations to quickly see these issues and take action. For example, consider an online bank. A report can be generated that tells site designers and administrators that most new users eventually end up going to the loan application page, but have to sift through four or five links before they get to it. Other reports can show that some people become new users and browse around the site for a while, but don't sign up for any service. They could also show that those same users later come back to the site, still navigate haphazardly, but eventually find and apply for a loan. Clearly, these users would have appreciated if that functional-

ity had been more visible. Thanks to log analysis, the organization can make the accurate assumption that this often requested service should be moved closer to the entry point of the site so it is more clearly seen. Attention to details such as these goes a long way toward improving the end user's experience and capturing more business.

Adapting the Customer Experience to the Customer

Having a different, customized face to the same data isn't something terribly new – it's called personalization. Imagine a company's reputation if it was able to change the user experience to custom fit the browsing habits of each user. Suppose an online newspaper wants to provide the latest breaking news it possibly can, but in a personalized way to its customers. Now, instead of just posting news, they need to understand how that news is being used. By performing data analysis on the information gathered from the portal logs, you can learn what types of news a person likes to see the most, and customize his or her homepage to show that. Rather than having the users manually change or customize preferences, why not automatically present the data they are seeking? By analyzing aggregated data about each individual's use of a site, it is possible to present what the user wants, without him or her asking for it or needing to choose it. As a user's browsing habits change over time, the site can also adapt. For instance, if a person likes sports, it wouldn't be anything special to just give them the sports page when he enters the site. It would be more impressive if you record that, on their last visit they searched for the final standings of the U.S. Open. Then, when they returned to the site, they were immediately shown highlights of the latest tennis tournament, along with its standings. The customized data you provide need not be restricted to the last couple of visits alone. As seasons change, so do the sports offerings. So, you should have rules in place to accommodate both the user's short-term interests and long-term tastes. The site would be adaptive to each user's personal viewing habits, providing for a truly organic experience.

Adapting the Portal Interface to Improve Your Business Model

Good site management should not only encompass how the site looks and what functionality should be placed on which pages, but should also encompass active log management. That's where all of the knowledge is; that's where smart decisions can be made. For instance,

"Portal technologies are in a powerful position to provide cutting-edge statistics"

one e-commerce Web site happens to be a conglomeration of different, smaller companies purchased over the years. They currently have a portal solution that opens up each of the proprietary inventories. Thanks to portals, you can now record user purchases across product domains. If you have a particularly strong showing one month for a certain combination of purchases, you can always offer exclusive discounts that help customers realize that you are aware of their buying habits and needs. Also, if the logs show that users are shopping much more frequently at certain parts of your site than others during a given time period, you could pitch to your ad investors that those spots on the Web site are more valuable and thus cost more to advertise on. As long as you have that user data to back up your claims, and as long as you're reasonable with your price, the ad companies aren't going to turn down your offer. It's a win-win-win situation: for them, it means more potential users clicking on their ads; for you, it means more revenue; for your customers, it means providing contextually accurate advertising to improve their overall user experience – take that, Google!

Becoming Predictive to Improve Your Business Model

New software and services are born in two major forms: assumption and demand. With assumption, there is risk because without really knowing precisely what your customers want, you try your best to presume what services they could need and what services they would use. From time to time, this train of thought will pay off, but not with as much return as anticipated. In the demand world, one will meticulously review metrics across multiple applications to try and deduce how customers are combining separate services for their own use. Some companies will even ask users to rate the site or take a survey to acquire feedback.

With the data analysis of portal logs in place, none of that hassle is needed anymore. As you record a user's steps through the site, you gain intimate knowledge of how he or she is using your data – data not just within one application, but data that crosses boundaries. By understanding the flow of a customer's visit to a site, it is possible to have a vast repository of extremely accurate statistics. You can extrapolate from this knowledge ideas to help companies more quickly predict exactly what kind of new service customers are looking for and preemptively provide it to them. So custom-developed analysis tools that interpret these new types of logs not only help to serve as a record and guide to the newly linked data being accessed, but also as a knowledge-based and rule-based foundation to the intelligent portal.

The Intelligent Portal

Portal technologies are in a powerful position to provide cutting-edge statistics to managers, planners, and technologists alike. Managers want to be able to effectively gather meaningful feedback from a system so as to make better and more informed decisions. Planners need to know how a business process or service is being utilized so as to provide better evolutionary interfaces to make working with the data easier. Technologists want to know what data is being requested so that they can brainstorm and come up with the next best service before a customer has even thought that he or she might want it. All of these ideas are viable; all of these things can be realized.

When getting started, I recommend using a blend of data analysis tools along with a custom implementation. IBM's more advanced portal technology, called IBM WebSphere Portal Extend, comes with tools and built-in functionality designed to empower your data analysis process. Included in the edition is a tool called Site Analyzer that captures, stores, and reports on-site usage and content relevancy. WebSphere Portal Extend also provides rules-based personalization, which enables business managers to define business rules that personalize content for users and groups. The software's collaborative filtering uses recommendation engines that utilize advanced statistical models and other forms of intelligent software to extract trends from the behavior of portal users. This approach adapts to changes in visitor interest without business rules. Finally, to make the analysis process truly effective for your organization, it is key to customize your tooling and techniques. If designed and implemented well, the vital business intelligence will enable you to build on your customer satisfaction as well as on your business model, making the return on your investment immediately worthwhile.

Through all of these different mindsets, when it comes to the potential for log analysis in a portal environment, what do they all have in common? Each stresses that the world today – from the user's or customer's point of view – should be more predictive than reactive. The less you have to ask your target audience about the things they want, the more money you'll save, and the less effort you'll expend. The key is to rapidly deliver services that are simultaneously personalized and contextually relevant.

Forget something?

Post-launch is NOT the time to be verifying web applications.

The wild blue yonder of operational monitoring and management is extremely unforgiving. Which means that going live with the monitoring software you used in development is a great way to go dead—quickly! You simply can't support operations if your staff is drowning in details provided by development profiling tools and debuggers. **Let NetIQ cover your apps...with AppManager.**

AppManager—the industry's easiest-to-use Systems Management suite—is a proven management system for monitoring J2EE application servers, databases, operating systems and even end-user response time. NetIQ's AppManager monitors ALL application components—not just your server. **NetIQ. Nobody does UNIX better. Nobody.**

Visit us at www.netiq.com/solutions/web to learn how we can help you address the challenges of your operational monitoring and management.

netiq
Work Smarter.

A brief comparison

JavaServer Faces (JSF) vs Struts

BY ROLAND BARCIA

Roland Barcia is a consulting IT specialist for IBM Software Services for WebSphere in the New York/New Jersey Metro area. He is the author of one of the most popular article series on the developerWorks WebSphere site, www-106.ibm.com/developerworks/websphere/techjournal/0401_barcia/barcia.html, and is also a coauthor of ***IBM WebSphere: Deployment and Advanced Configuration*** (www.phptr.com/title/0131468626). You can find information about him at http://web.njit.edu/~rb54
barcia@us.ibm.com

My JSF article series and Meet the Experts appearance on IBM developerWorks (see References) received a lot of feedback.

I would have to say the most common question or feedback came along the lines of comparing Struts to JSF. I thought it would be a good idea have to compare JSF to Struts by evaluating various features that an application architect would look for in a Web application framework. This article will compare specific features. Those on which I will focus include:

- Maturity
- Controller flexibility/event handling
- Navigation
- Page development
- Integration
- Extensibility

Certainly, there are other areas in which you might want to do a comparison, such as performance, but I'll focus on the set I just mentioned. I'll also spend more time on the Controller and Navigation sections because they are the heart of the frameworks. Performance of JSF is specific to the vendor implementation, and I always encourage people to perform their own performance tests against their own set of requirements because there are too many factors that can affect performance. A performance evaluation would be unfair. Other areas, such as page layout, validation, and exception handling, were also left out in the interest of saving space.

Maturity

Struts has been around for a few years and has the edge on maturity. I know of several successful production systems that were built using the Struts framework. One example is the WebSphere Application Server Web-based administrative console. JavaServer Faces (JSF), however, has been in draft for two years. Several companies, including IBM, as well as the creator of Struts, Craig McClanahan, have contributed to the creation of JSF during that time. Nonetheless, it will take some time to see a few systems deployed.

Struts definitely has the edge in this category. With JSF, however, you can rely on different levels of support depending on which implementation you choose. For example, the JSF framework inside WebSphere Studio comes with IBM support.

Controller Flexibility/Event Handling

One of the major goals of Struts was to implement a framework that utilized Sun's Model 2 framework and reduced the common and often repetitive tasks in servlet and JSP development. The heart of Struts is the Controller. Struts uses the Front Controller Pattern and the Command Pattern. A single servlet takes a request, translates HTTP parameters into a Java ActionForm, and passes the ActionForm into a Struts Action class, which is a command. The URI denotes which Action class to go to. The Struts framework has a single event handler for the HTTP request. Once the request is met, the Action returns the result back to the front controller, which then uses it to choose where to navigate to next. The interaction is demonstrated in Figure 1.

JSF uses the Page Controller Pattern. Although there is a single servlet that each faces request goes through, the job of the servlet is to receive a faces page with components. It will then fire off events for each component and render the components using a render toolkit. The components can also be bound to data from the model. The faces life cycle is illustrated in Figure 2.

JSF is the winner in this area because it adds many benefits of a front controller, but at the same time gives you the flexibility of the Page Controller. JSF can have several event handlers on a page, whereas Struts is geared to one event per request. In addition, with Struts your ActionForms have to extend Struts classes, creating another layer of tedious coding or bad design by forcing your model to be ActionForms. JSF, on the other hand, gives developers the ability to hook into the model without breaking layering. In other words, the model is still unaware of JSF.

Navigation

Navigation is a key feature of both Struts and JSF. Both frameworks have a declarative navigation model and define navigation using rules inside their XML configuration file. There are two types of navigation: static navigation – when one page flows directly to the next; and dynamic navigation – when some action or logic determines which page to go to.

Both JSF and Struts currently support both types of navigation.

STRUTS

Struts uses the notion of forwards to define navigation. Based on some string, the Struts framework decides which JSP to forward to and render. You can define a forward by creating an Action as shown in the snippet below.

```
<action path="/myForward"
  forward="/target.jsp">
  </action>
```

Struts supports dynamic forwarding by defining a forward specifically on an Action definition. Struts allows an Action to have multiple forwards.

```
<action-mappings>
  <action name="myForm" path="/
  myAction" scope="request"
  type="strutsnav.actions.
  MyAction">
   <forward name="success"
     path="./target.
  jsp">
   </forward>
   <forward name="error" path="./
error.jsp">
   </forward>

  </action>
 </action-mappings>
```

Developers can then programmatically choose which forward to return.

```
public ActionForward execute(
ActionMapping mapping,
ActionForm form,
HttpServletRequest request,
HttpServletResponse response)
throws Exception {

ActionErrors errors = new
 ActionErrors();
ActionForward forward = new
 ActionForward(); // return
  value
MyForm myForm = (MyForm) form;
  try {

// do something here

  } catch (Exception e) {

// Report the error using the
appropriate name and ID.
   errors.add("name", new
    ActionError("id"));
   forward = mapping.
```

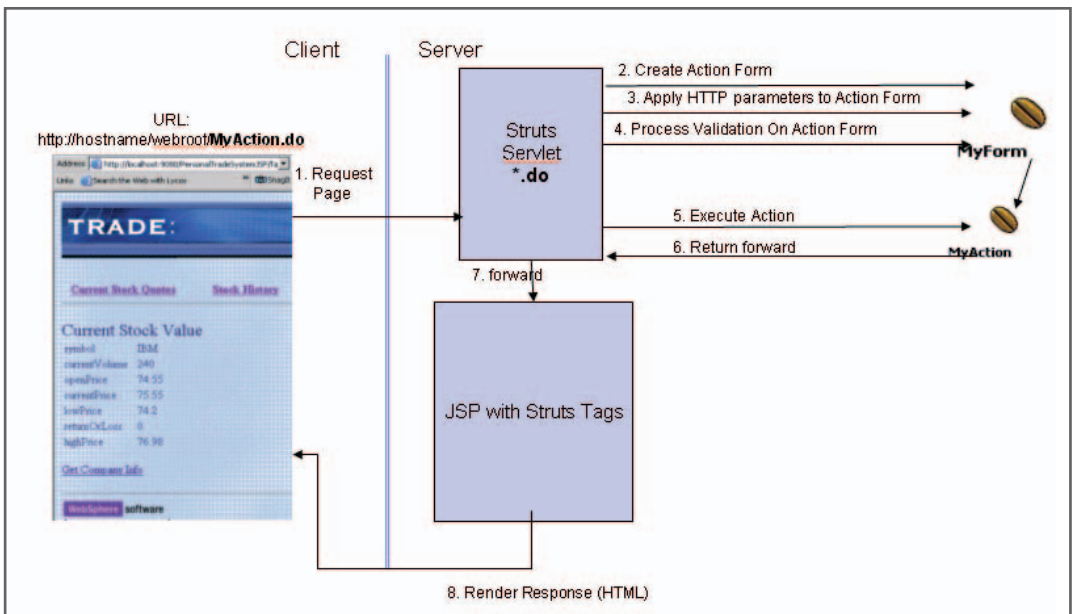


FIG 1: STRUTS LIFE CYCLE

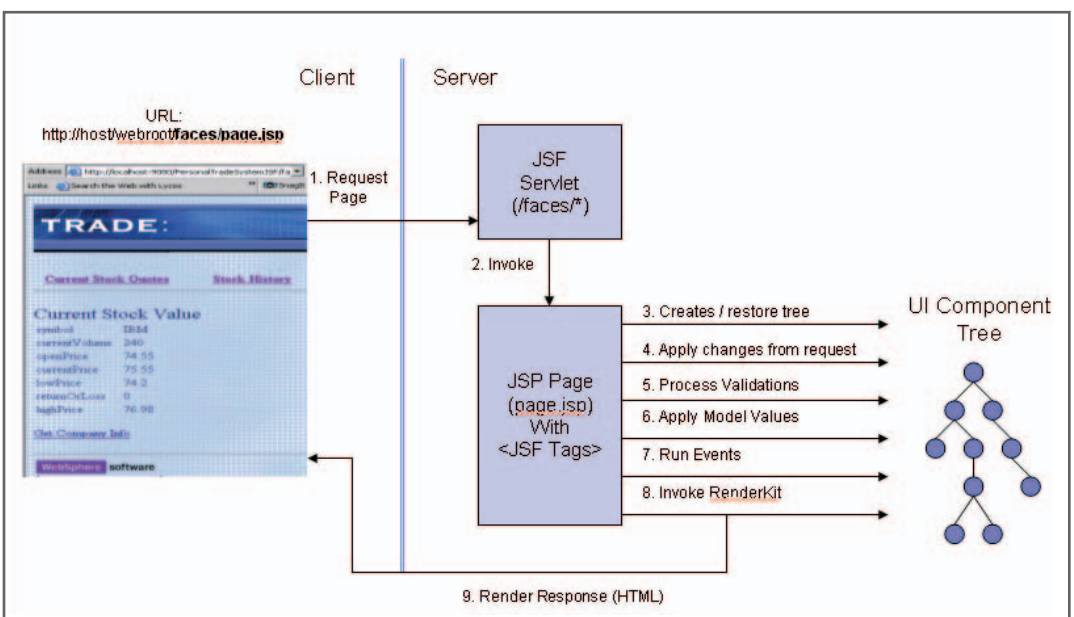


FIG 2: JSF LIFE CYCLE

> "Struts was designed to be model neutral, so there are no special hooks into a model layer."

```
    findForward("success");
return (forward);
  }

  forward = mapping.
   findForward("success");
    return (forward);

 }
```

JSF STATIC NAVIGATION

JSF supports navigation by defining navigation rules in the faces configuration file. The following code shows a navigation rule defining how one page goes to the next.

```
<navigation-rule>
  <from-view-id>/FromPage.jsp</
   from-view-id>
  <navigation-case>
   <from-outcome>success</from-
    outcome>
   <to-view-id>/ToPage.jsp</to-
view-id>
  </navigation-case>
 </navigation-rule>
```

However, unlike Struts, JSF navigation is applied on the page level and can be action-independent. The action is hard-coded into the component, allowing for finer-grain control on the page. You can have various components on the page define different actions sharing the same navigation rule.

```
<hx:commandExButton type="submit"
value="Submit"
styleClass="commandExButton"
id="button1" action="success" />
```

JSF also supports dynamic navigation by allowing components to go to an action handler.

```
<hx:commandExButton type="submit"
value="Submit"
styleClass="commandExButton"
id="button1" action="#{pc_
FromPage.doButton1Action}" />
```

Developers can then code action handlers on any class to make the dynamic navigation decision.

```
public String doButton1Action() {
  return "success";
 }
```

Even though navigation rules don't need to specify the action in order to support dynamic navigation, JSF allows you to define the action on the navigation rule if you so choose. This allows you to force a specific navigation rule to go through an action.

```
<navigation-rule>
  <from-view-id>/FromPage.jsp</
   from-view-id>
  <navigation-case>
   <from-action>#{pc_FromPage.
    doButton1Action}</from-action>
   <from-outcome>success</from-
    outcome>
   <to-view-id>/ToPage.jsp</to-
    view-id>
  </navigation-case>
 </navigation-rule>
```

Both Struts and JSF are pretty flexible from a navigation stand point, but JSF allows for a more flexible approach and a better design because the navigation rule is decoupled from the Action. Struts forces you to hook into an action, either by a dummy URI or an Action class. In addition, it is easier in JSF to have one page with various navigation rules without having to code a lot of if-else logic.

Page Development

JSF was built with a component model in mind to allow tool developers to support RAD development. Struts had no such vision. Although the Struts framework provides custom libraries to hook into ActionForms and offers some helper utilities, it is geared toward a JSP- and HTTP-centric approach. SF provides the ability to build components from a variety of view technologies and does it in such a way to be toolable. JSF, therefore, is the winner in this area.

Integration

Struts was designed to be model neutral, so there are no special hooks into a model layer. There are a few reflection-based copy utilities, but that's it. Usually, page data must be moved from an ActionForm into another Model input formating, requiring manual coding. The ActionForm class provides an extra layer of tedious coding and state transition.

JSF, on the other hand, hides the details of any data inside the component tree. Rich components such as data grids can be bound to any Java class. This allows powerful RAD development, such as the combination of JSF and SDO. I will discuss this further in future articles.

Extensibility

Both Struts and JSF provide opportunities to extend the frame-

work to meet expanding requirements. The main hook for Struts is a RequestProcessor class that has various callback methods throughout the life cycle of a request. A developer can extend this class to replace or enhance the framework.

JSF provides equivalent functionality by allowing you to extend special life-cycle interfaces. In addition, JSF totally decouples the render phase from the controller, allowing developers to provide their own render toolkits for building custom components. This is one of the powerful features in JSF that Struts does not provide. JSF clearly has the advantage in this area.

Conclusion

In general, JSF is a much more flexible framework, but this is no accident. Struts is a sturdy framework and works well. JSF was actually able to learn a great deal from Struts projects. I see JSF becoming a dominant framework because of its flexible controller and navigation. Furthermore, JSF is built with integration and extensibility in mind. If you are starting a new project today, you'd have to consider many factors. If you have an aggressive schedule with not much time to deal with evaluating different vendors or dealing with support for new JSF implementations, Struts may be the way to go. But from a strategic direction and programming model, JSF should be the target of new applications. I encourage developers to take time to learn JSF and begin using them for new projects. In addition, I would consider choosing JSF vendors based on component set and RAD tools. JSF isn't easier than Struts when developing by hand, but using a RAD JSF tool like WebSphere Studio can greatly increase your productivity.

References

- *Developing JSF Applications Using WebSphere Studio:* www-106.ibm.com/developerworks/websphere/techjournal/0401_barcia/barcia.html
- *Developing JavaServer Faces Portlets Using WebSphere Studio:* www-106.ibm.com/developerworks/websphere/techjournal/0406_barcia/0406_barcia.html
- *Meet the Experts: Roland Barcia on JSF and JMS:* www-106.ibm.com/developerworks/websphere/library/techarticles/0407_barcia/0407_barcia.html
- *JSF Central:* www.jsfcentral.com
- *IBM WebSphere – Deployment and Advanced Configuration*: www.amazon.com/exec/obidos/tg/detail/-/0131468626/qid=1092062894/sr=8-1/ref=sr_8_xs_ap_i1_xgl14/103-2058110-8899867?v=glance&s=books&n=507846

How to create a true security token timeout in WebSphere Application Server 5.1

Making Single Sign-on Work

BY JON FORCK

Jonathan Forck is a J2EE developer who has done key work on several online banking applications developed for a group of banks. Jonathan has written his own implementation of the security interface for WAS 5.1 and is currently working on a plug-in to the Eclipse Framework that will be a clear-text TCP/IP monitor for SSL-encrypted Web sites.
jforck@gmail.com

Single sign-on was developed with the idea of making life more convenient for the user, but you may have to put a little more work into it to make that convenience a reality.

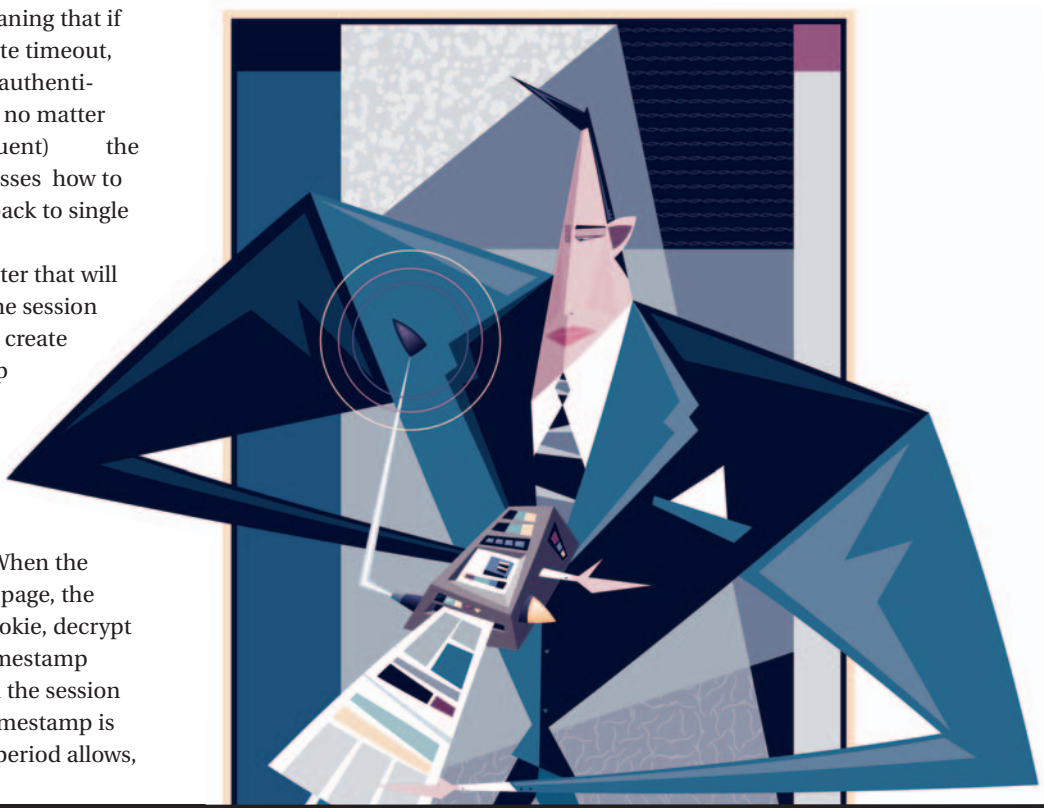
WAS 5.1 offers a single sign-on option, meaning that you can have one application server authenticate a user, then send the user to a separate application server without requiring the user to sign on again. However, if you enable this feature your security system will no longer have a true timeout. The only option is to set a "lifetime" timeout – meaning that if your app has a 15-minute timeout, the user will have to be authenticated every 15 minutes, no matter how frequent (or infrequent) the usage. This article discusses how to work around this drawback to single sign-on.

First, implement a filter that will log the user out when the session times out. The filter will create an encrypted timestamp and place it into a session cookie. When the user requests an app entry page, the filter will place the timestamp cookie. When the user requests any other page, the filter will retrieve the cookie, decrypt it, and make sure the timestamp has been created within the session timeout period. If the timestamp is older than the timeout period allows, the user is logged out and then redirected to the login page. This works great in a portal that delegates access to multiple secure applications.

Implementation

1. Define the filter in web.xml to receive a parameter that will be a passphrase for the encryption, and a parameter consisting of a comma separated list of entry pages. These pages will only set the cookie; they will not check the timestamp. This will generally be the servlet that places the user's info in the session (see Listing 1).
2. Create an encryption/decryption processor. Use passphrase encryption in the cookie so you can create the original token in the portal page and then pass it on to the various other applications in your portal easily.
3. Create a filter to handle the process (see Listing 2).
4. Add the methods to handle setting the cookie, getting the cookie, validating the cookie, and logging the user out (see Listing 3).

You can now set the server's security timeout to a lengthy period and still have a true security timeout.



LISTING 1

```
<filter>
 <filter-name>SessionTimeoutFilter</filter-name>
 <display-name>Session Timeout Filter</display-name>
 <filter-class>
  com.syscon.filters.SessionTimeoutFilter
 </filter-class>
 <init-param>
  <param-name>passphrase</param-name>
  <param-value>WJ Rocks</param-value>
 </init-param>
 <init-param>
  <param-name>entryURLs</param-name>
  <param-value>/j_security_check</param-value>
 </init-param>
</filter>
```

LISTING 2

```
package com.syscon.filters;

import java.io.*;
import java.util.*;
import java.text.*;
import javax.servlet.*;
import javax.servlet.http.*;

import com.syscon.encryption.PassphraseEncryptionEngine;

public class SessionTimeoutFilter implements Filter {
  HashMap entryURLs;
  private static final String DATE_FORMAT =
"ZddMMyyHHmmssSSSS";
  private static final String COOKIE_NAME =
"SYSConTimeout";
  PassphraseEncryptionEngine encryptionEngine;
  SimpleDateFormat formater;

  public SessionTimeoutFilter() {
    super();
  }

  public void init(FilterConfig config) throws
   ServletException {
    String passphrase = config.getInitParameter("passphr
     ase");
    if (passphrase == null) {
      throw new ServletException("Must provide a pass
       phrase parameter");
    }
    encryptionEngine = new PassphraseEncryptionEngine(pas
     sphrase);
    String entryPoints = config.getInitParameter("entryU
     RLs");
    if (entryPoints != null) {
      StringTokenizer st = new StringTokenizer(entryPoin
       ts, ",");
      String url = "";
      while (st.hasMoreTokens()) {
        url = st.nextToken();
        entryURLs.put(url, null);
      }
    }
    formater = new SimpleDateFormat(DATE_FORMAT);
  }

  public void doFilter(ServletRequest request,
ServletResponse response,
      FilterChain chain) throws IOException,
ServletException {
    HttpServletRequest req = (HttpServletRequest) request;
    HttpServletResponse res = (HttpServletResponse)
response;
    if (!entryURLs.containsKey(req.getServletPath())) {
      // Get the cookie out of the session.
      Cookie oldCookie = getCookie(req);
      if (!isCookieValid(oldCookie, req, res)) {
        logout(req, res);
        return;
      }
    }
    // Set the new cookie
    setCookie(res);
    chain.doFilter(request, response);
  }

  ...
}
```

LISTING 3

```
    private boolean isCookieValid(Cookie oldCookie,
HttpServletRequest req, HttpServletResponse res) {
    try {
      // Get the value of the cookie.
      String cookieVal = oldCookie.getValue();
      // Decrypt value
      String timestamp = encryptionEngine.
       decrypt(cookieVal);
      Date oldDate = formater.parse(timestamp);
      Date expireDate = new Date();
      Calendar cal = Calendar.getInstance();
      cal.setTime(expireDate);
      cal
          .add(Calendar.SECOND, req.getSession().getMaxI
            nactiveInterval()
              * -1);
      expireDate = cal.getTime();
      // If token expired, log the user out.
      if (oldDate.before(expireDate)) {
        return false;
      }
    } catch (Exception e) {
      System.err.println(e);
      return false;
    }
    return true;
  }

  private Cookie getCookie(HttpServletRequest request)
throws ServletException {
    Cookie[] cookies = request.getCookies();
    if (cookies != null) {
      for (int i = 0; i < cookies.length; i++) {
        if (COOKIE_NAME.equals(cookies[i].getName())) {
          return cookies[i];
        }
      }
    } else {
      throw new ServletException("Cookie not found");
    }
  }

  private void setCookie(HttpServletResponse res) {
    Date newDate = new Date();
    String formatedDate = formater.format(newDate);
    String cypherDate = encryptionEngine.
encrypt(formatedDate);
    Cookie sessionCookie = new Cookie(COOKIE_NAME, cypher-
Date);
    sessionCookie.setPath("/");
    res.addCookie(sessionCookie);
    return;
  }

  private void logout(HttpServletRequest req,
HttpServletResponse res) {
    RequestDispatcher rd = req
        .getRequestDispatcher("/ibm_security_logout?logou
tExitPage=/");
    rd.forward(req, res);
    return;
  }
```

Using IT to improve productivity and increase revenue

Real-Time Insight for the Real World

BY DAVID LAPP

No industry has done more to streamline and improve core business processes with IT than the financial markets industry. Its business transactions have been conducted electronically for decades. Electronic communication is ingrained in the industry's institutions. In fact, new requirements for technology have often emerged first in the financial markets industry.

Yet with the volatility and risk inherent in their business, as well as increasing volume, globalization, and regulation, capital markets companies continue to look to IT to help them improve productivity and gain new revenue. They do so because they need to attain more real-time insight for trading, to sell complex products with downsized sales staff to an ever-expanding range of customers, and to coordinate routine high-value transactions across multiple market infrastructures.

As a result, financial markets companies want to standardize their proprietary networks, which are often fragmented by country, type of financial instrument, etc., using the Internet and emerging technologies. They want to use simple tools to help them get a better handle on the exchange of information in the trading process.

IBM has long been an IT provider to the financial markets industry, with 100% of Fortune 500 securities companies using IBM WebSphere and 96 of the top 100 financial institutions using IBM Tivoli. "Even though this is a highly competitive industry, IBM is still the vendor to beat," said David Lapp, Financial Markets Segment manager for IBM Software. "Our messaging middleware is nearly ubiquitous, and customers that choose IBM for other functions like business process monitoring and workflow get added synergies as a result."

This experience is the foundation of IBM's Industry Middleware Solutions for Financial Markets, which are designed to address the industry's most pressing challenges. The solutions are part of IBM's effort to deliver middleware software based on customer preference for buying products that address their industry-specific projects. Each solution contains functions from IBM's WebSphere, Lotus, Tivoli, DB2, and Rational middleware brands and IBM's industry-specific products, combined with applications from independent software vendors (ISVs) and industry-expert services.

There are five IBM middleware solutions for financial markets companies. They are modular, which allows companies to implement pieces at their own pace, and are based on open standards that support platform independence.

"Our solutions are intended to help these companies better automate and link their trading processes to improve productivity, but also to exploit new business opportunities more quickly," said Bob Covington, lead architect of Global Financial Markets for IBM. "They're also designed to help deal with infrastructure problems with trading or connecting to counter-parties."

IBM's Middleware Solution for Trade and Order Management provides the infrastructure software capability to instantaneously alert traders and systems to trading opportunities. It does so by continuously analyzing market data and integrating the real-time data with enterprise information. This complex data analysis allows capital markets firms to pursue sophisticated trading strategies. It can be used to increase returns on proprietary trading or enhance services for sophisticated institutional investors. The solution helps capital markets firms exploit arbitrage opportunities and eliminate trade-offs between speed and robustness of analysis.

IBM's Middleware Solution for Front Office Insight also helps financial markets companies serve increasingly sophisticated customers such as hedge funds and get full value from their proprietary research. Capital markets companies are grappling with how to serve these customers, often with fewer resourc-

es. They also have to deal with house research being perceived as low-value by the buy-side, with duplicative infrastructure due to fragmented product information.

IBM's Middleware Solution for Front Office Insight helps firms improve customer service – despite having fewer resources and more complex products – by aggregating customer, product, and market data in customer-facing business processes. It does so by extending applications to employee and customer devices, thereby improving productivity and increasing sales. The solution also helps firms differentiate services through the efficient use of research and proprietary data.

Investment Data Technologies, a provider of software systems and managed investment data, is one of only three companies in Australia that obtains managed funds data directly from fund managers. It serves the country's top banks, insurance companies, financial advisor groups, and fund managers. Investment Data Technologies needed to provide its financial advisors with an effective and efficient way to access managed investment information and upload transaction records.

Working with IBM, Investment Data Technologies developed a secure, Java-based data exchange mechanism. As a result, it has a more reliable, convenient, secure, and less costly information exchange. In addition, the solution has made it easier for Investment Data Technologies to expand its business geographically and to larger customers.

Financial markets companies also must manage processes beyond their walls on a daily basis, which can be rather challenging. Variations among the intermediaries in a transaction cause exponential growth in complexity. Manual steps introduce errors. Current processes rely on point-to-point integration and changing standards require constant attention. The result is the duplication of applications, processes, functions, and data warehouses; an inability to automate and monitor risk management activities; and the need to make multiple settlements, statements, customer inquiries, histories, and adjustments.

IBM's Middleware Solution for Financial Information Interchange helps improve connections and efficiencies by managing critical business processes involving exchanges, settlement agencies, outsourced service providers, and more. It uses financial messaging as the backbone to provide more efficient connectivity to applications and infrastructures. It allows migration to a single middleware platform while protecting investments and automating manual processes. This reduces costs, improves visibility in the processes, and increases reliance on industry standards.

HSBC Trinkaus & Burkhardt KGaA offers customers in Germany and Luxembourg an array of financial services, including private and commercial banking, corporate investments, and portfolio management. HSBC Trinkaus & Burkhardt also offers interest and currency management. The availability of mission-critical applications is crucial to their productivity and success. Knowing this, systems administrators at HSBC Trinkaus & Burkhardt made infrastructure management a top priority.

The use of IBM software has made enterprise systems management much easier, improving staff productivity and providing them with more time to plan new projects. It also has resulted in significant cost savings and helped to increase the availability of systems

WebSphere Journal Coming Next Month...

Creating a Healthy, Optimized WebSphere MQ Workflow Environment
25 questions that guide the way

Delivering Applications on Demand in SOA Environments
Fulfilling the promise

An Interview with Rick Thomas
Talking to the program director for CICS Tools Marketing

The Importance of Understanding Your Customer's Industry
The holistic perspective

and applications.

Financial markets companies also need to improve their trade execution processes, which are increasingly complex, require too much manual intervention, and impact the bottom line with high trade failure rates.

IBM's Middleware Solution for Post-Execution Process Integration helps reduce failed trade rates and operational costs as well as improve automation and efficiency. It does so by integrating front-, middle-, and back-office applications, processes, and workflows and automating post-execution processes. It helps companies detect trade failures earlier, decrease manual errors, improve reference data quality, and gain the ability to simulate new processes before implementation.

Credit Lyonnais provides corporate and investment banking as well as asset management to more than six million clients worldwide. It also offers retail banking services in France. Credit Lyonnais needed to consolidate a matrix of redundant, globally distributed systems to enable straight-through processing of equity trades to improve its responsiveness to changes in market demand. It also needed to meet rising customer expectations and business volumes as well as be able to develop and introduce new services. To do so, Credit Lyonnais needed to enhance the speed, security, and quality of its equity capital operations.

Working with IBM, Credit Lyonnais created an integrated operating environment and redefined and automated key business processes. As a result, Credit Lyonnais has become more focused and responsive to financial market changes. Specific benefits include a 20% cost reduction in business and IT operations, 99% automated custodian reconciliation, the virtual elimination of failed trades, increased business responsiveness, and faster time-to-market with new products and services.

Another challenge facing securities firms is managing market, credit, and operational risk. While risk is a part of their business, financial markets companies' current processes are not fully capable of responding to continuously evolving regulations and new product development. Current compliance processes are ad hoc and inadequately documented. The data required for measuring risk and ensuring compliance is spread throughout product-driven silos, resulting in an inability to monitor and manage internal processes to ensure compliance. This is true even though compliance – beyond the letter of the law – is required to stay in business.

IBM's Middleware Solution for Risk and Compliance integrates specific risk and compliance processes in the context of an enterprise infrastructure. It helps securities companies efficiently achieve compliance with changing regulations and improve the management of enterprise risk. It does so by providing data retention and archiving (the foundation for supervisory review) as well as the process and data infrastructure necessary for managing requirements. The result is data quality and consistency, and increased accountability and responsibility through business activity monitoring.

With approximately 1,100 branch offices and more than 47,000 employees, the Dresdner Bank Group is active in over 60 different countries. In terms of assets, market value, and customers, Dresdner is one of the leading banking groups in Europe. Faced with new banking industry regulations such as Basel II and increasing demands for information and continual growth, Dresdner Bank's Information Technology Unit for Credit and Risk Systems set out to tackle two challenges by implementing a new data warehouse. The first mission was to build a single source of information to provide business intelligence applications, and the second was to develop an application to support its internal auditing work.

Using IBM DB2 software, Dresdner Bank is building an easy-to-manage repository for huge quantities of data that will accommodate an ever-increasing field of users and more and more data. It will also be flexible enough to change the balance of system power exactly when Dresdner needs to do so. In addition, it will provide Dresdner's analysts with state-of-the-art tools to meet continually evolving demands for information.

IBM's middleware solution for Risk and Compliance, like all five solutions for financial market customers, can be implemented piece-by-piece based on the customer's needs. "Customers can work on an issue basis or take a very broad approach to industry-wide challenges," said Covington. "Our solutions take into account the industry's technological prowess, inherent volatility, and increasing globalization."

"While risk is a part of their business, financial markets companies' current processes are not fully capable of responding to continuously evolving regulations and new product development"

Run a servlet-Struts application as a portlet-Struts application in WebSphere Portal v5

Migration Guide

BY ASIM SADDAL

Asim Saddal works in the Portal practice in IBM Software Services for WebSphere.
asaddal@us.ibm.com

This article offers guidance to migrating existing servlet Struts–based applications to portlet Struts–based applications – with a minimum of steps – using WebSphere Studio Application Developer v5.0 with the WebSphere Portal v5.0 test environment.

It does not cover expanding the functionality of the Web application (e.g., using portlet communication), nor is it an introduction to Struts, WebSphere Portal, or portlet programming. The purpose of this article is to enable you – with the fewest changes possible – to run a servlet-Struts application as a portlet-Struts application in WebSphere Portal v5. For further information on portlet development, refer to the Portlet Development Guide.

In short, the abstract Portlet class is the central abstraction of the Portlet API. The Portlet class extends HTTPServlet of the Servlet API. All portlets extend the Portlet class indirectly, and inherit from HttpServlet. Therefore, portlets are a special type of servlet, with properties that allow them to easily plug into and run in the Portal Server. Unlike servlets, portlets cannot forward requests, send redirects or errors to browsers directly, or write arbitrary markup to the output stream. Portlets are more dynamic than servlets – they can, for example, be installed or their settings can be changed without restarting the Portal Server.

The portlet container relies on the J2EE architecture implemented by the WebSphere Application Server. As a result, portlets are packaged in WAR files similar to J2EE Web applications and are deployed like servlets. Like other servlets, a portlet is defined to the application server using the Web application deployment descriptor (web.xml). This file defines the portlet's class file, the servlet mapping, and read-only initialization parameters.

Exporting the Existing Application

From the previous environment, in WebSphere Studio:

1. If multiple projects exist for same application, merge all them into a single project.
2. Export the project to the file system as a WAR file with source files.

Note: For deployment in the WebSphere Portal, we'll actually be exporting the Web container (WAR file) only. WebSphere Portal creates an enterprise application wrapper for the WAR file at deployment.

Import the Application into the Portal Project

In the new portal development environment in WebSphere Studio:

1. Create a new portlet application project, specifying the Create empty portlet and J2EE Level 1.3 options.
2. Import the WAR file you exported earlier to the existing project and press Finish.
3. Turn off the Overwrite existing resources without warning option.
4. Answer No to Overwrite portlet.tld.
5. Answer Yes to Overwrite portlet.xml and web.xml.

Note: After importing the WAR file, the "Incorrect servlet reference for the portlet" error message will appear. Ignore it, it will be fixed later.

Setup Environment for WebSphere Portal

From WebSphere Studio:

1. Select your portlet project in the J2EE Navigator view.
2. Select File > Properties.
3. Select Java build path.
4. Make sure you have the definitions shown in Listing 1 inside the c:\<workspace>\<projectname>\.classpath file.

Note the following:

a. Make sure to check your classpath entries in the new Studio Products tooling after migration. Ensure that the classpath entries are pointing to the new WebSphere Portal v5.0 paths.

b. When you create a portlet project for WebSphere Portal v5.0, you can

choose WPS_V5_PLUGINDIR as a class variable.

5. Select File > Import File System.
6. Press Next.
7. Select the following directory as a source directory:

```
<WebSphere_Studio_root>/wstools/
eclipse/plugins/com.ibm.wps_v5_
<WebSphere_Portal_version>
```

   For example: com.ibm.wps_v5_5.0.0 – where <WebSphere_Studio_root> is the root directory of WebSphere Studio and <WebSphere_Portal_version> is the specific 4.x version of WebSphere Portal.
8. Select the following directory as a target directory: <portlet_project>/Web Content/WEB-INF/tld – where <portlet_project> is the name of your project.
9. Override the existing portlet.tld file with the new one.

THE STRUTS PORTLET FRAMEWORK

In WebSphere Portal 5.0, the supported Struts framework is Struts 1.1 RC 1. In a standard installation, you will find the Struts framework in the <wp_root>\dev\struts\ directory.

In WebSphere Studio, the Portlet Deployment Descriptor should be checked for this application to verify that the Struts filter chain is specified. The filter chain is used to implement solutions for some of the differences between the versions of WebSphere Portal.

1. The Struts application must be migrated to the newer version of the Struts Portlet Framework. Unzip the PortalStrutsBlank.war file located in the <wp_root>/dev/struts/Struts portlet directory. This is an actual Struts application, so the directory structure is similar to many Struts applications. Copy the following JAR files in the WEB-INF/lib directory from the PortalStrutsBlank.war to the WEB-INF/lib directory of the Struts application:
   - commons-beanutils.jar
   - commons-collections.jar
   - commons-dbcp.jar
   - commons-digester.jar
   - commons-fileupload.jar
   - commons-lang.jar
   - commons-logging.jar
   - commons-pool.jar
   - commons-resources.jar
   - commons-services.jar
   - commons-validator.jar
   - jakarta-oro.jar
   - jdbc2_0-stdext.jar
   - PortalStruts.jar (Struts Portlet Framework)
   - PortalStrutsCommon.jar (Struts Portlet Framework)
   - PortalStrutsTags.jar (Struts Portlet Framework – new)
   - struts.jar
   - StrutsUpdateForPortal.jar (Struts Portlet Framework)

   Note: Overwrite any file if it already exists.
2. Remove the following files from the WEB-INF/lib directory, they exist:
   - commons-services.jar
   - jdbc2_0-stdext.jar
   - commons-dpcp.jar
   - commons-pool.jar
   - commons-resources.jar
3. Move the TLD files from WEB-INF to the WEB-INF/tld directory. Note that the location of the TLD files has changed; they are now in the WEB-INF/tld directory. Update the path of these files inside the web.xml file and also inside the JSP files where the taglib tag is defined. Make sure you don't have multiple entries for the same TLD in the web.xml file.
4. Update the TLD files with the versions from the WEB-INF/tld directory of the PortalStrutsBlank.war. The TLD files that ship with the Struts Portlet Framework and should be used in the Struts application are listed here.
   - struts-bean.tld
   - struts-chtml.tld
   - struts-html.tld
   - struts-logic.tld
   - struts-nested.tld
   - struts-portal-html.tld
   - struts-portal-wml.tld
   - struts-template.tld
   - struts-tiles.tld
   - struts-wml.tld
5. Remove the ForwardAction class (if it exists) when migrating to the current release. The ForwardAction class can be found in the WEB-INF/classes/org/apache/struts/actions directory. Struts also ships an IncludeAction class. The Struts implementation of IncludeAction writes directly to the response object; this behavior is not supported for an action in WebSphere Portal, so I recommend that you use ForwardAction instead of the IncludeAction.
6. In WebSphere Studio, right-click on the project name and select Refresh so WebSphere Studio updates the project with the new library and property files.

Modifications Needed to Run in WebSphere Portal

To run the application in the

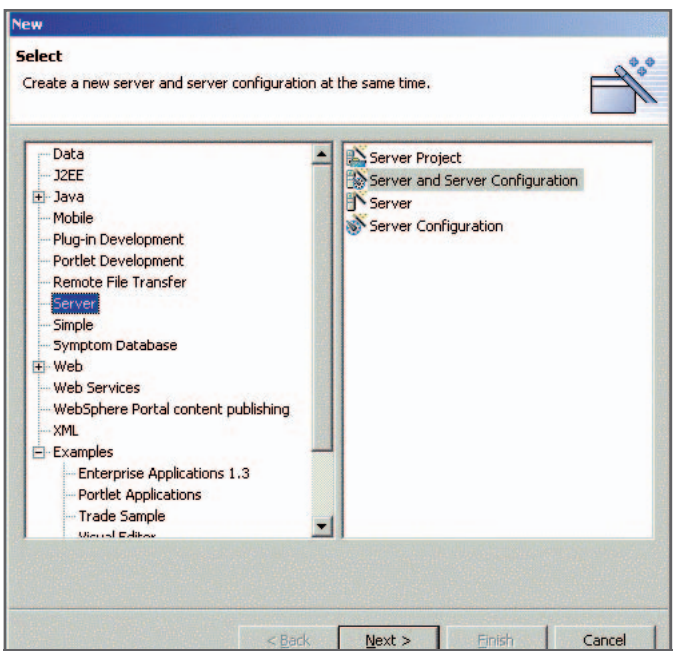


FIG 1: CREATING A SERVER AND SERVER CONFIGURATION

portal environment, the following changes are needed in some of the application configuration files.

WEB DEPLOYMENT DESCRIPTOR: WEB.XML

Most of the configuration for a Struts application is in the Web deployment descriptor file (web.xml).

1. Define a unique ID for the Web application.

```
<web-app id="WebApp_1">
        …
  </web-app>
```

2. Define the display name for the portlet.

```
<display-name>StrutsPortletUI</
display-name>
```

3. Change the <servlet-class> to com.ibm.wps.portlets.struts.WpsStrutsPortlet.

```
<servlet id="Servlet_1">
      <servlet-name>MyStrutsApp</
servlet-name>
      <servlet-class>com.ibm.wps.
portlets.struts.WpsStrutsPortlet
      </servlet-class>
      ...
 </servlet>
```

Note: The servlet ID should be suffixed with a unique ID to prevent conflicts with other portlets. Define a servlet ID for all the defined servlets inside web.xml.

4. In a portlet, you need to use a servlet URL mapping for your portlet. You would typically use a path prefix servlet mapping:

```
<servlet-mapping id="Unique_
ServletMapping">
  <servlet-name>Struts</servlet-
name>
  <url-pattern>/Struts/*</url-pat-
tern>
</servlet-mapping>
```

5. A normal Struts application requires a servlet mapping to associate paths to actions. The Struts Portlet Framework handles this using a pseudo servlet mapping that needs to be specified as an init parameter:

```
<init-param>
  <param-name>struts-servlet-
mapping</param-name>
  <param-value>*.do</param-
value>
</init-param>
```

6. The Struts application needs to specify what the initial screen is for the application.

```
<welcome-file-list>
     <welcome-file>index.jsp</
welcome-file>
 </welcome-file-list>
```

7. The Struts Portlet Framework takes advantage of this module support to provide both mode and device (markup) differentiation of the Struts configuration.

There are two paths that can be specified in the web.xml for supporting differentiation of modes and device types:

a. The first is the search path, used to determine which module (and which struts-config.xml file) to use. It also determines the base directory for locating JSPs. The following init param setting will set the search path to consider markup name and mode when searching for modules:

```
<init-param>
  <param-name>SubApplicationSearch
Path</param-name>
  <param-value>markupName, mode</
param-value>
</init-param>
```

b. The second path is the include path, used to include common JSP files in the search. The search will progress from the most qualified to the least qualified, based on the path settings. The search completes when the first file match is found:

```
<init-param>
  <param-name>IncludesSearchPath</
param-name>
  <param-value>locale</param-
value>
</init-param>
```

PORTLET DEPLOYMENT DESCRIPTOR: PORTLET.XML

1. In order to run our Struts application as a portlet we also need to create a portlet.xml file. Keep in mind that the portlet href attribute defined in portlet.xml must map to the servlet ID in the web.xml file. The definition of the servlets in the web.xml must be in the same order as the definition of portlets in the portlet.xml.

```
<portlet id="Portlet_1" href="WEB-
INF/web.xml#Servlet_1">
        <portlet-name>Mail</port-
let-name>
```

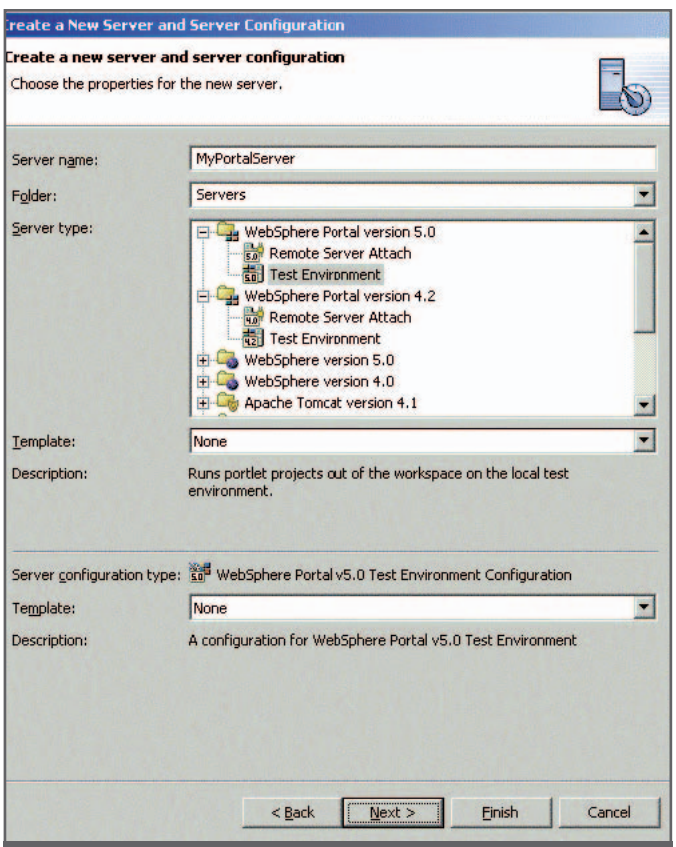


FIG 2: THE SERVER CREATION WIZARD

web services conference & EDGE expo

Web Services Edge 2005 East

International Web Services Conference & Expo

Hynes Convention Center, Boston, MA

February 15-17, 2005

The Largest i-Technology Event of the Year!

Guaranteed Minimum Attendance 3,000 Delegates

Tuesday, 2/15:	Wednesday, 2/16:	Thursday, 2/17:
Conference & Expo	Conference & Expo	Conference & Expo

Join us in delivering the latest, freshest, and most proven Web services solutions... at the *Fifth Annual Web Services Edge 2005 East – International Conference & Expo* as we bring together IT professionals, developers, policy makers, industry leaders and academics to share information and exchange ideas on technology trends and best practices in secure Web services and related topics including:

- Transitioning Successfully to SOA
- Federated Web Services
- ebXML
- Orchestration
- Discovery
- The Business Case for SOA
- Interop & Standards
- Web Services Management
- Messaging Buses and SOA
- SOBAs (Service-Oriented Business Apps)
- Enterprise Service Buses
- Delivering ROI with SOA
- Java Web Services
- XML Web Services
- Security
- Professional Open Source
- Systems Integration
- Sarbanes-Oxley
- Grid Computing
- Business Process Management
- Web Services Choreography

3-Day Conference & Education Program

features:

- Daily keynotes from companies building successful and secure Web services
- Daily keynote panels from each technology track
- Over 60 sessions and seminars to choose from
- Daily training programs that will cover Web Service Security, J2EE, and ASP.NET
- FREE full-day tutorials on .NET, J2EE, MX, and WebSphere
- Opening night reception

Interested in Exhibiting, Sponsoring or Partnering?

Becoming a Web Services Edge Exhibitor, Sponsor or Partner offers you a unique opportunity to present your organization's message to a targeted audience of Web services professionals. Make your plans now to reach the most qualified software developers, engineers, system architects, analysts, consultants, group leaders, and C-level management responsible for Web services, initiatives, deployment, development and management at the regions best-known IT business address – The Hynes Convention Center in Boston.

For exhibit and sponsorship information please contact Jim Hanchrow at 201.802.3066, or e-mail at jimh@sys-con.com.

Sponsored by:

JDJ · WebServices Journal · SD Times · wldj · .NET Developer's Journal · WebSphere Journal · Information Storage+Security Journal · XML Journal · asp.netPRO · MX developer's journal · eaindustry Consortium

Contact for Conference Information: Jim Hanchrow, 201-802-3066, jimh@sys-con.com

PRODUCED BY SYS-CON EVENTS

www.sys-con.com/edge

```
    ...
  </portlet>
```

2. Specify the Struts filter chain inside the portlet.xml by adding the following lines:

```
<config-param>
<param-name>FilterChain</param-name>
<param-value>StrutsTranscoding</param-value>
</config-param>
```

The filter chain is used to implement solutions for some of the differences between the versions of WebSphere Portal.

STRUTS CONFIGURATION FILE: STRUTS-CONFIG.XML

1. The Struts configuration used by Struts portlets references the 1.1 version of the DTD. Change the DOCTYPE if it is of a different version.

```
<!DOCTYPE struts-config PUBLIC
          "-//Apache Software Foundation//DTD Struts Configuration 1.1//EN"
          "http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">
```

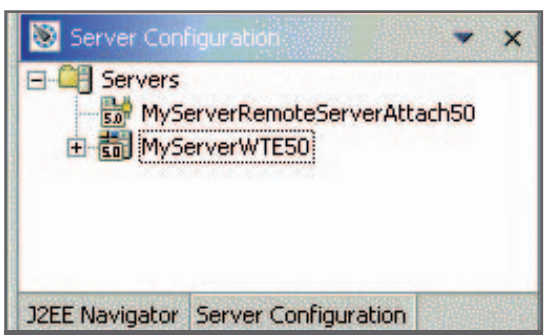


FIG 3: THE SERVER CONFIGURATION VIEW SHOWS THE NEW SERVER AND SERVER CONFIGURATION

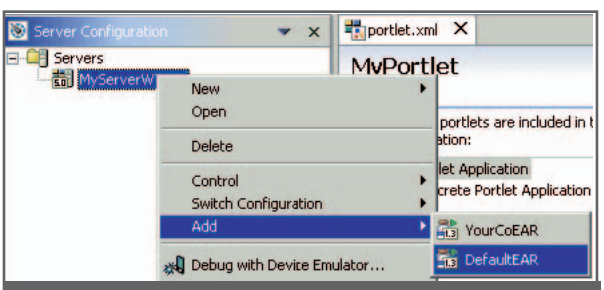


FIG 4: CLICK ON THE APPROPRIATE SERVER PROJECT AND SELECT ADD.

2. For the Struts Portlet Framework to operate properly, the RequestProcessor must be configured. The RequestProcessor is responsible for the main functionality of processing a request. The default RequestProcessor can be overridden in the struts-config.xml via the controller element. In the struts-config.xml, the following Portal Server RequestProcessor must be specified:

```
<controller processorClass="com.ibm.wps.portlets.struts.WpsRequestProcessor">
</controller>
```

CHANGES TO THE JSPS

The JSPs for Struts applications in the portal environment have to be modified to adapt to the way the portal server expects portlet URIs to be created.

1. Remove the following tags from each JSP:
   - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
   - <HTML> and </HTML>
   - <HEAD> and </HEAD>; Delete all enclosed within these tags, including <META> content, <TITLE> content, and <LINK> content.
   - <BODY> and </BODY>

2. Portlet JSPs cannot link directly to content (e.g., images or JavaScript files) within the portlet's WAR directory structure. Instead, they have to use the services of the portlet container to create portlet URIs from which the content can be accessed. Use the encodeURL() method of ServletResponse to access content within the portlet WAR structure. For example:

```
<IMG border="0" src="<%= response.encodeURL("/images/logo.jpg")%>">
```

or

```
<script language="JavaScript" type="text/javascript" src="<%= response.encodeURL("/js/myScript.js")%>"></script>
```

The encodeURL() method returns the relative URL of the content, without the hostname and domain information.

REBUILD THE PROJECT

After making all these changes, rebuild the project by selecting Project from the Web perspective and choosing the Rebuild All option from the Project menu.

Deploying a Portlet to a WebSphere Portal Test Environment

To deploy a portlet to a test environment:

1. Create a server and server configuration.
2. Add projects to the server.
3. Deploy and test the portlet.

CREATING A SERVER AND SERVER CONFIGURATION

1. Inside WebSphere Studio, select File > New > Other…. The "New" panel appears. Select "Server" on the left pane and "Server and Server Configuration" on the right (see Figure 1).
2. Click Next. The creation wizard appears. Select WebSphere Portal version 5.0 > Test Environment as the Server type.

Note: The procedure for creating a test environment in WebSphere Portal is the same as the one for creating a test environment with WebSphere Studio.

When the wizard has finished, the server and server configuration are shown in the Server Configuration view.

ADDING PROJECTS TO THE SERVER

The next step is to add the target portlet project to the server.

1. In the Server Configuration view, expand Servers.
2. Right-click on the appropriate server project and select Add.
3. Select the enterprise application project that the portlet project that you want to debug is associated with.

DEPLOY AND TEST THE PORTLET

In order to deploy and test the portlet in a WebSphere Portal test environment, right-click on the project and select "Run on Server…". This will start the WebSphere Portal test environment and will show the portlet on a portal page.

Deploying a Portlet to WebSphere Portal Server

1. Build and export the portal project to create a portlet WAR file by right-clicking the portal project and then selecting Export.
2. Install this WAR file into WebSphere Portal by using Portal Administration functions.
3. Add it to the page container.

Summary

This article discussed the minimal steps required to migrate an existing Struts servlet–based application to a Struts portlet–based application by using WebSphere Studio Application developer with a WebSphere Portal test environment.

LISTING 1

```
<classpathentry kind="var" path="SERVERJDK_50_PLUGINDIR/jre/lib/rt.jar"/>
<classpathentry kind="var" path="WAS_50_PLUGINDIR/lib/cm.jar"/>
<classpathentry kind="var" path="WAS_50_PLUGINDIR/lib/cmInt.jar"/>
<classpathentry kind="var" path="WAS_50_PLUGINDIR/lib/dynacache.jar"/>
<classpathentry kind="var" path="WAS_50_PLUGINDIR/lib/ivjejb35.jar"/>
<classpathentry kind="var" path="WAS_50_PLUGINDIR/lib/j2ee.jar"/>
<classpathentry kind="var" path="WAS_50_PLUGINDIR/lib/naming.jar"/>
<classpathentry kind="var" path="WAS_50_PLUGINDIR/lib/ras.jar""/>
<classpathentry kind="var" path="WAS_50_PLUGINDIR/lib/runtime.jar""/>
<classpathentry kind="var" path="WAS_50_PLUGINDIR/lib/servletevent.jar"/>
<classpathentry kind="var" path="WAS_50_PLUGINDIR/lib/utils.jar"/>
<classpathentry kind="var" path="WAS_50_PLUGINDIR/lib/xerces.jar"/>
<classpathentry kind="var" path="WCM_PLUGINDIR/lib/databeans.jar"/>
<classpathentry kind="var" path="WCM_PLUGINDIR/lib/wpcpauthor.jar"/>
<classpathentry kind="var" path="WCM_PLUGINDIR/lib/wpcpquery_src.jar"/>
<classpathentry kind="var" path="WCM_PLUGINDIR/lib/wpcpresources.jar"/>
<classpathentry kind="var" path="WCM_PLUGINDIR/lib/wpcpruntime.jar"/>
<classpathentry kind="var" path="WCM_PLUGINDIR/lib/wpcpruntimecommon.jar"/>
<classpathentry kind="var" path="WPS_V5_PLUGINDIR/portlet-api.jar"/>
<classpathentry kind="var" path="WPS_V5_PLUGINDIR/wpsportlets.jar"/>
<classpathentry kind="var" path="WPS_V5_PLUGINDIR/wps.jar"/>
```

Following After a Wily Thread

BY LEWIS CIRNE

Within the next five years, more and more enterprises will standardize on the J2EE platform, resulting in thousands more commercial Java application deployments. And in many cases, that means migrating existing legacy and mainframe applications to J2EE. These applications, while varying greatly in their technological complexity and their business functions, will all share a common trait: the need to be monitored and managed proactively so that organizations can prevent outages before they occur.

Case in Point

WebSphere portals can crash unexpectedly. In one company's case, the WebSphere portal crashed once every two months. The cause was unknown. Even using legacy management tools, the actual cause of the failures was not apparent. Before the IT department could begin resolving the problem, the situation would escalate to the CEO.

Customers were not happy.

The company's problem was by no means unique. When a WebSphere portal crashes, the cause is almost always some thread-related issue in the application server. Threads are the lifeblood of the running JVM, and are therefore the scarcest resource in the application environment.

"Keeping e-businesses up and running requires more than legacy systems management tools"

Keeping WebSphere Portals Up

In order to dispatch a transaction request, the JVM must have one or more available "worker threads." If a JVM runs out of available threads, the application is effectively closed for business. The result may be lost customers, lost revenue, or both.

Enterprises must be able to react to systems failures in a timely manner. Better still, they should be able to predict problems before they happen. To speed problem resolution, and to prevent outages in the first place, it is critical for an application performance management system to watch the threading activity of all WebSphere servers, all the time, in real time.

IT departments need to know how many threads are available, and, more importantly, where they are executing. For example, by placing an alert on the thread-level concurrency as it relates to a back-end system such as CICS, an IT team can be alerted to a problem and its probable cause well before it turns into an outage.

Legacy systems management tools, like the ones used by the company mentioned above, monitor applications externally and are thus unable to see threads. From an availability standpoint, if even one thread is available, the system appears to be "up"; and worse yet, the impending system crash is totally invisible to "outside in" legacy management systems.

By monitoring the application's internal gearing and interactions, IT teams can trace the transaction path of a customer's incoming request – by customer ID – for investigation and analysis after the transaction has been completed. This far exceeds the monitoring capabilities of legacy systems management tools.

Application management systems provide a broader range of tools, such as alerts, historical recording, built-in reporting, easy customizability, and enterprise scalability. Moreover, they provide visibility into hard-to-find errors such as thread-related issues.

Conclusion

Keeping e-businesses up and running requires more than legacy systems management tools can deliver alone. It requires a proactive approach to application management and development.

In a real-time, critical environment, businesses cannot afford anything less.

Lewis Cirne, founder and CTO of Wily Technology, invented the company's core, patented Java Agent technology. As Wily's CTO, Lewis takes a leading role in the future technological development of the company and its products, with the goal of extending the services offered by Wily to customers deploying high-performance e-business solutions.
lcirne@wilytech.com