SYS-CON MEDIA

WEB SERVICES • LINUX • CLUSTERING • INTEGRATION

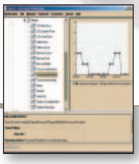
BEA WebLogic DEVELOPER'S JOURNAL

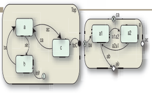
FEBRUARY 2004 - Volume:3 Issue:2

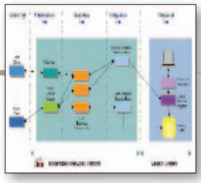
weblogicdevelopersjournal.com

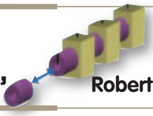
RETAILERS PLEASE DISPLAY
UNTIL MARCH 31, 2004
$8.99US $9.99CAN

DEPLOYING WEBLOGIC ON LINUX









EDITORIAL

Twin Beaks

BY JOE MITCHKO

While recently reading through a leading national newspaper, I happened to notice a slick, full-page advertisement for a new enterprise-class server. Made by a leading hardware manufacturer, it was one of those N-way Intel Xeon rack-mount systems that are in a whole league by themselves compared to the machines you and I have at home. This baby was more like a mainframe with its ability to handle several operating system platforms at once, and can literally take the place of several machines in your server room.

Being somewhat of a hardware aficionado, I just had to study the ad and see what's new in the world of microprocessor and operating system magic. Then something caught my eye – or to be exact something was missing. The ad made no mention of the hardware manufacturer's native Unix-based operating system! I've heard that the major hardware vendors are embracing Linux in a big way, but when Linux and Windows are mentioned to the exclusion of their bread and butter OS, times are a-changing.

What I find most amazing about the Linux story, and its rapid adoption by the corporate world, is its humble beginnings and the power of a seemingly grassroots effort by hundreds, if not thousands, of programmers around the world. The operating system is, in a sense, made for the people by the people; or for those Unix purists, for the programmer, by the programmer.

Further, I firmly believe that the forces that govern the software we see today are best described as evolutionary in nature. Similar to the way it works in nature, the evolutionary process is hard at work in the computing field making certain software products succeed over time while relegating others to disappearing into the proverbial trash bin. When you really look at it, it is simply survival of the fittest.

One of the most famous research projects dealing with evolution was Darwin's study of finches on the Galapagos Islands. There, among other things, he discovered that certain species with long beaks were good at gathering certain foods (like nuts), while species with short beaks were better at gathering other types of food. If a species happens to have a long beak, and the only food available requires a short beak, the species with the long beak will eventually die out.

If you carry this analogy to the world of software, you can draw the same parallels. If I am a company charging tens of thousands of dollars per CPU for my operating system, and another operating system comes along with the same capabilities and can be purchased for nearly nothing, I am not going to make it either.

Moreover, the software evolutionary process may not necessarily be driven purely by economics, but also by new innovations, such as Java technology. Software engineers from Sun were looking for a "new new" thing in the interactive video space but ended up applying the technology to the Internet. And we still feel the effects of their work today.

It is difficult to say how Linux-based J2EE platforms will fare in the marketplace as time goes on, and whether corporate America will continue to embrace Linux when things become rosier from an economic standpoint. I believe that Linux is a great OS, and whatever the price, such decisions will hopefully be made with its capabilities and merits in mind.

As you may have already guessed, this month we focus on Linux as a hardware platform for WebLogic. I hope you enjoy reading this issue!

AUTHOR BIO...

Joe Mitchko is the editor-in-chief of *WebLogic Developer's Journal* and a senior technical specialist for a leading consulting services company.

CONTACT: joe@sys-con.com

BEA WebLogic DEVELOPER'S JOURNAL

EDITORIAL ADVISORY BOARD
TOM ANGELUCCI, LEWIS CIRNE, KEVIN JONES, TYLER JEWELL, WAYNE LESLEY LUND, SEAN RHODY, CARL SJOGREEN

FOUNDING EDITOR
PETER ZADROZNY

EDITOR-IN-CHIEF
JOE MITCHKO

PRODUCT REVIEW EDITOR
JASON SNYDER

EXECUTIVE EDITOR
GAIL SCHULTZ

EDITOR
NANCY VALENTINE

ASSOCIATE EDITORS
JAMIE MATUSOW, JEAN CASSIDY

ASSISTANT EDITOR
JENNIFER VAN WINCKEL

WRITERS IN THIS ISSUE
ERIC BALLOU, ERIC GUDGION, MIKE HAVEY, PETER HOLDITCH, ANWAR LUDIN, JOE MITCHKO, KUNAL MITTAL, LAURENCE MORONEY, ROBERT PATRICK, ARAVIND, PAVULURI, STEVE POZARYCKI, VADIM ROSENBERG, NITIN SINGHVI, JASON SNYDER

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

PRESIDENT AND CEO
FUAT KIRCAALI

VP, BUSINESS DEVELOPMENT
GRISHA DAVIDA

GROUP PUBLISHER
JEREMY GEELAN

TECHNICAL DIRECTOR
ALAN WILLIAMSON

PRODUCTION CONSULTANT
JIM MORGAN

ART DIRECTOR
ALEX BOTERO

ASSOCIATE ART DIRECTORS
LOUIS F. CUFFARI • RICHARD SILVERBERG

ASSISTANT ART DIRECTOR
TAMI BEATTY

SENIOR VP, SALES & MARKETING
CARMEN GONZALEZ

VP, SALES & MARKETING
MILES SILVERMAN

ADVERTISING SALES DIRECTOR
ROBYN FORMA

DIRECTOR OF SALES AND MARKETING
MEGAN MUSSA

ADVERTISING SALES MANAGERS
ALISA CATALANO • CARRIE GEBERT

ASSOCIATE SALES MANAGERS
KRISTIN KUHNLE • BETH JONES

ADVERTISING INTERN
ALYMA HAWTHORNE

PRESIDENT, SYS-CON EVENTS
GRISHA DAVIDA

CONFERENCE MANAGER
MICHAEL LYNCH

NATIONAL SALES MANAGER
SEAN RAMAN

FINANCIAL ANALYST
JOAN LAROSE

ACCOUNTS RECEIVABLE
CHARLOTTE LOPEZ

ACCOUNTS PAYABLE
BETTY WHITE

VP, INFORMATION SYSTEMS
ROBERT DIAMOND

WEB DESIGNERS
STEPHEN KILMURRAY • CHRISTOPHER CROCE

ONLINE EDITOR
LIN GOETZ

CIRCULATION SERVICE COORDINATORS
SHELIA DICKERSON • EDNA EARLE RUSSELL

JDJ STORE MANAGER
BRUNILDA STARAPOLI

EDITORIAL OFFICES
SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645
Telephone: 201 802-3000 Fax: 201 782-9638
SUBSCRIBE@SYS-CON.COM
BEA WebLogic Developer's Journal (ISSN# 1535-9581)
is published monthly (12 times a year)
Postmaster: Send Address Changes to
BEA WEBLOGIC DEVELOPER'S JOURNAL,
SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645

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

SYS-CON MEDIA

BY LAURENCE MORONEY

What Brings the Future?

A NEW DIRECTION

The novel *Dune,* by Frank Herbert, a classic in science fiction, is littered with great references and quotes. One such quote is: "Knowing there's a trap is the first step in evading it." With this in mind I can't help but think of the future for BEA WebLogic. The trap I refer to is the consolidation of application servers and operating systems to implement application server platforms.

Consider this, at the recent PDC, Microsoft unveiled project Indigo: something that the development community has been crying out for – a .NET Enterprise Edition. Its similarity to J2EE concepts is even more striking than the architectural similarity between .NET and Java! This is no coincidence, and imitation is the greatest form of flattery. However, you have to realize that the framework upon which enterprise applications may be built in the Microsoft space will be part of the operating system itself. Look at IBM and consider their huge Linux play and it isn't difficult to see that they too are entering the same space – where the operating system is the application server. Why buy WebSphere separately from Linux when you can buy a server with IBM hardware, the IBM Linux operating system, and the WebSphere application server as a single solution?

Finally, look at the fathers of J2EE, Sun Microsystems, and their recent announcements around the Java enterprise system in which, you guessed it, a J2EE application server is part of the "new" Solaris.

From this you can only conclude that a consolidation is underway. Are the days of the independent application server numbered? I can't help but think of the bad old days when spell checkers were independent of word processors. They were vital for a word processor to be a "real" word processor and hence evolved over time to being a part of the word processor. Nowadays it is unthinkable to have them separated. In a similar way, today we see an enterprise-class application server as a vital part of the server-side operating system. It isn't much of a leap from that to envisage an application platform, comprised of an operating system + application server.

So where does this leave BEA and WebLogic?

If the future is as I have outlined above, and if it comes to pass in a similar vein to the word processor example, then they could be headed into a trap. However, remembering Herbert's quote, knowing there is a trap is the first step in evading it. There are a number of options that spring to mind in how they could evade it and maybe even increase their market share.

The first is that BEA could innovate strongly in what their application server can offer that others do not, hence increasing its value and making it a compelling purchase on top of the one that comes with the Enterprise OS, be it Microsoft Windows, IBM Linux, or Sun Solaris. Indeed, they have already taken steps into this space with the excellent workflow engine that is provided with BEA WebLogic 8.1. In addition, they have an excellent suite of complementary products such as BEA WebLogic Portal or BEA Liquid Data for WebLogic.

Alternatively they could join in the trend and produce a BEA distribution of Linux that is preconfigured with the WebLogic application server in much the same way that IBM is doing. With BEA's reputation and expertise in building quality products that are easy to use, this could be a valuable innovation for their customers. It would be logical that this platform would be the only one to offer WebLogic JRockit, which is probably the best Java runtime engine there is!

These are but two of many solutions to the potential trap. The bright people at BEA have doubtless discussed many more. My personal opinion is that while there is a threat, and a significant one, the level of innovation and dedication to quality shown by BEA means that they have the right stuff to face up to this threat, and find a way to thrive in spite of it!

AUTHOR BIO...

Laurence Moroney is a senior architect with the Innovations Laboratory at a major financial services company in New York City. He has been working in the software industry for over 10 years in such diverse arenas as casinos, airports, jails, the INS, and financial services.

CONTACT: lmoroney@philotic.com

Cyanea

Deploying WebLogic on Linux

You Are Not Alone

BY **ERIC GUDGION**

AUTHOR BIO

Eric Gudgion is a professional services engineer with Wily Technology, Inc. He has more than 19 years of computing experience covering most commercial computer environments. Most of that time has been spent working with operating systems, networking, application servers and performance tuning products.

CONTACT...

egudgion@wilytech.com

The rising business trend toward using open source software platforms has brought an increase in the number of critical applications deployed on Linux and BEA WebLogic. For many organizations, in fact, WebLogic deployments are their first major Linux installation.

This article provides an overview of deployment considerations when using a Linux/WebLogic combination.

Linux deployments span traditional Intel-based servers from grid environments to mainframe systems (IBM's z/VM with Linux guests for example). In this article we will only cover the Intel architecture; however, almost all the points covered in this article are applicable for non-Intel deployments.

Why Linux?

Why the increasing number of deployments? Linux provides an alternative to proprietary operating systems. It can offer lower cost of ownership for some customers and has a large following of skilled workers. The Linux operating system is highly configurable and the source is usually available, so you can change the behavior or recompile options that are specific for your site. Lastly, a number of vendors support Linux, allowing the customer to pick the application software and hardware that is right for them.

Picking Your Distribution

WebLogic currently supports the major Linux distributions (Red Hat and SuSE). Refer to the BEA site (http://edocs.bea.com/wls/certifications/certs_810/overview.html#1043408) for the updated list of supported configurations. Both Red Hat and SuSE contain additional features (like cluster services) that may be useful for your installation. At the time of this writing, Red Hat had just released Enterprise Linux v3, so check on the certification pages for this version of Linux as several important enhancements have been added to the kernel, like Native POSIX Threading Library (NPTL).

Picking Your JVM

BEA's JRockit JVM can be used on an Intel Linux deployment and can provide many benefits as it supports both 32- and 64-bit environments. JRockit is designed for server-side execution and has advanced features like adaptive optimizations that can improve performance of the application. If you are running on a different platform (zLinux, for example) refer to the BEA supported platform page for the supported JVM.

Installing the JVM (JRockit)

JRockit's installation is simple: download the installer for your platform, execute the downloaded file (./jrockit-8.1sp1-j2se1.4.1-linux32.bin), and follow the on-screen prompts.

If you're running on an Intel processor with Hyper-Threading enabled, you will have an extra step once the installation is completed. The cupid for each processor (real and virtual) must be readable by any process; this can be achieved automatically or by changing the /dev/cpu/X/cupid (X is the CPU number) file permissions. Refer to the JRockit Release Notes (http://edocs.bea.com/wljrockit/docs81/relnotes/relnotes.html) for all the details on enabling this support.

Installing BEA WebLogic

Just as with JRockit, the installation of BEA WebLogic is very simple. Download the distribution for your environment and execute the download (./platform811_linux32.bin). The installer provides a GUI (the default) or console (non-GUI) installation option. If you are installing on a platform without a GUI or on a remote system you can bypass the "–mode=console" option when you start the installer. Either option will walk you through the installation process, which is interactive and allows you to select installation options and the home directory.

Maintenance

A number of factors must be considered when deploying BEA WebLogic on Linux. For example, configuration of the J2EE application server and the surrounding ecosystem must be properly planned so that the best performance can be achieved. Before the environment is deployed, for best performance start the process of maintaining the environment. This preplanning will pay off once the application is in production.

Collecting performance metrics on the application and its supporting infrastructure is very important (even before production). Recording these metrics prior to production enables capacity estimates to be built and also allows a reference baseline to be created so that changes to the application or environment can be validated against the baseline prior to a production deployment.

Once in production, collecting and persisting these metrics allows a performance model to be established.

Most vendors have a service to keep you informed via e-mail about patches and updates. Be sure to sign up for these services and make sure the e-mails go to a number of people within the IT group responsible. After all, if the notifications only go to one user, you can imagine what would happen if that user happened to be on vacation and an emergency patch was posted.

Although some automatic update services are available, I would hesitate to use them and would opt for the notification of updates first. Then you can decide what is applicable for your installation and if any cross-vendor dependencies exist.

Although products from different vendors typically play well together, the combination of your applications and the vendor's solution may require testing within your environment before a production deployment. Use the measurements taken to compare the performance delta before and after deploying into production.

One tool to consider for your Linux deployments is Tripwire (www.tripwire.com). Both the open source and commercial variants can be very helpful in identifying the "what changed during the weekend" syndrome. Using Tripwire to create a baseline of the servers can be helpful when used in addition to your change management process to validate software and file consistency or rolling back changes.

Environment Visibility

A BEA WebLogic application often has a number of external touch points that are non-Java. Examples of these are Web servers and databases. The overall performance of the WebLogic application is influenced by how well these other components execute and the overall performance of Linux.

Examples of gathering EPA (Environment Performance Agent; see sidebar, page 10) data include the following;

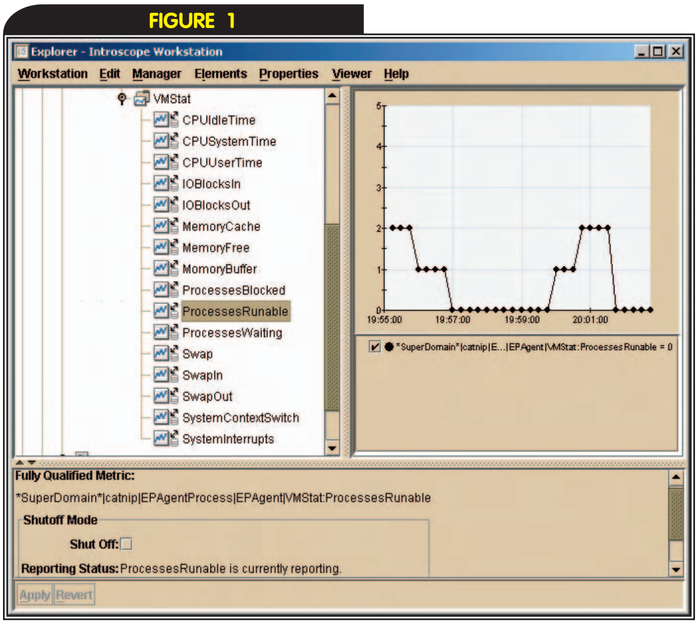
- ***Linux VM data***
  - Is too little memory available, causing Linux to swap?
  - How many tasks are pending and what is the load average?
- ***Web server data***
  - How many HTTP errors occurred between measurements?
  - Are the child threads hung?
- ***Database***
  - How much space is remaining?
  - What is the cache hit ratio?
- ***Network***
  - What IP is generating the most requests?
  - Any event alerts on the network?

What Should You Monitor?

This is a loaded question and the answer really depends on the application and your own goals for monitoring and measuring success.

As a general rule of thumb, in addition to the J2EE components

FIGURE 1



☑ ● *SuperDomain*|catnip|E...|EPAgent|VMStat:ProcessesRunable = 0

Fully Qualified Metric:
SuperDomain|catnip|EPAgentProcess|EPAgent|VMStat:ProcessesRunable

Shutoff Mode
Shut Off: □
Reporting Status: ProcessesRunable is currently reporting.

The ability to monotor non-Java metrics such as VM data is a key part of maintaining overall WebLogic application performance on Linux

within the application, anything that feeds the application, or which the application server relies on to process a request, should be monitored. Review the Environment Visibility section above and consider the touch points your own application has. How do you measure availability and acceptable performance and what are you going to actually do with the data you collect (which is very valuable)?

Collecting metrics like CPU, component response time, memory usage, thread, JDBC pool usage, and concurrent requests are a starting point in creating an understanding of the application performance. Certainly many other components are available and can be incorporated into the measurement.

One consideration you need to make before deploying the application is what happens when it does not perform within the guidelines you set for it (assuming you created a baseline before production).

Linux Configuration

The first step is to understand the physical machine. Using a few displays can help:

- Display the CPU information (cat/proc/cpuinfo). The CPU type will be displayed for each CPU in the machine.
- Display the memory information (cat/proc/meminfo). The memory size, swap, and cache details are displayed.
- Display the disk capacity and free space (du –lh).
- Display the Network configuration (ifconfig).

The information collected above will help you determine where the application files should reside, the network bindings, and the amount of memory you can use for the application (Java heap).

Review the services that are running on the machine. For example, should the machine running BEA WebLogic have an FTP or mail server running? Remove (or comment out) services that are not required and edit the /etc/xinetd.conf or /etc/inetd.conf (depending on your Linux distribution). Once the services you don't need have been removed, create a baseline of disk and memory usage. Use load generation tools to observe how Linux performs, how many IO operations occur per second and how much swap space is used (iostat and vmstat).

The baseline data can then be used for monitoring.

Runtime Secrets

Now that WebLogic is deployed on Linux, let's look at some of the process information from a Linux perspective.

Find the Linux process ID for WebLogic (ps –ef|grep java). Notice that Linux has a process for each thread so the display is a little different from other operating systems. For our example, we will assume the process ID (pid) is 27260.

If we needed to know what terminal started the server and if the terminal is remote, how would we do that? Access the /proc/fd directory, which contains the list of file descriptors used by this process. Now list fd 0 (standard input) using the list command (ls –l) and the actual device will be displayed. In this case it was /dev/pts/6. We can use the Linux who command to see who logged on that device and its IP address.

```
> cd /proc/27260/fd
> ls -l 0 lrwx------    1 root     root          64 Nov 20 14:21 0 ->
/dev/pts/6

>who
weblogic     pts/6    Nov 20 10:55 (192.168.1.105)
```

We can also display the startup command and the environment variables that are being used by this process. This can be useful when trying to track down whether a certain option has been passed to the process via a script. Using the Linux cat command, display the cmdline and environ files (cat cmdline environ).

Another useful trick can be in determining what files are being used by the process. Display the maps file (cat maps), which displays the files that have been opened. An example use case could be to determine if a certain JAR file was loaded and the directory that it was loaded from.

```
> grep trader.jar maps

/opt/bea/weblogic81/samples/domains/examples/examplesServer/stage/_apps-
dir_webservices_trader_ear/trader.jar
```

Tuning Considerations

Once the application has been in production for a short period of time (3–6 months), the operating system, application, and any of the touch points should be tuned – or at least the configuration parameters should be reviewed to ensure they are still appropriate. This is one of the benefits of persisting measurements made with the environment and its touch points. To measure but not persist the key metrics would be wasteful.

– continued on page 58

Troubleshooting Your Linux Deployment

Using performance-monitoring tools such as Wily's Introscope, the performance of the application and the other environment components that make up the whole application can be captured and recorded to a persistent store.

Using Introscope and Introscope features such as the Environment Performance Agent (EPA), which is designed specifically for the collection of metrics from non-Java touch points, can offer you a "whole application" view of the operating environment. For example, you can use Introscope EPA to collect vital operating system–level data and Web server data, combine that with J2EE application data collected using the Introscope and then display all of these metrics on a dashboard for viewing. The metrics are then converted into performance metrics that can be used by Introscope to provide a view of the overall performance of the application.

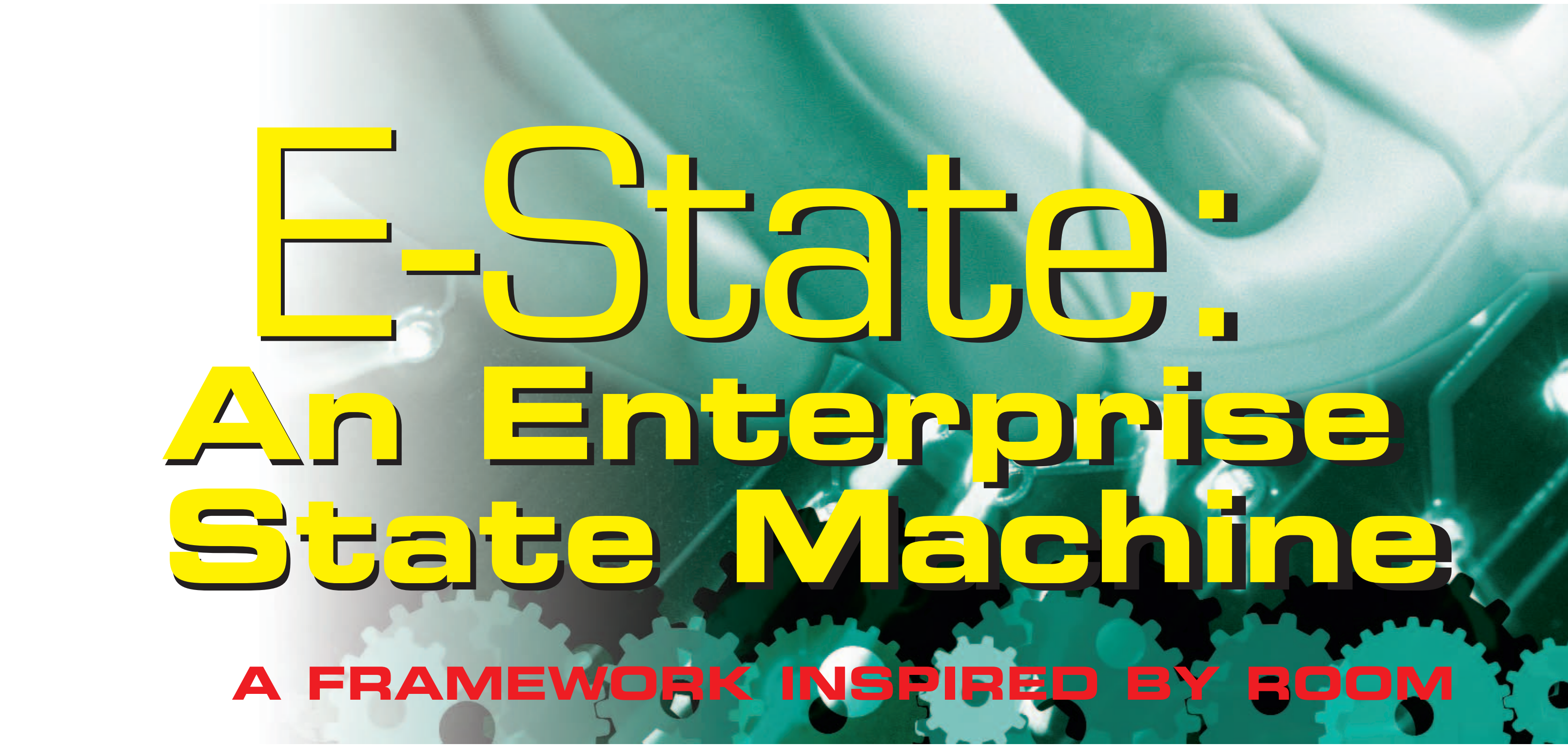
Tools like Introscope Transaction Tracer enable you to capture a request outside of the baseline for analysis or to create alerts to notify support staff of potential areas to investigate. These are some of the ways to address runtime issues.

Introscope LeakHunter can also be used to track potential memory leaks within the application. If leaks are found, the class name, method, and size will be available so that a programmer can correct the problem.

You can use Introscope to create dashboards for the various support teams within your organization before deployment so that if issues arise in production, your team members have data from the application server and supporting systems ready, enabling them to better assist in problem resolution.

Using Introscope EPA, real-time performance data from Linux can be collected and used for monitoring and alerting. When combined with the in-depth metrics Introscope collects from BEA WebLogic, a complete picture of the application and all of its supporting systems is available (see Figure 1).

oops.
N26MA
Forget something?



E-State: An Enterprise State Machine

A FRAMEWORK INSPIRED BY ROOM

BY **MICHAEL HAVEY**

AUTHOR BIO...

Michael Havey is a BEA consultant with eight years of industry experience, mostly with application integration. He is also a father, poet, philosopher, and sports fan.

CONTACT...

mhavey@bea.com

REPRODUCED WITH PERMISSION FROM BEA SYSTEMS.

E-State and Workflow

Workflow and state machines are, as argued in my earlier article, "State Machines and Workflow" (***WLDJ***, Vol. 3, issue 1), complementary implementation strategies for process-oriented applications. The state approach is a powerful abstraction for the succession of milestones found in many business processes. On the other hand, workflow – in particular, the Business Process Modeler (BPM) component of BEA WebLogic Integration – provides important enterprise-level services, such as external system integration, human worklist and task management, events and timers, and XML messaging processing. As the earlier article pointed out, a hybrid state-workflow solution has the following parts:

1. A state machine framework, consisting of:
   – ***A state model:*** A set of states and transitions, expressed in an XML document.
   – ***Actor database:*** An entity that has state. The state of the actor is persisted to a database by the state machine.
   – ***State machine engine:*** Injects events into an actor's state model and updates state accordingly. It also calls user-defined action classes when a state is entered or exited or a transition occurs.
   – ***Action classes:*** User-defined Java classes that respond to the entry or exit of a state or the execution of a transition for an actor in a given state model.
2. A BPM workflow that receives an event and injects it into the state machine.
3. A BPM workflow that sets a timer and injects a timeout event into the state machine when the timer expired.
4. A BPM workflow that is called by a state action to assign a worklist task or interact with an external system.

E-State is a reference implementation of the first part: the state machine framework. This article discusses its architecture.

E-State Architecture

Methodology

E-State's state models are based on those in the Real-time Object-Oriented Modeling (ROOM) methodology. ROOM state charts are hierarchical, meaning that a given state can have substates. This simple idea is disproportionately powerful. Flat state models simply don't scale from a human understanding point of view. As the number of states and transitions increases, the flat model becomes too hard to comprehend. Hierarchical state models can be understood in pieces, each piece being relatively simple to conceive. For example, consider Figure 1.

In this model the transition *ab* leads either of states *a1* or *a2* to state *b*; similarly, the transition *ac* causes the state to change from *a1* or *a2* to *c*. The transition *toC* for all states causes the state to change to *c*. Transition *ba* in state *a1* results in the new state *b*. The initial state is *b*. The state changes from *a1* to *a2* on transition *a1a2*, and from *a2* to *a1* on *a2a*1. Transition *ca* leads to a choice point: if the last *a* state is *a1*, it leads to *a1*, otherwise *a2*.

The hierarchical version of this scenario, depicted in Figure 2, is easier to understand.

First, states *a1* and *a2* are factored out as superstate *a* (right diagram); the overall picture (left diagram) is cleaner. Transitions *ab* and *ba*, from the perspective of the top state, simply shift between states *a* and *b*; but in state *a* we see that *ba* leads to substate *a1*, whereas *ab* originates from *a1*. The transition *ac* leads from either *a1* or *a2* to *c*; the transition *ca* leads from *c* to the last substate of *a*. In addition, the *toC* transition no longer needs to originate from each state; to event from the top state's nonextending transition point *toC* to state *c* satisfies the intended behavior.

As a consequence of its hierarchical design, ROOM offers two powerful features: group transitions and transitions to history. A *group transition* is one that occurs for a given state no matter what substate it is in; transition *ac* leads from state *a* to *c* regardless of whether *a*'s substate is *a1* or *a2*. A *transition to history* is a transition into the most recent substate of a given state; transition *ca* brings state *c* to the last substate of *a*.

Alternative methodologies are UML and Petri-nets, each of which supports hierarchical structures.

Engine

The heart of E-State is a stateless session Enterprise JavaBean (EJB) called StateMachine, which is shaded in the diagram in Figure 3.

The StateMachine EJB is configured to point to a particular state model, which is an XML file specifying, for a given application, the following:

- A set of states and a set of transitions
- A unique namespace, which enables multiple deployments, as discussed below
- The name of the Java "action" callback class that the state machine invokes while processing an event

The EJB applies the model to "actors." An actor, in ROOM methodology, is an "active" object whose behavior is best described with a state model. (In ROOM, an active object also has its own control thread of and its own set of inbound and outbound messaging interfaces. E-State's notion of an actor is more restrictive.) In E-State, an actor is an entity that has state, such as an insurance claim. The transition from one state to another in a model reflects the change in state of the actor; for example, a claim can be waiting, activated, or idle. E-State has three tables that track the state of an actor, namely Actor, Actor_Property, and Actor_State, and has corresponding entity EJBs (Actor, ActorProperty, and ActorState) to represent them, as shown in Figure 3. The StateMachine EJB is in part a facade to these entity EJBs; its actor-management methods include createActor(), getCurrentState(), getChildState(), getActorProperty(), getActorProperties() and setActorProperty().

The remaining methods of StateMachine (startMachine() and injectEvent()) constitute the engine that drives an actor's change of state. Actually, startMachine(), simply calls injectEvent(), passing the special "initial" event type to it, to execute the transitions originating from the initial transition points of each state in the model. The injectEvent() method, then, is the heart of the machine that moves the action of the business process along. This method invokes methods of the user-defined action class along the way, implementing the interface StateAction that is defined with the model. The purpose of the action class is to notify the client of important state machine events and to request logical decisions. The methods of the action class are shown in Table 1.

In the insurance example, action classes start workflows to perform task-related or cleanup work, or to start a timer. In WebLogic Integration 7.0, a workflow is started with the BPM API. In WebLogic Integration 8.1, the workflow is called as a Web service.

The methods of the StateMachine EJB are summarized in Table 2.

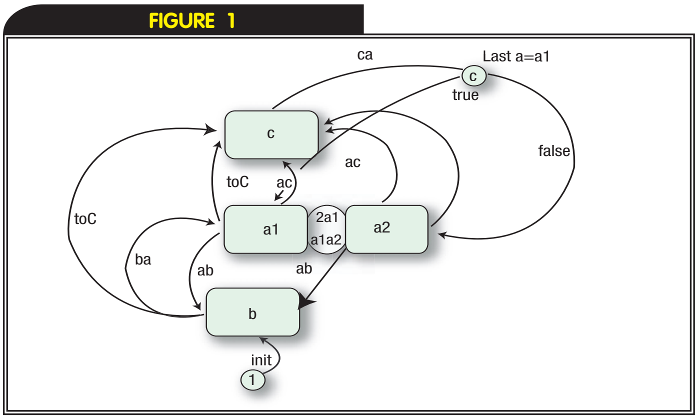
In the insurance example, the injector workflow calls the injectEvent() of the state machine. In WebLogic Integration 7.0, the workflow uses a business operation to call this method. In WebLogic Integration 8.1, it uses an EJB control.

Database Schema

Figure 4 shows the structure of tables to hold persistent actor state information.

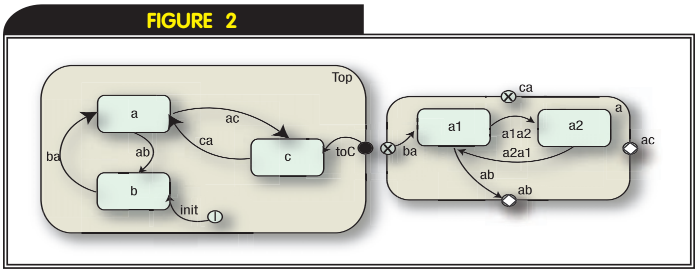
The main table, Actor, stores the current state of an actor for a particular model type. The current state is the leaf state that the actor is currently in. The primary key is a combination of the

FIGURE 1



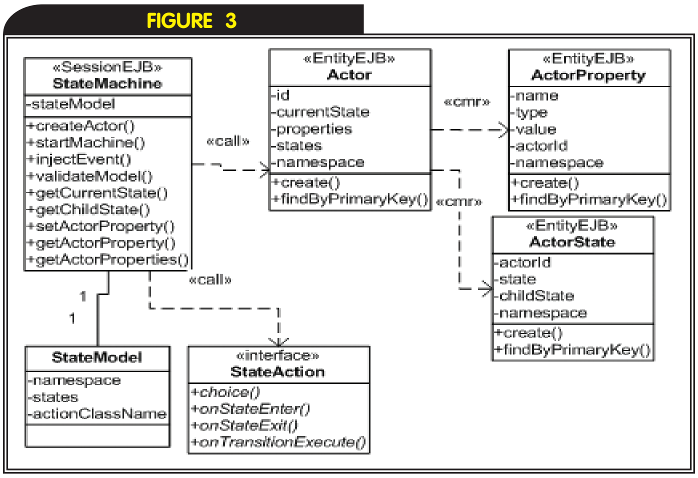
Flat model

FIGURE 2



Hierarchical model

FIGURE 3



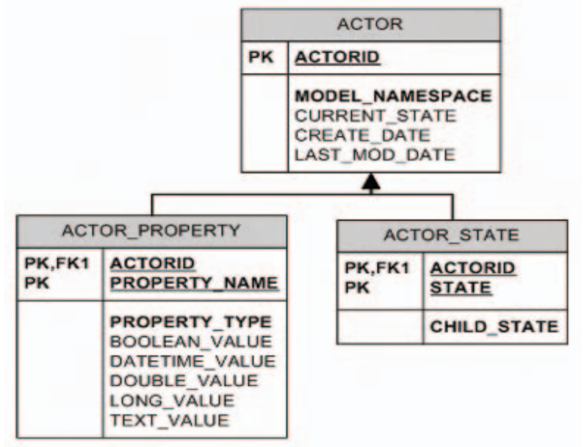
E-State object model

actor's unique identifier and its model namespace. An actor can have state in several model namespaces. Notably, if two versions of a model exist with distinct namespaces, the actor's state in each can be represented in the actor table.

The Actor_State table captures the active substate of each composite state of a given actor in a given namespace. This table is for internal use only; the engine uses it to process transitions to history. There is a one-to-many relationship between actor and actor_state.

The Actor_Property table stores user-defined properties for an actor in a given namespace. A property has a name (which is unique per actor per namespace), a type, and a value. This table is provided as a convenience to client applications to associate a set of data with an actor; more likely, that data is held in an application-specific data store.

FIGURE 4



E-State data model

TABLE 1

METHOD	ACTION
OnStateEnter	Notification that a state is being entered
OnStateExit	Notification that a state is being exited
OnTransitionExecute	Notification that a transition is being executed. The method returns true to allow the transition, false to prevent it (a guard, in ROOM parlance)
Choice	Request to execute a choice point decision. Return true or false to control which way the flow of control branches in the state model

Action class methods

TABLE 2

METHOD	ACTION
CreateActor	Creates a new record in the Actor table for the model represented by the StateMachine EJB
GetCurrentState	Returns the current leaf state for the actor for the model represented by the StateMachine EJB
GetChildState	Returns the current sub-state of the specified state for the actor for the model represented by the StateMachine EJB
GetActorProperty	Gets the value of the specified property for the actor for the model represented by the StateMachine EJB.
SetActorProperty	Sets the value of the specified property for the actor for the model represented by the StateMachine EJB
GetActorProperties	Gets the name, type, and value of each property for the actor for the model represented by the StateMachine EJB
StartMachine	Initializes the state model for the actor for the model represented by the StateMachine EJB by executing the initial transitions of each state in the model.
InjectEvent	Injects the specified event into the state model of the actor for the model represented by the StateMachine EJB

State machine EJB methods

Deployment

A different instance of the StateMachine EJB is deployed for each state model. The source code is the same for each (same home and remote interfaces, same implementation), but the configuration is different. The deployment descriptor for the EJB specifies a unique Java Naming and Directory Interface (JNDI) name (which clients use to locate EJB), as well as a reference to an XML file containing the model itself. For example, the StateMachine EJB for the insurance state model might have the JNDI name "state_insurance" and point to file "Insurance.xml". To interact with this model, a client application accesses its EJB with the "state-insurance" JNDI name and calls its methods. This peculiar approach has significant advantages:

- ***Life cycle:*** To make a new state model available for processing means to deploy a StateMachine EJB that points to it. To decommission that model means to undeploy that EJB. To promote a change to the model means to redeploy the EJB with the modified model file.
- ***Versioning:*** If an existing state model has a new major version, the new version can be deployed as a separate EJB and coexist with the earlier version. For example, "state-insurance-1.1" can coexist with "state-insurance".

The data model is also conducive to versioning. A given actor can have persistent state for multiple models, including different versions of the same model, as long as the models have distinct names.

Most business processes run for a long time, which makes the management of application upgrades especially challenging. Two scenarios are hard to solve:

1. A minor patch where there are actors in progress on the unpatched release.
2. A major patch meant only for new actors; old actors are to continue with the earlier version.

E-State is an intelligent solution to these problems:

1. Applying the minor patch means redeploying the existing EJB for that model. The actors will pick up where they left off, but with the patch in place.
2. Applying the major patch means deploying a new EJB with a distinct namespace, but keeping the existing EJB deployed as is. The models are independent, allowing old actors to continue with the earlier release while new actors use the new release.

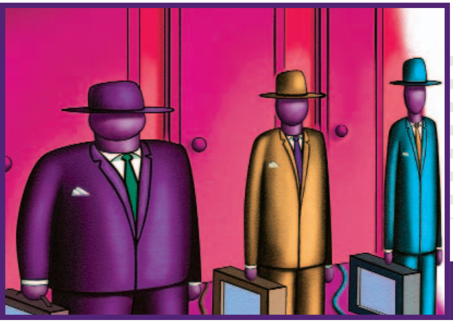
Conclusion

E-State is an enterprise state machine framework inspired by the ROOM methodology that, when combined with BPM workflows offering key integration services such as systems integration, events, timers, worklist, and XML, provides powerful solutions for development of process-oriented business applications. E-State includes a runtime engine, a schema for state models, persistence services, and user-defined "action" classes, which are invoked by the engine when a transition occurs, or when a state is entered or exited. The action classes likely call workflows to leverage BPM services; workflows, in turn, when triggered by events, call the engine to trigger transitions.

References

- Selic, Gullickson, and Ward. (1994). *Real-Time Object-Oriented Modeling*. Wiley.

Get Optimized.™



Application Management with WebLogic Server for Developers, PART 4

THE BASICS OF WRITING CUSTOM JAVA APPLICATIONS USING JMX

BY **ROBERT PATRICK & VADIM ROSENBERG**

AUTHOR BIOS...

Robert Patrick is a director of technology in BEA's CTO Office and coauthor of the book *Mastering BEA WebLogic Server: Best Practices for Building and Deploying J2EE Applications*. Robert has spent his career helping customers design, build, and deploy high performance, fault-tolerant, mission-critical distributed systems using BEA Tuxedo and BEA WebLogic Server.

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

CONTACT...

rpatrick@bea.com
vadimr@bea.com,

REPRODUCED WITH PERMISSION FROM BEA SYSTEMS.

This article is the fourth in a series of articles on BEA WebLogic Server administration and management for developers (**WLDJ**, Vol. 2, issues 10–12). The first installment focused on administration concepts and terminology, and the graphical tools for packaging an application and setting up and configuring a WebLogic Server domain.

In the second article, we focused on application deployment, runtime management, and the monitoring facilities available with WebLogic Server that did not require knowledge of JMX. The last article discussed the basic concepts and terminology of JMX and the WebLogic Server 8.1 JMX infrastructure, as well as showing you how to use JMX-specific tools that come with WebLogic Server 8.1.

In this article, we'll show you the basics of how to write custom Java applications that use JMX to configure, administer, and manage BEA WebLogic Server 8.1–based applications.

JMX Programming Fundamentals

When writing a Java application that uses BEA WebLogic Server's JMX capabilities, the first thing you need to do is decide whether to use the standard JMX MBeanServer interface or WebLogic Server's strongly typed interface. As we discussed in our last article, the MBeanServer interface allows you to write Java applications that manage any JMX-compliant application through its weakly typed, Java reflection–style interface. While this interface is perfect for tool vendors that want to work with a wide variety of applications and discover functionality at runtime, it can be more tedious to write and debug when the JMX program is intended to manage a well-known JMX-compliant application. BEA WebLogic Server 8.1 provides a strongly typed interface that is simpler to use and provides better compile-time checking of your code.

Using the MBeanServer Interface

The basic steps to writing a JMX program that uses the MBeanServer interface are:

- Obtain a reference to the MBeanServer implementation
- Determine the MBean(s) of interest
- Determine the MBean's attributes and/or operations of interest
- Invoke the appropriate methods on the MBeanServer to perform the action

When writing a JMX application for managing BEA WebLogic Server, the easiest way to obtain a reference to the MBeanServer implementation is to look it up from the admin server's JNDI tree using the JNDI name weblogic.management.server. Administration MBeans are accessible through the admin server's MBeanServer. All configuration changes for a domain must be made through the admin server's MBeanServer.

Managed servers also have their own MBeanServer. Through these MBeanServers you can access local configuration and runtime MBeans. Currently, the MBeanServers on the managed servers are only accessible through the

WebLogic-specific MBeanHome interface discussed in the next section.

BEA WebLogic Server's MBeanServer supports transparent remote access capabilities through the normal WebLogic RMI mechanisms so your JMX program does not need to concern itself with whether the MBeanServer is in the local or remote process. If you intend to change MBean attributes or invoke operations that modify the domain, you will need to authenticate your application to WebLogic Server with sufficient permissions to do so. Although the JNDI authentication mechanism is deprecated in favor of JAAS-style authentication, we show the older authentication mechanism for brevity (see Listing 1)

To get information about an MBean, you first need to know its object name. JMX uses the javax.management.ObjectName class to represent an MBean's object name. If you don't know what MBeans are available, use one of the MBeanServer's query methods to get a list of the matching MBeans. For example, use the queryNames() method with null arguments to return a java.util.Set containing the ObjectName objects for all registered MBeans, as shown here:

```
Set mbeansSet = mbeanServer.queryNames(null, null);
Iterator mbeans = mbeansSet.iterator();
while (mbeans.hasNext()) {
    ObjectName mbeanName = (ObjectName)mbeans.next();
    ...
}
```

Once you determine the object name for the MBean of interest, you get detailed information about the MBean's attributes and operations by using the MBeanServer's getMBeanInfo() method (see Listing 2).

Finally, you invoke the appropriate method on the MBeanServer to get or set the attribute or to invoke the operation (see Listing 3).

If you know the target JMX application you want to manage, the relevant MBean object names and their attributes and operations of interest, it is possible to make the JMX programming more straightforward if you are willing to give up some of the flexibility. Listing 4 shows a simple JMX program to get the default execute queue's configured number of threads for the domain's admin server.

Notice that we use the fact that there is an MBean of type AdminServer in the default domain to get the domain name and the server name from the related MBeans. With BEA WebLogic Server 8.1, the default domain name is always weblogic so we could have skipped calling getDefaultDomain() and simply used weblogic as the domain name (or omitted it entirely since it is the default) when creating the object name for the admin server MBean.

Using the Strongly Typed WebLogic JMX Interface

BEA WebLogic Server 8.1 also provides an MBeanHome interface that gives you access to the strongly typed interface. You obtain references to a server's MBeanHome implementation by looking it up in JNDI. As was the case with the MBeanServers, administrative MBeans must be accessed through the admin server's special MBeanHome, known as the Administration MBeanHome. To obtain a reference to the Administration MBeanHome implementation, do a JNDI lookup on the admin server using the JNDI name weblogic.management.adminhome. This JNDI name is defined as the ADMIN_JNDI_NAME constant on the MBeanHome interface to help insulate your program from any further JNDI changes.

All servers, including the admin server, also have local MBeanHome implementations that provide access to local configuration and runtime MBeans. These local MBeanHome implementations are accessible by performing a JNDI lookup directly against the server of interest using the JNDI names weblogic.management.home.localhome or weblogic.management.home.<server_name>, where <server_name> is the name of the WebLogic Server instance. The MBeanHome interface defines two additional constants, LOCAL_JNDI_NAME and JNDI_NAME, that you should use in place of weblogic.management.home.localhome and weblogic.management.home, respectively. The admin server also has references to all managed servers' local MBeanHome implementations through the weblogic.management.home.<server_name> JNDI names.

To get a reference to the Administration MBeanHome, use code that looks very similar to the code shown to obtain the MBeanServer:

```
MBeanHome mbeanHome = null;
try {
    ... // Same as earlier example

    mbeanHome =(MBeanHome)
        ctx.lookup("weblogic.management.adminhome");
}
catch (NamingException ne) { ... }
```

The MBeanHome interface provides a wide variety of methods to create MBeans and get different types of MBeans. For example, the getAllMBeans() methods return the type-safe stubs for all MBeans in the specified domain, for which you could then use Java reflection to determine the set of attributes and operations each MBean supports. Of course, you typically choose to use the strongly typed interface because you already know the types of MBeans that you need to manipulate. The getAdminMBean() methods allow you to get a type-safe reference to an administration MBean:

```
String domainName = mbeanHome.getDomainName();
ServerMBean myserver = (ServerMBean)
    mbeanHome.getAdminMBean(”myserver”, “Server”, domainName);
```

Once you have the type-safe reference, you can access the attributes and operation directly:

```
int listenPort = myserver.getListenPort();
```

Listing 5 shows the strongly typed interface version of the same program contained in Listing 4. Notice that we are using the getAdminMBean(name, type, domain) method to locate the MBean references of interest. This interface tends to be much simpler than the JMX standard interface where we had to use ObjectName representations of the MBean's object name. However, it is still a little confusing when you need to supply more than the Name and Type attributes to uniquely identity the MBean because you have to tack on the extra attribute name-value pairs, delimited by commas, to the value of the name attribute. For example, the following code snippet from Listing 5 shows the form of the name argument where we also have to specify the Server attribute to uniquely locate the ExecuteQueueMBean of interest:

```
String defaultExecuteQueueName =
    "weblogic.kernel.Default,Server=" + adminServerName;
ExecuteQueueMBean defaultExecuteQueueMBean = (ExecuteQueueMBean)
    mbeanHome.getAdminMBean(defaultExecuteQueueName,
                            "ExecuteQueue", domainName);
```

We hope you can see that the type-safe version of the program is simpler and provides better compile-time checking. If you need to write JMX programs to automate the management of your BEA WebLogic Server–based applications, we recommend using the type-safe interface in most circumstances. If you are building JMX management tools to work with a variety of JMX-compliant application, then using the JMX MBeanServer interface will make your job easier.

Summary

In this article, we showed you the basics of the two different JMX programming interfaces that are available to build JMX management programs for BEA WebLogic Server 8.1. The JMX standard MBeanServer interface provides a loosely typed, reflection-style interface that allows tool vendors to write tools that discover MBeans and their attributes and operations at runtime. The strongly typed WebLogic JMX MBeanHome interface provides a simpler interface for building JMX management programs to perform predefined tasks with a WebLogic Server–based application.

The next article in this series will dive into the more advanced Java APIs for building custom JMX programs that use JMX notification with monitors and timers. Our final installment will discuss creating custom MBeans and extending the Admin Console to display them.

Listing 1: Authentication mechanism

```
MBeanServer mbeanServer = null;
try {
    Hashtable props = new Hashtable();
    props.put(Context.INITIAL_CONTEXT_FACTORY,
              "weblogic.jndi.WLInitialContextFactory");
    props.put(Context.PROVIDER_URL, “t3://AdminServer:7001”);
    props.put(Context.SECURITY_PRINCIPAL, “weblogic”);
    props.put(Context.SECURITY_CREDENTIALS, “weblogic”);
    InitialContext ctx = new InitialContext(props);
    mbeanServer =(MBeanServer)
        ctx.lookup("weblogic.management.server");
}
catch (NamingException ne) { ... }
```

Listing 2: getMBeanInfo method

```
MbeanInfo mbeanInfo = null;
try {
    mbeanInfo = mbeanServer.getMBeanInfo(mbeanName);
}
catch (InstanceNotFoundException infe) { ... }
catch (IntrospectionException ie) { ... }
catch (ReflectionException re) { ... }

MBeanAttributeInfo[] attributes = mbeanInfo.getAttributes();
MBeanOperationInfo[] operations = mbeanInfo.getOperations();
```

Listing 3: Invoke the method

```
MBeanAttributeInfo attribute = attributes[j];
String attributeName = attributes.getName();
if (attribute.isReadable()) {
    Object attributeValue = null;
    try {
        attributeValue =
            mbeanServer.getAttribute(mbeanName, attributeName);
    }
    catch (MBeanException mbe) { ... }
    catch (AttributeNotFoundException anfe) { ... }
    catch (InstanceNotFoundException infe) { ... }
    catch (ReflectionException re) { ... }

    ...
}
```

Listing 4: JMXSample.java

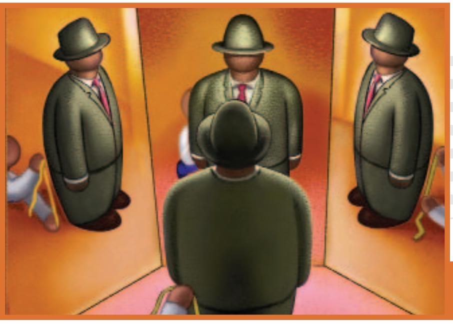
```
package wldj;

import java.util.Properties;
import javax.management.AttributeNotFoundException;
import javax.management.InstanceNotFoundException;
import javax.management.MalformedObjectNameException;
import javax.management.MBeanException;
import javax.management.MBeanServer;
import javax.management.ObjectName;
import javax.management.ReflectionException;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

public class JMXSample
{
    private static void usage()
    {
        System.out.println("Usage: java wldj.JMXSample <admin_url>
[<username> <password>]");
    }

    private static void handleException(String message, Exception e)
    {
        System.err.println(message + ": " + e.getMessage());
        e.printStackTrace();
        System.exit(1);
    }

    public static void main(String[] args)
    {
        String url = null;
        String username = null;
        String password = null;

        switch (args.length) {
        case 3:
            username = args[1];
            password = args[2];
        case 1:
            url = args[0];
            break;
        default:
            usage();
            return;
        }

        Properties props = new Properties();
        props.put(Context.INITIAL_CONTEXT_FACTORY,
                  "weblogic.jndi.WLInitialContextFactory");
        props.put(Context.PROVIDER_URL, url);
        if (username != null) {
            props.put(Context.SECURITY_PRINCIPAL, username);
            props.put(Context.SECURITY_CREDENTIALS, password);
        }

        try {
```

Create software so brilliant
it can manage itself.

```
            InitialContext ctx = new InitialContext(props);
            MBeanServer mbeanServer = (MBeanServer)
                ctx.lookup("weblogic.management.server");

            String defaultDomain =
                mbeanServer.getDefaultDomain();
            ObjectName adminServerMBeanName =
                new ObjectName(defaultDomain +
                ":Name=Admin Server,Type=AdminServer");
            Object domainMBean =
mbeanServer.getAttribute(adminServerMBeanName, "ActiveDomain");
            ObjectName domainMBeanName = new
ObjectName(domainMBean.toString());
            String domainName =
(String)mbeanServer.getAttribute(domainMBeanName, "Name");
            Object adminServerMBean =
mbeanServer.getAttribute(adminServerMBeanName, "Server");
            ObjectName serverMBeanName = new
ObjectName(adminServerMBean.toString());
            String adminServerName =
(String)mbeanServer.getAttribute(serverMBeanName, "Name");

            String defaultExecuteQueueName = domainName + ":Name=weblogic.kernel.Default,Server=" + adminServerName + ",Type=ExecuteQueue";
            ObjectName defaultExecuteQueueMBeanName =
                new ObjectName(defaultExecuteQueueName);
            Integer threadCount =
(Integer)mbeanServer.getAttribute(defaultExecuteQueueMBeanName,
"ThreadCount");
            System.out.println("Server " + adminServerName +
                                  " in domain " + domainName +
                                  " has " + threadCount +
                                    " threads configured for the " +
                                  "weblogic.kernel.Default" +
                                  " execute queue");
        }
        catch (NamingException ne) {
            handleException("JNDI Error", ne);
        }
        catch (MalformedObjectNameException mone) {
            handleException("Bad Object Name", mone);
        }
        catch (AttributeNotFoundException anfe) {
            handleException("Attribute Not Found", anfe);
        }
        catch (InstanceNotFoundException infe) {
            handleException("Instance Not Found", infe);
        }
        catch (MBeanException me) {
            handleException("MBean Error", me);
        }
        catch (ReflectionException re) {
            handleException("Reflection Error", re);
        }
    }
}
```

Listing 5: WL_JMXSample.java

```
package wldj;

import java.util.Properties;
import javax.management.InstanceNotFoundException;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import weblogic.management.MBeanHome;
import weblogic.management.configuration.AdminServerMBean;
import weblogic.management.configuration.ExecuteQueueMBean;
import weblogic.management.configuration.ServerMBean;

public class WL_JMXSample
{
    private static void usage()
    {
        System.out.println("Usage: java wldj.WL_JMXSample <admin_url>
[<username> <password>]");

    }

    private static void handleException(String message, Exception e)
    {
        System.err.println(message + ": " + e.getMessage());
        e.printStackTrace();
        System.exit(1);
    }

    public static void main(String[] args)
    {
        String url = null;
        String username = null;
        String password = null;

        switch (args.length) {
        case 3:
            username = args[1];
            password = args[2];
        case 1:
            url = args[0];
            break;
        default:
            usage();
            return;
        }

        Properties props = new Properties();
        props.put(Context.INITIAL_CONTEXT_FACTORY,
                    "weblogic.jndi.WLInitialContextFactory");
        props.put(Context.PROVIDER_URL, url);
        if (username != null) {
            props.put(Context.SECURITY_PRINCIPAL, username);
            props.put(Context.SECURITY_CREDENTIALS, password);
        }

        try {
            InitialContext ctx = new InitialContext(props);
            MBeanHome mbeanHome = (MBeanHome)
                ctx.lookup(MBeanHome.ADMIN_JNDI_NAME);

            String domainName = mbeanHome.getDomainName();
            AdminServerMBean adminServerMBean =
                (AdminServerMBean) mbeanHome.getAdminMBean("Admin
Server", "AdminServer", "weblogic");
            ServerMBean serverMBean =
                adminServerMBean.getServer();
            String adminServerName = serverMBean.getName();

            String defaultExecuteQueueName =
                "weblogic.kernel.Default,Server=" +
                adminServerName;
            ExecuteQueueMBean defaultExecuteQueueMBean =
(ExecuteQueueMBean)mbeanHome.getAdminMBean(defaultExecuteQueueName,
"ExecuteQueue", domainName);
            int threadCount =
                defaultExecuteQueueMBean.getThreadCount();
            System.out.println("Server " + adminServerName +
                                  " in domain " + domainName +
                                  " has " + threadCount +
                                    " threads configured for the " +
                                  "weblogic.kernel.Default" +
                                  " execute queue");
        }
        catch (NamingException ne) {
            handleException("JNDI Error", ne);
        }
        catch (InstanceNotFoundException infe) {
            handleException("Instance Not Found", infe);
        }
    }
}
```



Creating Web Services

USE YOUR EXISTING WSDL TO GENERATE A SERVICE CONTROL

BY **KUNAL MITTAL**

AUTHOR BIO

Kunal Mittal is a consultant for implementation and strategy for Web services and services-oriented architectures. He has co-authored and contributed to several books on Java, WebLogic, and Web services.

CONTACT...

kunal@kunalmittal.com

You can create a Java Page Flow application in BEA WebLogic Workshop that utilizes a Service control generated from a Web service.

This article uses the example of an external Web service listed on the XMethods.net site. We will import the WSDL for the Web service into BEA WebLogic Workshop and see the custom Service control that is generated. We will then consume this control within a simple Page Flow application. We can also create a sample JWS file that allows you to quickly test the control.

Getting the WSDL File from XMethods.net

Visit the XMethods.net site. It lists a number of Web services. My example uses the "Dead Or Alive" Web service. The description (taken right from the listing) is, *"includes methods to search the DeadOrAlive database by name, and search for people born on or died on a date (dd/mm). Also a method to get today's birthdays."*

According to the listing, this Web service runs on .NET technology. The URL to the WSDL is www.abundanttech.com/webservices/deadoralive/deadoralive.wsdl.

There is a link to this WSDL from the Web service listing on the XMethods site. Feel free to use another Web service from that site as you apply this article. You will need the URL to the WSDL file to create the Service control. Or, you can download the WSDL file to your local machine and import it into the Workshop environment. However, in general that is not a good idea as you will not know if the WSDL changes in the future.

Starting WebLogic Workshop

The next step is to set up BEA WebLogic Workshop. I assume that you have basic familiarity with the tool and have the latest release of BEA WebLogic Platform 8.1 installed. If not, you will need to download and install that from the BEA Web site. Launch Workshop and create a new application. Within the application create a new Web project.

Generating the Service Control from the WSDL File/URI

Create a folder in which to store this control. In my example, I named the folder "ServiceControls". Now, create a new Java control and then select Web services. The wizard will prompt you to name the control. Give your control a name like "DeadOrAliveCtrl". The next step is to enter the URI for the Web service. You can point to the URI or to a local copy of the WSDL file. That's it. You're done. WebLogic Workshop will generate a control that consumes this Web service. Figure 1 shows the Service control that is generated for you. If you are interested in it, click the source tab to see the source code.

If you have a different Web service as an example, you will see the corresponding control. The same applies if you want to leverage any other Web service, such as the Amazon.com or Google.com Web services within WebLogic Workshop.

You can now use this control as part of any other WebLogic Workshop application. For this article, we will generate a Java Page Flow from this control. Since this is an automated step, it is a quick and easy way to test the consumption of the Web service and the control generated by Workshop.

Generating a Java Page Flow

This is a one-step process. Right-click on your control and select “Generate Page Flow”. Use the wizard to give your page flow a name such as “DeadOrAlive”, or based on the Web service you selected. Figure 2 shows the generated Java Page Flow.

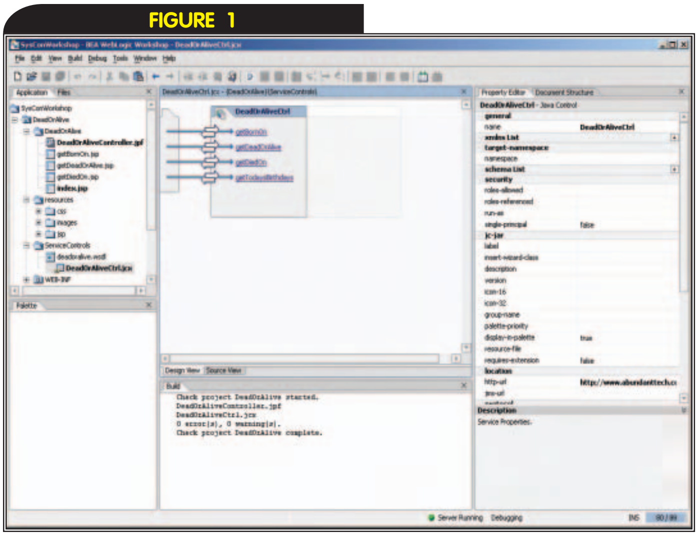
The Page Flow can be used to test your control. Click on the “Start” button from your debug menu. This will start an instance of the BEA WebLogic Server; build and deploy your application, and then load a test browser that allows you to test the operations. Play around a little bit and test the different operations that your control supports. Figure 3 shows the test browser.

One thing you will notice is that the results displayed are meaningless. By default, the index.jsp displays something like

```
ServiceControls.DeadOrAliveCtrl$getDeadOrAliveResult@4d3078
```

for the results. This doesn’t help much. You will need to change the Page Flow code to correct this problem.

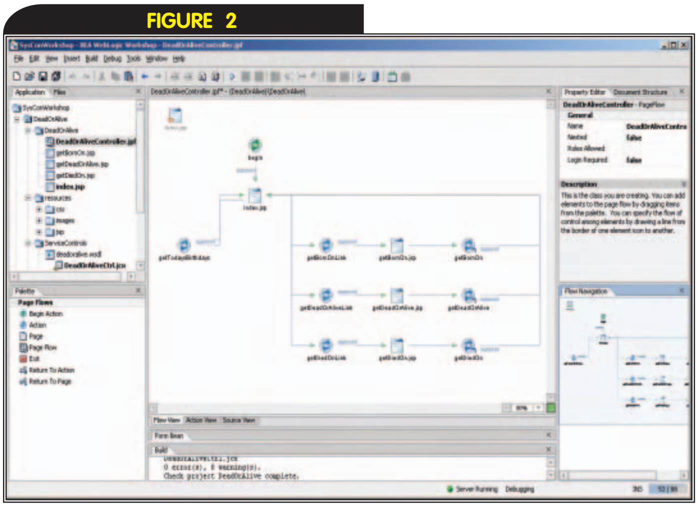
FIGURE 1



```
Check project DeadOrAlive started.
DeadOrAliveController.jpf
DeadOrAliveCtrl.jcx
0 error(s), 0 warning(s).
Check project DeadOrAlive complete.
```

Service control from a WSDL

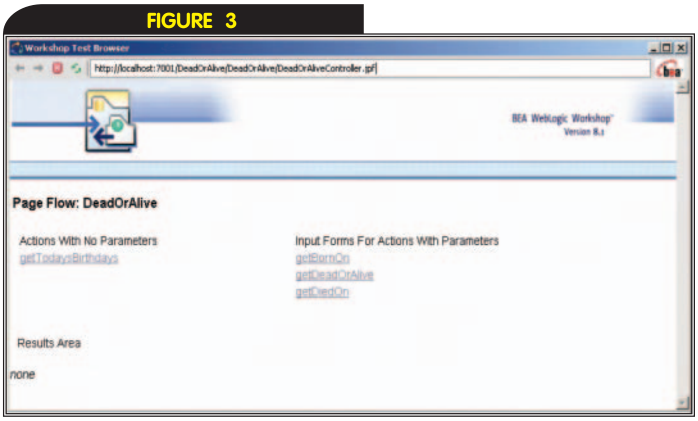
FIGURE 2



```
DeadOrAliveCtrl.jcx
0 error(s), 0 warning(s).
Check project DeadOrAlive complete.
```

Page Flow from the control

FIGURE 3



http://localhost:7001/DeadOrAlive/DeadOrAlive/DeadOrAliveController.jpf

Page Flow: DeadOrAlive

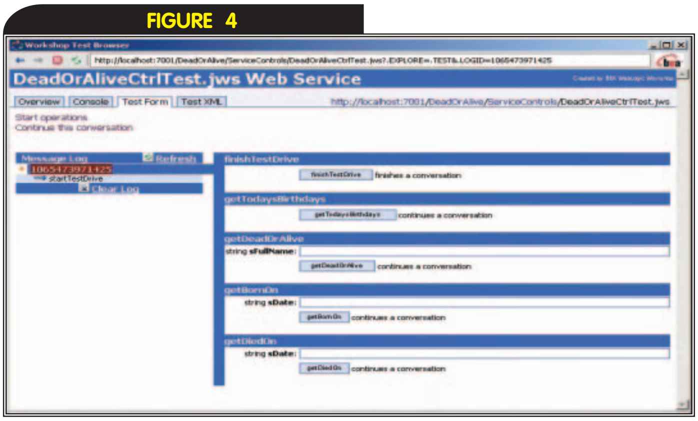
Actions With No Parameters
getTodaysBirthdays

Input Forms For Actions With Parameters
getBornOn
getDeadOrAlive
getDiedOn

Results Area
none

Test browser for the Page Flow

FIGURE 4



http://localhost:7001/DeadOrAlive/ServiceControls/DeadOrAliveCtrlTest.jws?.EXPLORE=.TEST&.LOGID=1065473971425

DeadOrAliveCtrlTest.jws Web Service

http://localhost:7001/DeadOrAlive/ServiceControls/DeadOrAliveCtrlTest.jws

Start operations
Continue this conversation

1065473971425
startTestDrive

finishTestDrive
finishTestDrive finishes a conversation

getTodaysBirthdays
getTodaysBirthdays continues a conversation

getDeadOrAlive
string sFullName:
getDeadOrAlive continues a conversation

getBornOn
string sDate:
getBornOn continues a conversation

getDiedOn
string sDate:
getDiedOn continues a conversation

Test browser for the Web service

Generating a Web Service from the Service Control

Another quick way of testing your control is to use the “Generate Test JWS file” option. This option is available by right-clicking on the control and will generate a simple Web service. Once you have generated the JWS file, select it and then click on “Start” from the Debug menu. This will once again bring up the test browser with the Web service. Click on “Start TestDrive” and then “Continue this conversation”. This is like a handshake between the Web services and now you can execute operations on the Web service you are consuming, from your Web service (see Figure 4).

Summary

In this article I showed you how BEA WebLogic Workshop can be used to consume Web services that are running on other machines. All you need to know is the location of the WSDL file. From that, Workshop can generate everything you need to consume that Web service, in a matter of minutes. This shows you the power, not only of BEA WebLogic Workshop, but of Web services, which enable tools such as Workshop to automate and provide this capability.

References

- *Workshop Help:* http://edocs.bea.com/workshop/docs81/doc/en/core

Modernizing Legacy Systems, PART 1

WHY MODERNIZE?

BY **ANWAR LUDIN**

AUTHOR BIO

Anwar Ludin specializes in service-oriented architectures for the financial sector. He currently works as an independent consultant for financial institutions in Switzerland, where he helps design J2EE architectures based on the BEA WebLogic platform.

CONTACT...

anwar.ludin@agilethinker.com.

When I started working as a WebLogic consultant a couple of years ago, I systematically avoided projects with legacy systems integration. At the time, I was interested only in "pure" J2EE projects. The technology was very new and there was more than enough to do in order to help organizations adopt the J2EE/WebLogic platform as a viable solution for their e-business initiatives.

Today, J2EE has matured and the BEA WebLogic Platform has become an essential part of mission-critical applications in large organizations. But a new trend is also emerging: the need to integrate WebLogic with legacy systems in order to provide a unified, business process-centric IT environment. Moreover, organizations realize that in order to streamline their processes and provide value-added services to their customers, some critical business processes have to be "ported" or "re-hosted" from the legacy system to the WebLogic platform. This is what I call "modernizing" legacy systems. This series will discuss why the process of modernizing legacy systems is becoming an increasingly important business-driven requirement for organizations. I will present a step-by-step, incremental approach in order to minimize risks during the modernization effort. This article is an overview of the entire process and looks at why the modernization effort should be initiated. In part 2, I'll concentrate on integration and "re-hosting" patterns as well as building a common business domain model between the legacy system and the WebLogic platform. In part 3, I'll concentrate on the runtime and workflow aspects of the "modernized" platform.

Defining Modernization

Let's start by defining "modernizing" legacy systems and why such an effort should be initiated. First, what is a legacy system? I'm certain that the definition I'm going to give will make most Cobol programmers very unhappy! My perception of a legacy system is a software platform developed in a procedural language such as Cobol or RPG, hosted on a mainframe or an IBM iSeries. In most cases the maintenance costs of the system tend to increase over time while the available technical expertise around it decreases. Another way of defining a legacy system is the moment where the system cannot service new business requirements because of limitations such as technology and would therefore require modifications far beyond maintenance. However, most organizations have invested enormous amounts of intellectual property and money in these systems and expect to capitalize their return on investment, not counting that most of their mission-critical business processes run on them. So what is modernization? Modernization is the process of modifying a legacy system beyond maintenance in order to provide new added-value services such as enhanced reliability, flexibility, and security.

It is also worth mentioning here what modernization is not. It's not "revamping" or "screen-scraping," which consists mainly of replacing "green-screen" terminals with new presentation interfaces such as Web-enabled thin clients. Therefore, revamping affects presentation only. Modernization is also very different from "replacement," which is an effort to rebuild the system from scratch by a total code-rewrite process. Replacement may be an extremely risky process with a high chance of failure. In our case, the process of modernizing a legacy system involves the process of gradually "re-hosting" specific business processes initially written in Cobol to the BEA WebLogic Platform as J2EE components. The process is incremental and continuous, thus minimizing risks. Reasons why such an initiative could be desirable are:

- Cobol is a procedural language ill-suited for today's enterprise-scale application developments. Therefore, the maintenance costs of legacy systems surpass the modernization costs.
- Modernizing an existing legacy system is less costly and risky than completely replacing it because the process is less disruptive. Besides, organizations can continue to leverage their existing infrastructure during the effort.
- The WebLogic Platform provides the essential building blocks for delivering component-based and service-oriented architectures, essential to today's e-business initiatives.
- Tools such as BEA WebLogic Workshop enhance productivity far beyond the Cobol development environments.
- The Java programming language and the J2EE platform, because of their component-based approach, promote software reuse and facilitate system maintenance in the medium to long term.
- BEA WebLogic Server provides additional services such as scalability, security, and availability that would be hard to implement if created from scratch.

Modernization can take two approaches: in a black-box approach, we are not interested in understanding the inner workings of the legacy system. We simply use interfaces to the system and provide component wrappers to it. In a white-box approach, the inner workings of the legacy code are studied. For example, Cobol modules are examined in order to grasp the underlying business processes and rewrite them as EJB components. Often, the white-box approach might be required to provide true added value and move beyond enterprise application integration.

The Step-by-Step Approach

A modernization effort can span several years. During that time many factors, such as technology, can dramatically change. It is therefore essential to take a step-by-step, incremental approach towards modernizing legacy systems in order to minimize the risks of failure. In my experience, mostly in modernizing banking systems, I have found the following steps useful:

- Build the business case
- Understand the legacy system
- Understand the value-added services provided by the WebLogic platform and define the target architecture
- Prioritize use cases for porting business processes
- Integrate the legacy system with the WebLogic platform
- Port the business processes

Building the business case basically states the rationale and the reason why a modernization effort should be initiated over a less drastic change such as a maintenance effort. For example, for a back-office banking package, the business case could state that B2B business processes require the implementation of a WebLogic Server "front end" to the legacy system and some of the business processes such as portfolio management and straight-through processing should be re-hosted by the WebLogic platform. Basically it emphasizes that the return on investment for modernization will be higher than simply maintaining the legacy system.

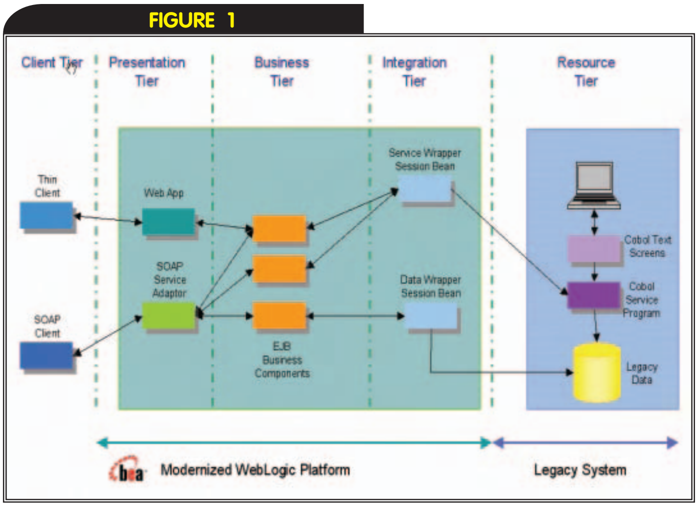
Once the business case has been approved, the next step is to understand the legacy system's architecture. In our banking example, the legacy system consists of a two-tier architecture using "dumb" terminals for the presentation layer and modules written in Cobol hosted on an AS/400 encapsulating the business logic. At this point, after studying the system using a white-box approach, we might discover that some modifications are required in order to provide an integration path to the WebLogic Platform. For example, we might discover that some Cobol modules contain a mix of presentation logic and business logic. We could therefore "refactor" those modules as service programs (a service program is a special kind of Cobol module that can be easily encapsulated and called from Java code) in order to separate business logic from presentation logic. An important point to emphasize is that understanding the legacy system requires the knowledge and help of Cobol programmers and they should be included as stakeholders early in the process. Finally, this step results in a good understanding of the legacy system's architecture as well as possible ways of integrating the legacy system with the BEA WebLogic platform.

Leveraging value-added services provided by the WebLogic platform is usually easier because now we are in the realm of J2EE programming. The objective here is to define a component-based, service-oriented architecture for providing functional and nonfunctional services such as security and availability. However, the major task at this stage of the modernization process is to devise an architectural mapping between the two-tier legacy system and the WebLogic platform. For example, as illustrated in Figure 1, we might decide to build a five-tier architecture with the following tiers:

- Client
- Presentation
- Business logic
- Integration
- Resource

The mapping process will then have to specify which modules of the legacy system will be mapped or "migrated" to specific tiers of the BEA WebLogic Platform as presentation-tier or business-tier components. This architectural transformation is actually a very critical step in the entire process. It has to achieve a certain number of goals, or qualities, in order to provide true added value to

FIGURE 1



Sample architecture

the modernization effort. The target architecture based on the WebLogic Platform should therefore have the following characteristics:

- Service-oriented architecture
- Component-based design
- Layered architecture

A service-oriented architecture publishes discrete business functions and processes through discoverable interfaces. The processes can then be leveraged by other WebLogic applications or services as fundamental building blocks for higher-level services. Componentization and component-based design bring structure to the unstructured legacy Cobol code. Basically, it maps coarse-grained business processes as WebLogic hosted session or entity beans. The achieved added value is a higher level of code reuse as well as an interface-based "plug and play" architecture. Finally, the modernized architecture should be designed in layers where services are implemented on top of components that are in turn implemented using a common business domain class hierarchy.

The next step is to prioritize business processes – gathering a certain number of use cases identified in the legacy system by the project stakeholders as those with the most added value. They should become our initial candidates for the modernization process. In our banking example, the project stakeholders might decide that straight through processing provides enormous added value. They could therefore decide that the order-processing and account-management Cobol modules should be the first ones to be ported to the WebLogic platform as EJB components. At this point we have a development plan and we feel confident that we have identified the most valuable business processes of the legacy system to consider.

The next step is integration, which provides a way of having the legacy system and the modernized WebLogic platform "coexist" in a single, unified business platform. Once again, our objective is to take an incremental approach in order to minimize risks. We cannot afford to do a single "big-bang" replacement of the entire legacy system. Therefore, integration provides a pragmatic, intermediary step. During integration, we usually build adapters in order to "hide" the legacy system. For example we could use coarse-grained component wrappers implemented as EJBs in order to encapsulate the legacy system. Basically, the integration step provides the bridge between the legacy world and the WebLogic Platform. Integration should be transparent so that at any point in the effort, from an external perspective, the modernized WebLogic and legacy systems should provide a unified set of services, independent of their location. From a technical perspective, integration can be done by using different technologies such as JMS, SOAP/Web services, or JCA.

Finally, the last and longest step of the entire process, porting the business processes identified by project stakeholders and implementing them as EJB components, can be initiated. This componentization is a continuing effort and can be separated into two main aspects: code and data migration. The process of code migration consists of "re-creating" some equivalent functionality available in a legacy Cobol module and replacing it with its modernized EJB counterpart. If a whiteboard approach is used, the actual legacy code could be analyzed and the algorithm implemented inside it rewritten in Java as an EJB. In a blackboard approach, only the module's inputs and outputs are analyzed, in other words its functional interface, and a corresponding EJB with an equivalent interface is developed in Java. Both approaches are viable. For example, in a legacy banking system the whiteboard approach might be the only way to modernize an undocumented taxation module. Ever heard of the sentence "the code is the document"? In both cases the overall behavior of the legacy and modernized platforms should remain the same.

Data migration can be a bit trickier. Basically, data migration consists of modernizing the legacy database schema, perhaps because it has become unmanageable over time. For example, the original schema might be implemented using a DB2 database hosted on an AS400. It might be interesting for the modernized system to redesign the schema using entity beans and leveraging WebLogic's container-managed persistence services in a database-independent manner. Data migration can be done in two ways: data migration during code migration and data migration after code migration. Personally, for the sake of keeping a consistent behavior between the modernized WebLogic Platform and the legacy system, I recommend using a data-migration-after-code-migration approach. This will help keep the modernized and legacy systems consistent and in sync without requiring techniques such as data replication. If this approach is taken, then as mentioned under "integration," component wrappers should be used to encapsulate the legacy data.

Conclusion

This article introduced a step-by-step approach to modernizing legacy systems with the BEA WebLogic Platform. Modernization is the process of incrementally and continuously replacing specific modules of a legacy system developed in a procedural language such as Cobol or RPG by EJB components. The objectives of such an effort are to either provide additional business services going beyond maintenance of the legacy system or to leverage the services of the WebLogic Platform such as security, availability, and transaction processing. Modernization is a pragmatic approach in order to minimize risks and often has higher returns on investment than a complete replacement of the legacy system, which is also extremely risky. I have also mentioned some of the qualities, such as a service-oriented architecture and component - based platform, that the modernized architecture should exhibit in order to maximize reuse as well as return on investment.

This article provided a high-level overview of the entire process. In upcoming articles, I will delve into the details and show how to use specific tools such as BEA WebLogic Workshop to implement the different phases of the process.

JOIN THE EXCHANGE!

"Java, .NET, Web Services, MX, XML & Open Source Technologies & Solutions"

International Developer Conference & Expo

4TH Annual

edge 2004 EAST

KEYNOTE SPEAKERS

Hjelm
ORBITZ

Mendels
MACROMEDIA

Sutor
IBM

DEVELOPMENT TECHNOLOGIES EXCHANGE

February 24-26, 2004

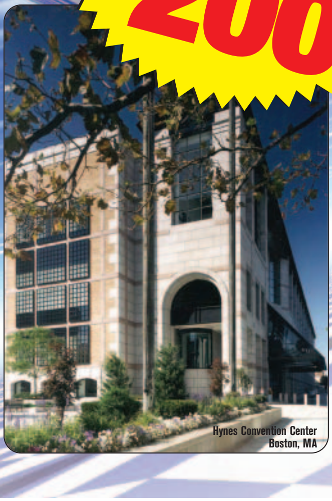
Hynes Convention Center, Boston, MA

ARCHITECTING JAVA, .NET, WEB SERVICES, MX, XML, AND OPEN SOURCE

Tel: 201 802-3058

Fax: 201 782-9651

Register By FEBRUARY 21, 2004 SAVE Up To $200



PROVIDING DEVELOPERS THE LATEST IN:

- Application Integration
- Desktop Java
- Mobility
- ASP.NET
- VS.NET
- JavaServer Faces
- SOA
- Interoperability
- Rich Internet Applications

Preregister for ***FREE*** full-day tutorials featuring the hottest Java, .NET, XML, Web Services, and Security issues. (ADVANCE REGISTRATION REQUIRED)

FREE Exhibition Floor (ADVANCE REGISTRATION REQUIRED)
Meet with industry-leading companies showcasing the newest products and services.

- Product Demonstrations Theater
- Opening Night Reception
- Internet Café
- Case Study Forum

Microsoft .net MX ® WebSphere XML

For more information
visit www.sys-con.com/edge
call 201 802-3058
e-mail events@sys-con.com

Over 100 participating companies will display and demonstrate over 300 developer products and solutions.

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

Over 100 sessions – training, certifications, seminars, case studies, and panel discussions – promise to deliver real-world benefits, the industry pulse, and proven strategies.

Full-Day Tutorials Targeting

- Java
- .NET
- XML
- Web Services
- Open Source
- Security

Conference program available online!
www.sys-con.com/edge

PLATINUM SPONSOR: macromedia® GOLD SPONSORS: Microsoft® ca Computer Associates® SILVER SPONSOR: Novell. CORPORATE SPONSOR: IONA

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

LinuxWorld Magazine · LinuxWorld.com · Java Developer's Journal · ineta · JavaWorld · WebServices Journal · Linux Business Week · XML Journal · OASIS · SD Times · eaindustry consortium · PerfectXML · .NET Developer's Journal · asp.netPRO · WebSphere Developer's Journal · BEA WebLogic Developer's Journal · wireless business&technology · CF Advisor · ColdFusion Developer's Journal · ASP street.com · HSP Street.com · PowerBuilder Developer's Journal · MX developer's journal

SPECIAL INSERT:
Edge 2004 East Conference & Expo February 24 – 26, 2004

SPECIAL INSERT:
Edge 2004 East Conference & Expo February 24 – 26, 2004

International Developer Confe

edge 2004 EAST

DEVELOPMENT TECHNOLOGIES EXCHANGE

February 24-26, 2004

Hynes Convention Center, Boston, MA

ARCHITECTING JAVA, .NET, WEB SERVICES, MX, XML, AND OPEN SOURCE

JOIN THE EXCHANGE!

Become a part of the Development Technologies Exchange in Boston this February. Learn from the leading voices in the i-Technology world, see the latest developments from the companies that push the envelope of innovation, and network with the movers and shakers of IT.

The Edge 2004 East Conference Program holds the information you need to excel in today's competitive environment.

Features & Attractions

- 3 Days Packed with Education and Training
- Keynotes & Panel Discussions from Industry Leaders
- FREE Web Services Security Tutorial
- FREE .NET Tutorial "The Smart Client Perspective"
- Industry-Leading Certification Programs
- "Birds of a Feather" Discussions
- Round Table Discussions
- Opening Day Welcome Reception
- Compelling Case Studies & Best Practices
- Hands-On Labs
- Featured Product Demonstrations
- Exhibit Floor Featuring More than 40 Companies and Hundreds of Products
- Real-time SYS-CON Radio Interviews

Who Should Attend

- Software Developers
- Software Engineers
- CTOs
- CIOs
- Development Managers
- Application Developers
- IT Directors
- Technical Directors
- Analysts
- Consultants
- Programmers
- IT Managers
- Technical Architects
- Team Leaders
- Software Consultants

Register By FEBRUARY 21, 2004 SAVE Up To $200

VISIT www.sys-con.com/edge FOR TIMES and SCHEDULE

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

Keynotes

Chris Hjelm

CHIEF TECHNOLOGY OFFICER, ORBITZ

Tuesday, February 24, 2004 – 10:00 a.m.

Chris Hjelm is chief technology officer for Orbitz and leads the company's technology strategies, engineering, site architecture, and operations. In this role, he is responsible for all aspects of technology, ensuring that Orbitz continues its industry-leading role in providing the lowest cost and most consumer-friendly technology platform in online travel. Before joining Orbitz, Hjelm served as SVP for technology at eBay Inc., overseeing all aspects of its technology. Under his leadership, eBay achieved record levels of site quality and reliability during a period of accelerating global business growth and new product introductions. Chris is also known for his work in creating and advancing world-class technology platforms at high-growth Internet companies as well as in Fortune 500 environments.

David Mendels

GENERAL MANAGER, MACROMEDIA

Wednesday, February 25, 2004 – 10:00 a.m.

SOA+RIA=ROI

Service-oriented architectures are imperative to drive systems and data integration, but for the integration to matter it needs to reach users in meaningful ways. Rich Internet applications (RIAs) combine the richness of the desktop with the reach of the web. Bringing web services and rich Internet applications together fulfills the promise and value proposition of service-oriented architectures, as users benefit from actionable, intuitive, and effective data and information. This next generation enterprise application architecture will deliver solutions that capitalize on existing systems and increase return on investment.

David Mendels is general manager for designer and developer products for Macromedia. Since joining Macromedia in 1992 he's played a major role in the development of Macromedia's product and business strategies and in his current role supervises the gamut of Macromedia's designer and developer products, including Flex, Flash, Dreamweaver, and ColdFusion. In that capacity, he has overall responsibility for the business unit and directly manages product marketing, product management, and product development. Mendels began his career at Macromedia in the international department, moving to Japan to lead the company's field operations, sales, and marketing there. After returning to the United States in 1995, Mendels held positions running business development, alliances and, general management of a number of Macromedia product groups.

Dr. Robert Sutor

DIRECTOR OF WEBSPHERE INFRASTRUCTURE SOFTWARE, IBM

Thursday, February 26, 2004 – 10:00 a.m.

Web Services: Surviving the Mid-Life Crisis

Mid-life can be tough on those who had fast-paced, high-profile youths but suddenly find themselves with dependents and responsibilities put upon them by others. So it is with Web services as it enters the middle phase of its standardization process and moves from being a favorite of the early adopter crowd to being a dependable citizen for the IT mainstream. How is it adapting to the challenge? Have companies like IBM delivered on the early promises of Web services? Do we need to adjust expectations now for Web services so we're not faced with major disappointments later? Dr. Sutor will discuss these and other issues, as well as explain how Web services is a critical part of how companies can become on-demand e-businesses.

Dr. Bob Sutor is IBM's director of WebSphere Infrastructure Software. He is responsible for business and product leadership for the WebSphere Application Server and WebSphere Studio software offerings. Previously, Bob was IBM's director of Web Services Technology, responsible for driving the cross-IBM Web services initiative to continue and advance IBM's leadership in providing Web services solutions, products, and services.

SPECIAL INSERT:
Edge 2004 East Conference & Expo February 24 – 26, 2004

DAY 1 FEBRUARY 24

	Java	.NET	Web Services
8:00	Registration		
9:00	Empowering Java and RSS for Blogging	NET Compact Framework Performance Tips and Tricks	WS-CAF: Standardized Web Services Transactions and Composite Applications
10:00	Opening Keynote – Chris Hjelm, CTO, Orbitz		
11:00	**EXPO OPEN**		
1:00	Keynote Panel Discussion - 21st Century Computing: Is IT Fulfilling its Promise - or Breaking It?		
2:00	Enterprise Architecture & Open Source	Using the Enterprise Instrumentation Framework	Web Services Orchestration, Managemen and Security - Will They Play Together?
3:00	Squeezing Java	J2EE/.NET Interoperability	Session TBA
4:00	Opening Night Reception		

DAY 2 FEBRUARY 25

	Java	.NET	Web Services
8:00	Registration		
9:00	Desktop Java	Best Practices and Techniques for Building Secure ASP.NET Applications	Building Interoperable Web Services
10:00	Keynote Presentation - David Mendels, General Manager of Products, Macromedia		
11:00	**EXPO OPEN**		
1:00	Keynote Panel Discussion - The Open Source Debate		
2:30	Next Phase in Evolution of J2EE	Microsoft Office 2003: A Solutions Platform	Service-Oriented Integration: Making the Right Choices To Support The Next-Generation of Integration
4:00	Aspect Oriented Programming & Java	.NET Framework: Exploring What's New in the Base Class Library for "Whidbey"	Impress the Boss: Roll Your Own Web Services Initiative

DAY 3 FEBRUARY 26

	Java	.NET	Web Services
8:00	Registration		
9:00	Simplifying J2EE Applications	Session TBA	Government Real-Time Fraud Detection Using Web Services
10:00	Keynote Presentation - "Web Services: Surviving the Mid-Life Crisis", Dr. Robert Sutor, IBM		
11:00	Apache Axis	Introduction to BizTalk Server 2004 and Web Services	Exploring the Dark Side
12:00	**BREAK**		
1:00	ANT/JUnit	BizTalk Server 2004 Technical Drilldown	ID, Please. The Case for Giving Web Services an Identity
3:00	Session TBA	Moving your Architecture to .NET	Session TBA

	XML	MX
s	XML Security Integration Challenges	Enterprise Infrastructure for Rich Internet Applications with Macromedia Flex
r,	SOA Foundation Components: Building an XML Content Router	Code Base RIA's
	Real Best Practices for XML Web Services Management and Security	Using Flash with Web Services

Microsoft®

Full Day .NET Tutorial Presented by Microsoft

The Smart Client Perspective

	XML	MX
	XForms - Simplifying the Development of Transactional Web Forms	Using Web Services with ColdFusion
	Session TBA	ColdFusion Components
	Using XML Schemas Effectively in WSDL Design	Rapidly Build Web Services Applications with ColdFusion and Studio

westbridge

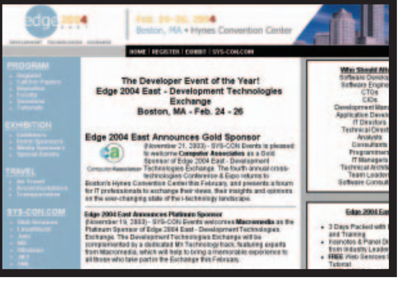
Full Day Security Tutorial Presented by Westbridge Technology

Strategies for Web Services Security Success

	XML	MX
	What's New in XSLT 2.0?	Leveraging Web Services
	Rich Internet Applications: An XML-Based Approach	Session TBA
	Cutting the Crap: Using Rules to Clean Up XML	Code-based Rich Internet Applications with Macromedia Flex
	Session TBA	Session TBA

FOR COMPLETE SESSION DETAILS & INFORMATION VISIT
www.sys-con.com/edge

FOR ASSISTANCE
Tel: 201 802-3058
Fax: 201 782-9651



edge 2004 EAST

Feb. 24–26, 2004
Boston, MA • Hynes Convention Center

HOME | REGISTER | EXHIBIT | SYS-CON.COM

PROGRAM

EXHIBITION

TRAVEL

SYS-CON.COM

The Developer Event of the Year!
Edge 2004 East - Development Technologies Exchange
Boston, MA - Feb. 24 - 26

Edge 2004 East Announces Gold Sponsor

(November 21, 2003) - SYS-CON Events is pleased to welcome Computer Associates as a Gold Sponsor of Edge 2004 East - Development Technologies Exchange. The fourth annual cross-technologies Conference & Expo returns to Boston's Hynes Convention Center this February, and presents a forum for IT professionals to exchange their views, their insights and opinions on the ever-changing state of the i-technology landscape.

Edge 2004 East Announces Platinum Sponsor

(November 19, 2003) - SYS-CON Events welcomes Macromedia as the Platinum Sponsor of Edge 2004 East - Development Technologies Exchange. The Development Technologies Exchange will be complemented by a dedicated MX Technology track, featuring experts from Macromedia, which will help to bring a memorable experience to all those who take part in the Exchange this February.

Who Should Att
Software Develop
Software Engine
CTOs
CIOs
Development Man
Application Devel
IT Directors
Technical Direct
Analysts
Consultants
Programmers
IT Managers
Technical Archit
Team Leader
Software Consult

Edge 2004 Ea
- 3 Days Packed with and Training
- Keynotes & Panel Di from Industry Leader
- FREE Web Services Tutorial

SPECIAL INSERT:
Edge 2004 East Conference & Expo February 24 – 26, 2004

JAVA SESSIONS

Aspect-Oriented Programming & Java

RON BODKIN, NEW ASPECTS OF SOFTWARE

Aspect-Oriented Programming (AOP) has become a hot topic of discussion for enterprise development, with the recent news of support by IBM, JBoss, BEA, Eclipse, and IntelliJ.

- What's real?
- What is AOP?
- What problems can it solve today?
- How can you make an informed decision about using AOP?
- What is the best adoption strategy?
- What are the long-term possibilities for AOP?

This session answers these questions and gives an introduction to AOP for enterprise Java development. AspectJ is a seamless AOP extension to Java. It allows the consistent and flexible implementation of crosscutting concerns such as security, exception handling, testing, logging, caching, and business rules. This is a major improvement on scattered and tangled traditional implementations.

This tutorial introduces AOP and shows how to use AspectJ effectively in an enterprise context, including examples of tools support. Attendees should have experience with OO design and Java development. AspectJ is an open source project available from the Eclipse consortium.

BIO: Ron Bodkin is the founder of New Aspects of Software, which consults and trains on enterprise Java applications with an emphasis on Aspect-Oriented Programming. Ron is also a member of AspectMentor, a consortium of AOP experts. Ron previously worked for the AspectJ group at Xerox PARC, where he led the first AOP implementation projects and training for customers. Prior to that, Ron was founder and CTO at C-bridge, a consultancy that delivered enterprise applications using Java and XML frameworks.

Squeezing Java

ALAN WILLIAMSON, JAVA DEVELOPER'S JOURNAL

Java is a very powerful language, and while it offers the developer a rich array of tools, the fundamentals mustn't be overlooked. Improving your code at the core layer will result in great improvements in efficiency and produce (hopefully) fewer bugs. We'll look at the dos and don'ts of programming and learn lots of hints and tips that will accelerate your Java coding.

BIO: Alan Williamson is SYS-CON Media's technical director, and editor-at-large of *Java Developer's Journal*. In addition, he holds the CTO office at n-ary (consulting) Ltd, one of the first companies in the UK to specialize in Java at the server side.

Enterprise Architecture & Open Source

JAMES MCGOVERN, THE HARTFORD FINANCIAL SERVICES GROUP, INC.

Use of open source software within the enterprise is gaining momentum. The vast majority of organizations use some form of open source software, such as Linux, Apache, and JBoss, in production environments. The enterprise architecture, however, needs to incorporate the best thinking of the industry that not only includes using open source but contributing to it. The model in which open source software gets developed has practices that could assist an organization in becoming agile in their software development practices and allow them to develop software faster, with lower costs and better quality.

In this session you will learn:

- Two models of development: the cathedral and the bazaar
- Value proposition of using open source
- Harnessing the power of the mob: the value proposition of contributing to open source
- Making the build versus buy decision: additional thoughts

BIO: James McGovern is an industry thought leader and the author of the best-selling book, *A Practical Guide to Enterprise Architecture* (Prentice Hall). He is working on two upcoming books entitled *Agile Enterprise Architecture* and *Enterprise SOA*. He is employed as an enterprise architect for The Hartford Financial Services Group, Inc. He holds industry certifications from Microsoft, Cisco, and Sun. He is member of the Java Community Process and of the Worldwide Institute of Software Architects.

J2EE v1.4

BILL ROTH, E.PIPHANY

Day-to-day work with deadlines makes it difficult to keep abreast of the rapidly evolving landscape of J2EE, especially given the numerous constituent J2EE technologies. J2EE v1.4 is chockfull of new services that affect and benefit a wide range of enterprise development tasks. This talk will extract core material from the speaker's new *J2EE Developer's Handbook* and describe what's embodied by J2EE v1.4. The new Web services features provided by J2EE v1.4 in particular will be highlighted. The talk will also briefly address those services missing from the current J2EE standards but still needed when building enterprise applications.

BIO: Bill Roth is senior technical evangelist for E.piphany. With over 15 years in this industry, he has played numerous roles in engineering and product marketing including program manager for Java IDL, and managing product marketing and product management for J2EE 1.2 at Sun.

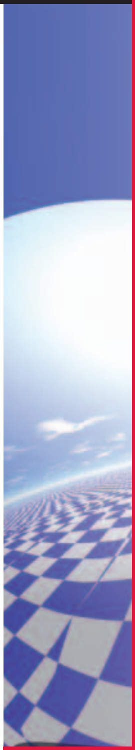
Apache Axis

CHRIS HADDAD, BURTON GROUP

Apache Axis is the very popular SOAP engine, which includes everything you need to start producing Web services. Discover just what Axis is and how you can utilize the power of this free engine to kick-start your Web services.

BIO: Chris Haddad has 13 years experience in Web services, protocols and standards (SOAP, UDDI, WSDL), application framework, development tools, key development languages (Java, C++, Perl, JavaScript), and enterprise application integration. His background includes Grand Central, Flamenco Networks, Adjoin, Securant, Employease, Jamcracker, TRX, and Raygun.

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.



Empowering Java and RSS for Blogging

JASON BELL

One of the fastest growing areas over the last few years is that of the blogging community. The ease at which you can post and publish information has enabled everyone to become his or her own publisher. One power of blogs has been the syndication of data via the RSS (XML) protocol. Discover how you can easily produce and consume RSS feeds within your Java applications for wider appeal and hook into the likes of JavaBlogs.

BIO: Jason Bell is a Java developer and IT development manager for a B2B portal in the UK. He also contributed to a number of open source projects and is an advocate of everyone reading the API docs.

Ant/JUnit

KYLE GABHART

A defined and easily repeatable process is one of the most necessary but often least-used aspects of good software development. A defined build process ensures that your project's software is built, deployed, and tested identically each time. Without this type of control and predictability, valuable time is often lost chasing down bugs that don't exist or rejecting solutions that were only partially implemented.

A critical measure of the success of software can be found in whether or not it executes successfully. Equally important, however, is whether or not that software does what it was intended to do. JUnit is an open source–testing framework that provides a simple means for developers to define their intentions for how their software should work. JUnit then provides test runners that process your intentions and verify that your code performs as intended. The result is software that not only works, but works in the correct way.

Apache's Ant is a powerful scripting tool that enables developers to define and execute routine software development tasks using the simplicity and extensibility of XML. Ant provides a comprehensive mechanism for managing software development projects, including compilation, deployment, testing, and execution. In addition, it is compatible with any IDE or operating system.

BIO: Kyle Gabhart is an independent consultant, mentor, and published author, specializing in J2EE, XML, and Web services technologies. He is a prolific writer, with his most recent work displayed on IBM's developerWorks Web site in the J2EE Pathfinder column. You can find him on the Web at www.Gabhart.com.

FULL DAY JAVA TRAINING

CREATE SERVER-BASED SOLUTIONS USING JAVA 2 ENTERPRISE EDITION

Developing J2EE Applications Using WebSphere Studio Application Developer*

February 25th & February 26th

J2EE is a standard coherent Java API for creating server-based applications, and WebSphere is an industry-leading enterprise application server. Students learn the steps necessary to develop J2EE applications (built using database, servlets, JavaServer Pages, and Enterprise JavaBeans) using WebSphere Studio Application Developer. Hands-on labs reinforce the lecture topics while providing practical experience.

BIO: Ken Greenlee is the chief technology officer of Kenetiks, Inc., a Java-WebSphere training and consulting company he founded in 1997. He has written many courses covering J2SE, J2EE, WebSphere Studio Application Developer, WebSphere Application Server, and WebSphere Portal Server. Prior to founding Kenetiks, he worked for IBM in North Carolina where he worked on development teams responsible for such products as VisualAge Smalltalk and OS/2. He holds a BS degree in computer science from Indiana University along with numerous professional certifications in Java and WebSphere.

*Available for Gold Pass + Training Pass registrants only.

Desktop Java

JOE WINCHESTER, IBM

Java is enjoying a renaissance on the desktop, and there are several reasons for this:

- The issues that plagued early client/server projects or Java desktop applications have largely been solved. Swing 1.4.2 delivered great performance improvements, good fidelity XP, and GTK look-and-feel classes. Java Web Start now exists as a way to deploy to a client PC the programs that run in a local JRE and enjoy the benefits of local caching, lazy update, and execution within Java's security model
- Java is becoming part of the default installation setup for many PC vendors such as Dell, HP, and Gateway.
- SWT provides a set of rich native controls over and above those offered by AWT, and many Java developers are finding that it's a useful GUI toolkit to use in their end-user applications.
- HTML is no longer being viewed as the only viable client for application development. A number of hybrid technologies that combine the benefits of server-side computing with a client GUI have arisen, including Ultra Lightweight Client, Droplets, JFlash, Macromedia Royale, Asperon, and Thinlets.

This session will look at the Java desktop space and discuss the issues and technologies, as well as what is at stake if Java can or can't recapture its lost pride as a client platform.

BIO: Joe Winchester is a software developer working on WebSphere development tools for IBM in Hursley, UK. Joe was part of the expert group for JSR 57.

VISIT www.sys-con.com/edge FOR TIMES and SCHEDULE

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

.NET SESSIONS

.NET Compact Framework Performance Tips and Tricks

JIM WILSON, JW HEDGEHOG, INC.

Learn the techniques that can be used to increase the responsiveness of user interface and network operations for users of applications built on the .NET Compact Framework. Look under the covers at advances and changes in the "Whidbey" release that significantly improve performance. Get a general overview of how the .NET Compact Framework works under the hood at runtime, with specific focus on performance implications. Next, we cover general user interface tips to increase performance. Explore how asynchronous infrastructure, such as threading, in the .NET Compact Framework can be leveraged to optimize both user interface and network operations. Learn about the architectural guidelines for creating applications that perform well under frequently changing network conditions.

BIO: Jim Wilson is president and cofounder of JW Hedgehog, Inc., a consulting firm specializing in the integration of smart devices as part of enterprise solutions with a focus on the .NET Compact Framework, .NET Framework, Web services, and other XML-related technologies. He develops and teaches DevelopMentor's Mobility training course, was one of Microsoft's original five MVPs for the .NET Compact Framework, has written articles for numerous publications, and has spoken at a myriad of events, including Microsoft's Tech-Ed and Mobility Developers conferences.

Best Practices and Techniques for Building Secure ASP.NET Applications

PATRICK HYNDS, CRITICALSITES

When the enterprise depends on your application, careful attention to security is essential. This session provides specific recommendations to follow when developing secure ASP.NET Web applications and services, and focuses on the details of configuring IIS for security. Understand how to use authentication, authorization, threat modeling, configuration settings, and secure database access to create secure systems, plus learn common coding techniques for storing secrets, error handling, data validation, and code access security.

BIO: Patrick Hynds, MCSD, MCSE+I, MCDBA, MCSA, MCP+Site Builder, MCT, is the Microsoft Regional Director for Boston and the CTO for CriticalSites. Named by Microsoft as the Regional Director for Boston, he has been recognized as a leader in the technology field. An expert on Microsoft technology (with at last count 54 Microsoft certifications) and experienced with other technologies as well (WebSphere, Sybase, Perl, Java, Unix, Netware, C++, etc.), Patrick previously taught freelance software development and network architecture. He has been a successful contractor who enjoyed mastering difficult troubleshooting assignments. A graduate of West Point and a Gulf War veteran, Patrick brings an uncommon level of dedication to his leadership role at CriticalSites. He has experience in addressing business challenges with blended IT solutions involving leading-edge database, Web, and hardware systems. In spite of the demands of his management role at CriticalSites, Patrick stays technical and in the trenches, acting as project manager and/or developer/engineer on selected projects throughout the year.

Using the Enterprise Instrumentation Framework

DEREK FERGUSON, EXPAND BEYOND CORPORATION

The Microsoft .NET Framework 1.1 and Windows Server 2003 offer a number of new features to help developers instrument their code. In this session, you'll learn about the challenges facing application management in today's distributed world. We will examine the new unified instrumentation API in the Enterprise Instrumentation Framework (EIF), including the new Windows Event Trace available in Windows Server 2003, configurable at-source event filtering, and how request-based event tracing using EIF allows you to put a request context around the trace messages that map to a business process flow in your application. We will also discuss the benefits to both the developer and the application administrator for using EIF in your application.

BIO: Derek Ferguson is chief technology evangelist for Expand Beyond Corporation (www.xb.com), the worldwide leader in mobile software for enterprise management. He is also editor-in-chief of *.NET Developer's Journal* and author of the book Mobile .NET (Apress).

.NET Framework: Exploring What's New in the Base Class Library for "Whidbey"

JULIA LERMAN

The base classes serve as the essential libraries for any developer. Continued evolution of the base classes provides numerous benefits, including the ability to write more reliable, faster solutions, easier-to-write code, and more solutions entirely in managed code. Take a look at the many features that are a part of that evolution, including features in IO, event-logging, and various features in System.

BIO: Julia Lerman is the owner of Julia Lerman, Inc (dba The Data Farm). and has been developing software applications and consulting to businesses since 1984. Julia is the founder and leader of VTdotNet, Vermont's .NET user group (www.vtdotnet.org), as well as a member of the User Group Relations Committee for INETA (International .NET Association) www.ineta.org. She writes on Visual Studios and .NET and in her pre-Visual Basic days, FoxPro Advisor. Having spent many years as a FoxPro. Julia is working very hard to help bring the same community spirit to .NET that is so unique to FoxPro.

Microsoft Office 2003: A Solutions Platform

JOHN HOLLINGER, INTERNOSIS

For all developers who would like to integrate custom business solutions with Microsoft Office products, this session will introduce you to the expanded developer features that have been included in the newest version of Microsoft Office. Come explore new XML-based programmability in everything from Word 2003 and Excel 2003 to FrontPage 2003 and SharePoint. Build powerful, modular solutions with Web services. Learn about InfoPath 2003 support for XML standards. Discover how to use the Microsoft Visual Studio Tools for the Microsoft Office System to automate and extend Microsoft Office Word 2003 and Microsoft Office Excel 2003 using Visual Basic .NET and Visual C# .NET. More than ever, Office has a solution for you.

BIO: John V. Hollinger serves as CTO of Internosis clients are able to leverage the latest technologies while planning for tomorrow's challenges. Mr. Hollinger is responsible for the performance of the Northeast/Commercial business unit, including all aspects of business development, client care, marketing, resource management, and financial management. Prior to joining Internosis, he played an instrumental role in the creation of New Technology Partners (NTP) and then went on to run a highly successful division of Xpedior, where his team was a four-time Microsoft Solution Provider Partner of the Year in New England and the recipient of the award on a worldwide basis in 1997. Previously, John was vice president of Advanced Technologies at AIG.

Advanced BizTalk Server 2004

BRUCE BACKA, CRITICALSITES

BizTalk Server 2004 is designed to provide Enterprise Application Integration (EAI), Business Process Automation (BPA), and Information Worker Integration solutions. Come join us for a technical drilldown on the new features and toolsets available.

BIO: Bruce Backa is the founder of CriticalSites. A noted business leader and consultant in the IT industry, he has acted as chief architect, technologist, and project manager for assignments involving large-scale technology and implementation strategies. He has held the positions of director of Technology and Business Research for the American Stock Exchange (AMEX) and director of Technology for American International Group. Bruce has been responsible for the architecture, implementation, and management of a worldwide client/server networking infrastructure for a Fortune 10 company with a platform of over 600 servers connecting 10,000 users across 50 cities throughout North America and offshore. In 1994 he founded NTP Software, a provider of business solutions for Windows NT and Windows 2000. At the recent World Economic Forum in Switzerland, Bruce was recognized as a technology pioneer. This follows a similar award from the National Computer Conference in 1974, where he was honored as a part of the Dartmouth College team that invented computer timesharing.

Moving your Architecture to .NET

CARL FRANKLIN, FRANKLINS.NET

This session is focus on how to migrate existing business components from VB6 COM Objects to VB.NET assembly components. We'll spend time discovering how to best move different tiers of a multi-tiered application from COM to .NET, as well as effective strategies on how to wrap existing COM components for interoperability. We'll also examine best practices for moving your application from a COM-based architecture to a .NET-based architecture.

BIO: Carl Franklin has been a luminary in the VB community since the very early days. Since then he has written extensively on the subject of VB. He has authored two books for John Wiley & Sons on sockets programming in VB, and in 1994 he helped create the very first Web site for VB developers, Carl & Gary's VB Home Page. He now teaches hands-on VB.NET classes for his company, Franklins.NET. He has taught developers from Citigroup, Aetna, Fidelity Investments, Fleet Bank, Foxwoods Casino, UTC, Hubbell, Microsoft, Mohegan Sun Casino, and Northeast Utilities, to name a few. Carl is cohost of a weekly talk show on his Web site for .NET programmers, called ".NET Rocks!" Carl is MSDN Regional Director for Connecticut.

Introduction to BizTalk Server 2004

BRIAN LOESGEN, NEUDESIC

BizTalk Server 2004, the latest and third version of BizTalk, is an exciting tool that all .NET application developers should know about. BizTalk fills two key business needs: enterprise application integration and business process automation. The powerful messaging platform allows for easy integration between applications, and also between enterprises. The business process automation engine, based on the new BPEL XML specification, allows for automation of even the most complex business processes. Throw in InfoPath for a user interface into BizTalk, tight integration with Visual Studio, Sharepoint, and Office-based monitoring and reporting tools, and you get an incredibly powerful tool.

This session will be an overview of BizTalk Server 2004. Concepts and tools will be demonstrated by walking through several real-world, practical examples of BizTalk usage. Particular attention will be paid to Web service integration and orchestration, showing why BizTalk Server 2004 deserves consideration as the integration "glue" in a services-oriented architecture.

BIO: Based in San Diego, Brian Loesgen is a principal consultant with Neudesic, a premier southern California solution provider and system integrator. At Neudesic, Brian uses his expertise to architect and deliver advanced Microsoft technology solutions, and to translate new, leading-edge technologies into real-world value. He is a coauthor of the *Professional XML*, *Professional ASP/XML*, *Professional Windows DNA*, *Professional ASP.NET Web Services*, and *Professional VB.NET Web Services* books from Wrox, and is currently working on the *BizTalk Server 2004 Developers Guide*, due in 2004. In addition, Brian has written technical white papers for Intel, Microsoft, and others. Brian has spoken at numerous major technical conferences worldwide and is a cofounder and president of the International .NET Association (ineta.org). He coleads the San Diego .NET user group, and leads the San Diego Software Industry Council Web Services SIG.

In his spare moments, Brian enjoys outdoor activities such as cycling, hiking in the mountains, kayaking, camping in the desert, or going to the beach with his wife Miriam and children Steven and Melissa.
Brian can be reached at brian.loesgen@ineta.org.

Attend a FREE One Day Tutorial

Presented By

Microsoft®

.NET–The Smart Client Perspective

Tuesday, February 24, 2004

For the business world, one of the most exciting promises of the information age is the opportunity to provide employees with access to information and the tools to act on that information whenever and wherever they need to. To a certain extent, that promise has already been achieved. Today, most businesspeople work on PCs that provide access to information, applications, and resources far beyond the boundaries of their local machine.

There are limits, however. Today's Internet model for information and application distribution assumes access to a network connection, but ubiquitous Web connectivity still lies in the future. And some computing tasks require robust functionality that can only be provided efficiently by "rich" client applications that reside on the local computer.

A challenge arises when your organization requires both the flexibility and immediacy that comes with online access to data and applications, plus the full functionality of traditional client software.

The answer: smart client software.

This day-long tutorial focuses on developing and deploying smart client applications.

Free .NET Tutorial when you register for a VIP Pass.
Seating is limited. Full conference attendees will receive priority seating for all tutorials,
all other seating is on a first-come, first-served basis.

VISIT www.sys-con.com/edge FOR TIMES and SCHEDULE

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

WEB SERVICES SESSIONS

WS-CAF: Standardized Web Services Transactions and Composite Applications

ERIC NEWCOMER, IONA TECHNOLOGIES

The Web Services Composite Application Framework is a collection of three specifications – Web Service Context (WS-CTX), Web Service Coordination Framework (WS-CF), and Web Service Transaction Management (WS-TXM) – designed to solve problems that arise when multiple Web services are used in combination ("composite applications") to support information sharing and transaction processing. As coauthor of the specification, Eric will discuss how WS-CAF addresses the underlying issues of Web service context propagation and transaction management to expand the scope, usability, and reliability of Web services for business process automation.

BIO: In the role of chief technology officer at IONA, Eric is responsible for IONA's technology roadmap and the direction of IONA's e-Business Platforms as relates to standards adoption, architecture, and product design.

Government Real-Time Fraud Detection Using Web Services

NEIL MCGOVERN, SYBASE, INC.

Government agencies are faced with increasing amounts of data and are challenged to make sense of, and act on, that data in real time. Failure to interpret and execute on data can result in security threats and, potentially, loss of life. Government agencies are increasingly investing in Web services solutions to address their need for real-time access to information.

The Canadian Passport Office is an example of a government agency leveraging Web services to exchange information in real time to combat terrorism and other illicit uses of fraudulently obtained passports. The agency selected IT consulting firm Pentelar and Sybase, Inc., technologies to electronically authenticate identity document data through the use of Web services and ebXML. Identification tasks performed across 10 provinces and 3 territories previously handled by physical document inspection are now conducted in real time with absolute security and reliability.

In this session, Sybase and Pentelar will discuss this successful pilot project and highlight the ebXML capabilities that enable the Canadian Passport Office to address real-time information exchange.

BIO: Neil McGovern has over 15 years of experience in the software industry. Currently, he is responsible for platform strategy for Sybase's infrastructure product line, including its enterprise-class RDBMS; Sybase ASE; and integration, portal, and tools product lines. Prior to that, Neil was responsible for building the Sybase (New Era of Networks) Adapter suites, commonly recognized as one of the most advanced and complete sets of adapters in the EAI marketplace. Neil developed ERP package expertise as CTO of Convoy Corporation, which specialized in PeopleSoft integration. He led the Development and Professional Services teams for Convoy Corporation, which was purchased by Sybase (New Era of Networks) in 1999 after recording triple-digit growth for three consecutive years.

Service-Oriented Integration: Making the Right Choices To Support The Next-Generation of Integration

DAVID CHAPPELL, SONIC SOFTWARE

Applications are increasingly being developed "built-to-integrate," providing the ability to easily expose key functionality through commonly defined interfaces. Gartner calls this concept SODA, or service-oriented development of applications, fitting into its overall service-oriented architecture landscape. When applied to the ever-present integration challenge, SODA represents a transition to service-oriented integration.

But making the right architectural decisions is absolutely vital to ensuring success with service-oriented integration projects – whether applications were built to integrate or not. Choices at all levels – from application-interface style to overall system architecture – can seriously affect the long-term value derived from integration projects. Application infrastructure products on the market today embody architectural directions that should be carefully examined before making purchase decisions. Any two products that claim to support a "service-oriented architecture" may, in fact, promote radically different architectures.

This presentation will examine the leading choices for supporting service-oriented integration: enterprise service buses (ESBs), integration brokers, and application suite platforms.

BIO: Dave Chappell, VP and chief technology evangelist for Sonic Software, has over 18 years of experience in the software industry. Dave and Sonic Software are involved in many standards organizations, technical committees, and expert groups, including the W3C, OASIS, WS-I, and the Java Community Process. Dave is an active contributor to the Apache Axis project and SOAPBuilders.

Dave also has extensive experience in distributed computing, including message-oriented middleware, CORBA, COM, and Web application server infrastructure. In 2002, Dave received the "Most Outstanding Individual Contributor to the Java Community" award from the editors of JavaPro Magazine.

Securing the Web: What Can Be Done Today

MARK SECRIST, HEWLETT-PACKARD CO.

Security is considered one of the main barriers to the adoption of Web services today. With the proliferation of emerging security standards, there is a lot of confusion over which ones are mature enough to use and how they might fit together. This session will present cover current and emerging security standards for Web services and show how they can be fit together architecturally to address various security concerns. At the conclusion, the audience will have a better understanding of these standards and how to begin securing their own Web services.

BIO: Mark Secrist is a senior software consultant for HP's Developer Resources Organization, with over 15 years of experience working in the software development industry. He provides technical consulting to HP and its customers in the areas of J2EE and Web services development, as well as development strategies for "management-ready" applications. In this capacity, he has written a number of white papers and trade publication articles, presented at a number of industry events, and provided training and consulting to many of our enterprise customers.

Exploring the Dark Side

PETER VARHOL, COMPUWARE CORPORATION

The growing use of services-oriented architectures puts pressure on application developers relying on Web services for key features of their applications. Performance, scalability, and reliability of these components affect the ability of applications to meet service-level agreements, yet can't easily be analyzed as a part of the application when developers have a problem. In fact, the Web service may be on a different software platform than the rest of the application. This session describes how developers can shed light on memory use in Web services written in either .NET or Java, even if they didn't write the code and wrote their own applications on the other platform.

BIO: Peter Varhol is a product manager at Compuware Corporation, where he provides direction for developer tools targeting enterprise applications and Web services. He holds graduate degrees in computer science and mathematics.

Building Interoperable Web Services Using WS-I Basic Profile

KEVIN LIU, SAP LABS, LLC

The promise of Web services interoperability is based on a set of standards, including XSD, SOAP, WSDL, and UDDI. However, early interoperability experiments such as the SOAPBuilder activity has proven that making these technologies seamlessly work together, inter- or intra-enterprise, is a bigger challenge than most of us expected. Most of the Web services technologies are still in the standardization process, and Web services platforms have to be based on pre-standard draft specifications. The inconsistencies and ambiguities left in these initial drafts can easily lead to different interpretations for the same features and substantially compromise the promised interoperability. This presentation explains how WS-I Basic Profile addresses these challenges, and what are the key advises you can get from the profile that can make your Web services interoperable. It explains why the profile disallows SOAP encoding and why RPC style should give way to document style.

BIO: Kevin Liu is an architect of XML Web services technologies at SAP Labs, California. Kevin helps to drive the adoption of Web services technologies in SAP's strategic products, as well as representing SAP in various standards bodies. He has over 10 years of experience in software engineering and in financial industries and holds multiple master's degrees in information management, finance, and economics.

ID, Please. The Case for Giving Web Services an Identity

ASHISH LARIVEE, NOVELL

Without identity management, Web services can be consumed by anyone. The challenge for Web services developers is to provide appropriate access based on the user's identity. As identity management moves into the forefront of technology, directory services will evolve from simple LDAP repositories used for authentication and storage to robust engines that provide identity integration, access management, and policy enforcement. This presentation will discuss how identity management and directory services provide a robust solution for Web services authentication, authorization, and single sign-on.

BIO: With more than nine years of experience in the software industry, Ashish Larivee has designed and developed many enterprise applications across a variety of platforms, including Microsoft, Lotus Notes/Domino, and J2EE. In 1999, Ashish joined SilverStream Software, acquired by Novell in July 2002, and has served in various roles in consulting, development, and technical marketing. In her current role, she helps define the strategy and product direction across Novell's Web Application Development Products.

Web Services Orchestration, Management, and Security - Will They Play Together?

PAUL LIPTON, COMPUTER ASSOCIATES

Web Services orchestration, management, and security are among the principal challenges facing implementers of service-oriented architectures today. There is still much confusion in the IT community about the standards themselves, which are at various stages of maturity. Also, their relevance to enterprise IT and how they might someday be able to effectively work together is often unclear. This session provides a useful overview of standards in these three critical areas of Web services – orchestration/choreography, management, and security – and more importantly, how each affects the other. Attendees will gain practical knowledge and a deeper understanding of future trends and the need to address certain real-world issues in order to create a more cost-effective and agile IT infrastructure.

BIO: Paul Lipton is the Web services technology leader for the field services organization and a technology strategist in the Office of the CTO at Computer Associates. He has been an architect and developer of enterprise systems for more than 20 years, and has worked closely with key CA customers to architect distributed solutions using J2EE, .NET, wireless, and Web services technology. Paul has represented CA in various standards organizations, and has participated in the Java Community Process. He has published magazine articles on many technologies including Web services, Java, .NET, EAI, wireless technology, and distributed systems.

Impress the Boss: Roll Your Own Web Services Initiative

BOB ZUREK, ASCENTIAL SOFTWARE

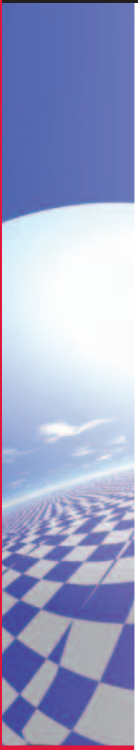
Some companies are leveraging open source software like Linux, JBoss, PHP, MySQL, and Apache SOAP to develop and deploy Web services. As this open source Web services stack continues to mature, companies will benefit beyond simply saving money on license fees; they will be able to assemble the stack and get a Web service up and running quickly as well.

Developers frustrated by their enterprise's lack of Web services interest can immediately master and build Web services at home with open source stack components with little investment except time and a broadband connection. Imagine impressing your company with a demo of wrapping that chunk of code as an open source Web service and calling it from a legacy Visual Basic application.

BIO: Bob Zurek, VP of Advanced Technology, Ascential Software, is responsible for Ascential's overall product strategy. He is instrumental in developing and driving its enterprise integration strategy, including its parallel processing framework, data quality, and Web services strategies. Previously, he was a senior analyst with Forrester Research; prior to Forrester, he held senior management positions at LumaPath, Centive Systems, and Sybase.

VISIT www.sys-con.com/edge FOR TIMES and SCHEDULE

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.



XML SESSIONS

Real Best Practices for XML Web Services Management and Security

KERRY CHAMPION, WESTBRIDGE TECHNOLOGY

Companies deploying Web services in a meaningful way are increasingly finding they need to address Web services management and security early in the architectural phase. Basic Web services connections are easy to do, but managing the security, performance, scalability, and the inevitable changes to the production environment requires some knowledge, expertise, and planning. This session cuts through the hype and outlines real-world mistakes many companies make when deploying Web services and the real best practices from companies who have successfully captured the value of XML Web services. It provides practical advice on how to successfully manage and secure your XML Web service environment.

BIO: Kerry Champion is president and founder of Westbridge Technology, a leading provider of XML Web services management and security solutions. A frequent writer and speaker on XML Web services, Kerry is a renowned technologist who has served in various senior technical and product strategy roles at enterprise security and messaging companies.

SOA Foundation Components: Building an XML Content Router

DAN STIEGLITZ, TIAA-CREF

One of the fundamental components for any burgeoning SOA will be an XML content router. This session explores the concepts, patterns, and open source software available that facilitate building an XML content routing system. The system can be exposed as a Web service or simply as a stand-alone J2EE component for use in your enterprise. The "restaurant" pattern is introduced as the principal design pattern for building the service, and this pattern's applicability to building generic services is discussed. Applying the router as an XML data integration tool is also discussed, as well as its potential for acting as a service orchestrator.

BIO: Visit www.sys-con.com/edge for complete speaker information.

What's New in XSLT 2.0?

STEVE HECKLER, ACCELEBRATE

XSLT 2.0, which may achieve W3C Recommendation status by conference time, offers unparalleled power in conjunction with XPath 2.0 for transforming XML documents. In this engaging, example-rich seminar, Steve Heckler demonstrates the most important new features of XSLT 2.0, including sequences, new data types and XML Schema support, regular expressions, multiple document output, grouping, new control-flow operators, and much more. Current/future support for XSLT 2.0 on the Java and .NET platforms will also be discussed. Most examples will use Saxon, but .NET examples will be included if .NET supports XSLT 2.0 by conference time.

BIO: Steve Heckler has been a leading IT trainer and application developer for nearly a decade. Prior to founding Accelebrate in March 2002, he served nearly seven years as vice president and then president of WestLake Internet Training. He holds bachelor's and master's degrees from Stanford University.

Using XML Schemas Effectively in WSDL Design

CHRISTOPHER PELTZ, HP

Developers building Web services today are beginning to see the value of using the document-style approach over RPC. Recent experience shows that to take full advantage of document style Web services requires a strong knowledge of XML Schemas and related XML standards. This session presents a number of important tips and techniques for properly using XML Schemas in the design of a Web services interface, including important XML-based development tools, binding considerations between XML and underlying objects, WSDL reusability through XML Schemas, and XML Schema naming best practices.

BIO: Chris Peltz is a senior software consultant within HP's Developer Resources Organization. He provides technical and architectural consulting to enterprise customers in the areas of J2EE, Web services, and mobile development. Chris has over 10 years of software experience in object-oriented technologies, 4GL development, GIS, and Web applications design.

Using Rules to Clean Up XML

GARY BRUNELL, PARASOFT

Garbage in, garbage out – it's an axiom that applies to many aspects of enterprise development, but none more so than building reliable and robust Web applications and integration projects with XML. Since its inception, XML has been seen as the cure-all for problems related to Web applications and integration projects. However, poorly written XML can slow down an integration project, or worse, cause the integration project to collapse. The key to successfully using XML in an integration project is to first understand the inefficiencies that may cause poorly written XML, and then apply a rule-based system that establishes policies to follow.

BIO: Gary Brunell joined Parasoft in the fall of 2001 to develop and head the company's professional services division. He is responsible for spearheading delivery services, technical support, and training initiatives as well as establishing process-improvement infrastructure.

XML: Getting Started with Minimum Investment

KETAN PATEL, DATAWATCH

Many organizations are evaluating XML as an enabler for sharing and presenting information. Whether part of a portal strategy, an alternative to EDI for billing, or a data conduit to an information repository, XML is core to many future projects. As IT managers evaluate XML's potential, they should understand its capabilities and limitations. XML is not the panacea for all technology issues. However, as an enabler, XML can provide a common language for sharing information inside and outside your organization. The key is to use the technology in projects that have a clear return and minimize the investments required.

BIO: Datawatch director, Technical Strategy, Ketan Patel has over 15 years of experience developing and marketing technology products to customers who are cautious about technology adoption. During his career, Ketan has focused on bridging the gap between technology and business, and is passionate about bringing information technology solutions to market.

XForms - Simplifying the Development of Transactional Web Forms

ASHISH LARIVEE, NOVELL

XForms is a W3C specification that specifies a declarative language for solving a common requirement for advanced user interaction, data validation, and XML processing. XForms is designed to be integrated into XHTML, but is not restricted to being a part of that language alone. It can be integrated into any suitable markup language. This session gives an introduction to XForms and explains how XForms in the client tier of the J2EE application architecture. In addition, it will cover the benefits of XForms and why it is a perfect fit for interacting with J2EE and Web services. A demonstration of XForms in a J2EE environment using an XForms- compliant browser and a sample application further illustrates the advantages.

BIO: With more than nine years of experience in the software industry, Ashish Larivee has designed and developed many enterprise applications across a variety of platforms including Microsoft, Lotus Notes/Domino, and the J2EE platform. In 1999, Ashish joined SilverStream Software, acquired by Novell in July 2002, and has served in various roles in consulting, development, and technical marketing. In her current role, she helps define the strategy and product direction across Novell's Web application development products.

Rich Internet Applications: An XML-Based Approach

CHRISTOPHE COENRAETS, LASZLO SYSTEMS

Rich Internet applications deliver groundbreaking, highly interactive user interfaces that provide a user experience similar and often superior to desktop applications. In this session, we will discuss the benefits of rich Internet applications, review the market landscape, and explore an XML-centric approach for developing rich Internet applications that can be delivered to any browser today. As a practical example, we will demonstrate how to build a rich Internet application version of an Amazon store using an XML-based presentation server to deliver the rich user interface, and the Amazon Web services API to deliver the content.

BIO: Christophe Coenraets worked at Sybase/Powersoft from 1994 to 2000. He started working with Java in 1996 and became the technical evangelist for the company's Java and Internet Application Division. Christophe then joined Macromedia as the company's JRun and J2EE technical evangelist. He joined Laszlo Systems early in 2003.

XML Security Integration Challenges

PHIL STEITZ, AMERICAN EXPRESS

This session will highlight the key security integration challenges faced by application teams developing enterprise solutions using loosely coupled, XML-based interfaces and service-oriented architectures. We will discuss how emerging standards and technologies address the practical security problems faced by development teams, the gaps that still remain, and the tradeoffs and compromises that architects and developers need to make to implement secure solutions today.

BIO: Phil Steitz is vice president, e-commerce applications development at American Express. Phil has over 20 years of experience as a developer, architect, and technology leader involved in distributed systems development. Before joining American Express, Phil served as a middleware architecture consultant, designing large-scale distributed systems for enterprise customers. Phil holds a PhD in mathematics from the University of Maryland.

Attend a FREE One Day Security Tutorial

Presented By

westbridge

Strategies for Web Services Security Success

Wednesday, February 25, 2004

Are you a developer, software architect, IT operator, or security administrator deploying or planning to deploy XML Web services? If so, this technical seminar is designed to arm you with the practical information and best practices you need to securely deploy XML Web services in your environment. Many questions will be addressed, such as: Why do Web services need special security? What standards are being created and what do they address? How do I leverage my existing environment to secure Web services? What are the different architectural and technical approaches to solving the problem? How can I centrally manage security in a decentralized environment?

Course Highlights/Benefits

- Overview of XML Web services security: Why is it important?
- Discussion of various standards (WS-security, SAML, XML-Enc, XML-Sig, XKMS)
- Architectural considerations
- Malicious Web services attacks
- Strategies for securing XML Web services today and in the future
- "Nuts and bolts" demonstrations of security solutions

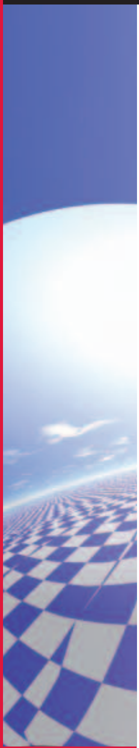
As part of the tutorial, we will show you how to secure your entire XML environment without adding any additional code.

Course leaders

Dan Neiman is a Senior Systems Engineer at Westbridge Technology and is responsible for training and working with real world customers in solving their security and management needs for XML and Web Services. Previously, Dan worked for Tumbleweed Communications where he directed the field engineers in designing and implementing secure email transports. He has also held positions at CSC and the Department of Defense. Dan holds a BS degree in Management and Decision Systems from Carnegie Mellon University.

Peter Bostrom is a Federal Technology Director at Westbridge Technology and works closely with federal customers. Previously, Peter was VP of Product Management at Sequation, a security solutions provider and CTO and co-founder of ExactOne, an Internet-scale distributed search engine. Peter also worked in a variety of sales, consulting and product management roles at Legent Corporation. He was awarded Systems Engineer of the Year award in his first year there. Peter can be heard on National Public Radioís WAMU in Washington, DC as a regular guest on iPublic Interestî hosted by Kojo Nnamdi and is president of the DC Association of Internet Professionals. Peter holds a BA in International Affairs from The American University after spending three years as an Army Ranger.

Free Web Services Tutorial when you register for a VIP Pass. Seating is limited.
Full-conference attendees will receive priority seating for all tutorials; all other seating is on a first-come, first-served basis.



MX SESSIONS

Enterprise Infrastructure for Rich Internet Applications with Macromedia Flex

KEVIN HOYT, MACROMEDIA

Learn how Macromedia's technology initiative, Flex fits seamlessly into today's new service-oriented architectures (SOA). We'll cover design patterns for rich clients, accessing web services and securing your Flex application.

BIO: As a senior sales engineer for Macromedia, Kevin Hoyt plays a primary role in evangelizing the product line, educating the customer base, and supporting the sales organization. Kevin is also a regular speaker at various user group forums throughout the United States.

Leveraging Web Services

TOM JORDHAL, MACROMEDIA

Web Services technology is changing the way we think about designing and building applications. Come and learn what all the fuss is about, find out exactly which problems web services solve, see web services created and used, and even discover how web services expose the world of .NET.

*BIO:*Tom has been a Principal Engineer at Allaire/Macromedia for over 6 years and has spent the last 2 years on the front lines of Web Services. He is one of the implementers of Apache Axis and is the Macromedia representative on the W3C Web Service Description WSDL 1.2 working group. Before getting involved in Web Services, he was the technical lead for the ColdFusion on UNIX products. Tom is currently 'back home' working on ColdFusion.

ColdFusion Components

GLENDA VIGOREAUX, BUSINESS SYSTEMS CONSULTANTS, INC.

ColdFusion Components combine the power of objects with the simplicity of CFML. This is the way object-based development was intended to be, and in this session, you'll learn about this combination first hand. Starting with a simple data-driven application, you'll gradually convert it into a highly scalable and manageable multitier application, and in the process, will amaze yourself at just how easy ColdFusion makes this process.

BIO: Glenda Vigoreaux has been a Certified Macromedia Instructor and Developer for the last 5 years. She is currently heading the web development division for Business Systems Consultants, Inc. the most successful reseller and services provider of software for associations in the greater Chicago area.

Rapidly Build Web Services Applications with ColdFusion and Studio

SIMON HORWITH, ETRILOGY

The last year has shown that web services are not just another passing fad and their promise of platform-independent distributed applications has been realized. Compared to other application server platforms, ColdFusion makes creating web services easy. This session covers how to create a ColdFusion Component (CFC) in Dreamweaver, as well as how to expose that CFC as a web service by just toggling one attribute of the CFC. That's right: in ColdFusion, it is just that easy.

BIO: Simon Horwith is Chief Technology Officer of eTRILOGY Ltd., a software development company based in London, England. Simon has been using ColdFusion since version 1.5 and is a member of Team Macromedia. He is a Macromedia Certified Advanced ColdFusion and Flash D eveloper and is a Macromedia Certified I nstructor. In addition to administering the CFDJ-List mail list and presenting at CFUGs and conferences around the world, Simon has also been a contributing author of several books and technical papers.

Code-based Rich Internet Applications with Macromedia Flex

CHRISTOPHE COENRAETS, MACROMEDIA

Learn how to use Macromedia's technology initiative Flex to create rich internet applications. This session will cover using components, layouts, and managers to build user interfaces as well as using Flexs XML-based language to create and manipulate client-side data models.

BIO: Please visit www.sys-con.com/edge for a complete speaker bio.

Using Macromedia Flash with Web Services

Web services, a technology that allows developers to execute remote procedures, are emerging as a revolutionary tool for web application development. Macromedia Flash MX 2004 Professional is a powerful tool for building applications that consume web services built in any technology, including Macromedia ColdFusion, Java, ASP.NET and PHP. In this session you will explore the visionary computing model that web services represent as you use Macromedia Flash components to develop a web service based application. You will learn how to discover web services, work with data and UI components, perform data binding, examine security issues and aggregate multiple web services into a cutting edge web service consumer.

For a Complete List of MX Sessions please visit www.sys-con.com/edge

RESERVE YOUR HOTEL ROOM NOW AT THE SHERATON BOSTON HOTEL!

THE OFFICIAL CONFERENCE HOTEL OF EDGE 2004 EAST

THE SHERATON BOSTON HOTEL
PRUDENTIAL CENTER
39 DALTON STREET
BOSTON, MA 02199

Arrangements have been made with the Sheraton Boston Hotel, which is conveniently located adjacent to the Hynes Convention Center. Specially reduced rates have been secured at this luxury, full-service hotel.

To learn more about The Sheraton Boston Hotel you can contact the hotel directly or you can make your reservations by calling Expo Travel International at (800) 829-2281 or (201) 444-0060 (direct). Fax reservations to (201) 444-0062. Credit card information is required to guarantee reservations and expedite confirmation. Confirmations will be mailed directly from the hotel.

To make online reservations visit www.expotravel.com

CONTACT INFORMATION:
The Boston Sheraton Hotel Reservations:
Phone (617) 236-2000
Fax (617) 236-1702

HOTEL ARRANGEMENTS ARE EASIER THAN EVER!
You have your choice – contact the hotel directly or call us.

The Official Conference travel agent, Expo Travel International.
Official Conference Travel Agent:
Expo Travel International
Toll Free: (800) 829-2281
Tel: (201) 444-0060
Fax reservations to (201) 444-0062

DRIVING DIRECTIONS TO THE SHERATON BOSTON FROM LOGAN INTERNATIONAL AIRPORT
Take the Sumner Tunnel in the right lane to Storrow Drive (Exit 26). Follow for 2 miles and then take the Fenway Exit (left exit). Bear left following "Boylston Street Inbound" and at the fifth light make a right onto Dalton Street.

VISIT www.sys-con.com/edge FOR TIMES and SCHEDULE

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

erence & Expo

Edge 2004 East
Development Technologies Exchange

For Assistance Tel: 201 802-3058 Fax: 201 782-9651

CONFERENCE: Feb. 24 – 26, 2004 EXPO: Feb. 24 – 25, 2004
Hynes Convention Center, Boston, MA

THREE WAYS TO REGISTER FOR CONFERENCE

1) On the Web: Credit Cards or "Bill Me." Please make checks payable to SYS-CON Events.
2) By Fax: Credit Cards or "Bill Me" 201-782-9651
3) By Mail: 135 Chestnut Ridge Road, Montvale, New Jersey 07645, Attention: Registration

Please note: Registrations are not confirmed until payment is received.

Please complete sections 1, 2, 3 and 4

1 YOUR INFORMATION (Please Print) □ Mr. □ Ms.

First Name ______________________ Last Name ______________________
Title ______________________
Company ______________________
Street ______________________
Mail Stop ______________________
City ______________________
State ______________________ Zip ______________ Country ______________
Phone ______________________
Fax ______________________ E-Mail ______________________

2 PAYMENT METHOD: (Payment in full due with registration)

□ Check or Money Order Enclosed (Registration confirmed upon receipt of payment)

Check # ______________________ Amount of Check $ ______________

Charge my □ Visa □ MasterCard □ American Express □ Discover

Name on card ______________________

Card # ______________________ Exp. Date ______________

Signature ______________________

Billing Address (if different from mailing address)
____

Total Registration fee $____________

3 PLEASE INDICATE YOUR CONFERENCE CHOICE

	Before 12/19/04	Before 1/24/04	Before 2/21/04	Onsite
❑ **GP: Gold Passport Three-Day Conference** Good for all three days of the conference program, including keynotes, all conference sessions & one day of J2EE training.	$1,395.00	$1,495.00	$1,695.00	$1,795.00
❑ **3D: Three Day Conference** Good for all three days of the conference program, including keynotes, all conference sessions (excludes J2EE training).	$1,295.00	$1,395.00	$1,495.00	$1,695.00
❑ **2D: Two Day Conference** Good for two days of the conference program, including keynotes, all conference sessions (excludes J2EE training).	$995.00	$1,195.00	$1,295.00	$1,395.00
❑ **1D: One Day Conference** Good for one day of the conference program, including keynotes, all conference sessions (excludes J2EE training).	$595.00	$695.00	$695.00	$795.00
❑ **TRAINING PASS:** Good for Developing J2EE Applications Java Training Session only. Select one: ❑ February 25th ❑ February 26th	$495.00	$595.00	$595.00	$695.00
❑ **VIP PASS FREE with preregistration** Select one: ❑ FREE - The Smart Client Perspective (Feb. 24) ❑ FREE - Strategies for Web Services Security Success (Feb. 25)	FREE	FREE	FREE	$99.00
❑ **EO: Expo Hall Only** Includes keynotes, expo presentations, and workshops.	FREE	FREE	FREE	$50.00

SPECIAL DISCOUNTS AVAILABLE
Take advantage of the Early Bird and Preregistration values available right now, or save even more with a group of 5 or more. For special group discounts contact Michael Lynch at mike@sys-con.com, or by phone at (201) 802-3058.

4

A. Your Job Title
❑ CTO, CIO, VP, Chief Architect
❑ Software Development Director/Manager/Evangelist
❑ IT Director/Manager
❑ Project Manager/Project Leader/Group Leader
❑ Software Architect/Systems Analyst
❑ Application Programmer/Evangelist
❑ Database Administrator/Programmer
❑ Software Developer/Systems Integrator/Consultant
❑ Web Programmer
❑ CEO/COO/President/Chairman/Owner/Partner
❑ VP/Director/Manager Marketing, Sales
❑ VP/Director/Manager of Product Development
❑ General Division Manager/Department Manager
❑ Other (please specify) ______________

B. Business/Industry
❑ Computer Software
❑ Computer Hardware and Electronics
❑ Computer Networking & Telecommunications
❑ Internet/Web/E-commerce
❑ Consulting & Systems Integrator
❑ Financial Services
❑ Manufacturing
❑ Wholesale/Retail/Distribution
❑ Transportation
❑ Travel/Hospitality
❑ Government/Military/Aerospace
❑ Health Care/Medical
❑ Insurance/Legal
❑ Education
❑ Utilities
❑ Architecture/Construction/Real Estate
❑ Agriculture
❑ Nonprofit/Religious
❑ Other (please specify)
____

C. Total number of employees at your location and entire organization (check all that apply):

	Location	Company
10,000 or more	01 ❑	01 ❑
5,000 – 9,999	02 ❑	02 ❑
1,000 – 4,999	03 ❑	03 ❑
500 – 999	04 ❑	04 ❑
100 – 499	05 ❑	05 ❑
100 or less	06 ❑	06 ❑

D. Please indicate the value of communications and computer products and services that you recommend, buy, specify, or approve over the course of one year:
❑ $10 million or more
❑ $1 million – $9.9 million
❑ $500,000 – $999,999
❑ $100,000 – $499,999
❑ $10,000 – $99,999
❑ Less than $10,000
❑ Don't know

E. What is your company's gross annual revenue?
❑ $10 billion or more
❑ $1 billion – $9.9 billion
❑ $100 million – $999 million
❑ $10 million – $99.9 million
❑ $1 million – $9.9 million
❑ Less than $1 million
❑ Don't know

F. Do you recommend, specify, evaluate, approve or purchase wireless products or services for your organization?
01 ❑ Yes 02 ❑ No

G. Which of the following products, services, and/or technologies do you currently approve, specify or recommend the purchase of?
❑ Application Servers
❑ Web Servers
❑ Server-Side Hardware
❑ Client-Side Hardware
❑ Wireless Device Hardware
❑ Databases
❑ Java IDEs
❑ Class Libraries
❑ Software Testing Tools
❑ Web Testing Tools
❑ Modeling Tools
❑ Team Development Tools
❑ Installation Tools
❑ Frameworks
❑ Database Access Tools/JDBC Devices
❑ Application Integration Tools
❑ Enterprise Development Tool Suites
❑ Messaging Tools
❑ Reporting Tools
❑ Debugging Tools
❑ Virtual Machines
❑ Wireless Development Tools
❑ XML Tools
❑ Web Services Development Toolkits
❑ Professional Training Services
❑ Other [Please Specify]

SYS-CON Events, Inc., and SYS-CON Media make no warranties regarding content, speakers, or attendance. The opinions of speakers, exhibitors, and sponsors do not reflect the opinion of SYS-CON Events and SYS-CON Media, and no endorsement of speakers, exhibitors, companies, products, or sponsors is implied.

SYS-CON EVENTS

If you require special assistance covered under the Americans with Disabilities Act, please call 201-802-3058 by February 10, 2004.

CANCELLATIONS, SUBSTITUTIONS, REFUNDS
Fax written request to SYS-CON Registration 201-782-9651. Requests for refunds received prior to January 16, 2004 will be honored less a 10% handling charge; requests received after January 16, 2004 and before February 6, 2004, will be honored less a 20% handling charge. No requests for refunds will be honored after February 6, 2004. Requests for substitutions must be made in writing prior to February 20, 2004. No one under 18 is permitted to attend. No warranties are made regarding the content of sessions or materials.

Speakers, sessions, and schedule are subject to change without prior notice.

No solicitation by anyone other than official exhibitors, sponsors or marketing partners is permitted. Such behavior is cause for expulsion without refund.

SPECIAL INSERT:
Edge 2004 East Conference & Expo February 24 – 26, 2004

SPECIAL INSERT: Edge 2004 East Conference & Expo February 24 – 26, 2004

VISIT www.sys-con.com/edge

SPONSORS & FEATURED EXHIBITORS

Computer Associates
Gold Sponsor

Computer Associates International, Inc. (CA) offers the most comprehensive portfolio of solutions for managing Web services and Web services platforms. CA's end-to-end solutions, including Unicenter® Web Services Distributed Management, dramatically reduce the cost and complexity of Web services deployments by providing visibility and control, optimized performance and increased enterprise responsiveness. For more information, please visit ca.com/webservices.

EAI Industry Consortium
Association Sponsor

The EAI Industry Consortium, a non-profit global advocacy group, developed to promote Enterprise Application Integration through sponsored research, the establishment of standards and guidelines, best practices and articulation of strategic and measurable benefits. The member driven consortium, designed as an EAI information hub encompasses marketplace education, resource tools and EAI trends, providing members a venue to develop, create and debate.

Ektron

Ektron is a leader in developer-friendly Web content authoring, publishing and management for non-technical end users. Ektron's easy-to-deploy, affordable software enables users to easily update, publish and manage Web content from anywhere without knowing HTML or XML, while working in an environment easily controlled by Web professionals. Nearly one million individuals worldwide use Ektron's award-winning products.

IONA

Artix is a family of Web services based service-oriented integration products that renovate IT assets and consolidate legacy middleware to reduce the complexity and cost of IT operations. Customers use Artix to maximize the value of existing systems through the deployment of best of standards-based middleware and enterprise Web services.

Kenetiks
Education Sponsor

Kenetiks is a worldwide provider of Java-WebSphere training and project mentoring with courses available for every skill level. Our Kenetiks Academy can show you how to build your own J2SE/J2EE applications for IBM WebSphere Application Server and IBM WebSphere Portal Server using WebSphere Studio Application Developer.

Macromedia
Platinum Sponsor

Macromedia is motivated by the belief that great experiences build great business. Macromedia provides the industry-leading software that empowers developers and designers to create and deliver those effective, compelling, and memorable experiences.

Microsoft
Gold Sponsor

Microsoft .NET is a set of Microsoft software technologies for connecting information, people, systems, and devices. It enables a high level of software integration through the use of Web services-small, discrete, building-block applications that connect to each other as well as to other, larger applications over the Internet.

Novell
Silver Sponsor

Novell helps customers dynamically access, provision, manage and deliver secure information services. Customers can flexibly integrate applications and data and ensure unified identity across systems. Novell's Web Services platform, exteNd, is a comprehensive, integrated approach to the visual development and deployment of services-oriented applications.

OASIS
Association Sponsor

OASIS is the non-profit, international consortium that has been providing open solutions for electronic data interchange since 1993. Dedicated from its inception to the technology now known as XML, OASIS is the world's largest independent, vendor-neutral organization for the standardization of XML applications in electronic commerce. The primary mission of OASIS and its members is to identify and resolve interoperability issues that exist between XML applications and technologies.

Sonic Software

Sonic Software is changing the economics of integration. Sonic provides the first comprehensive business integration suite built on an enterprise service bus (ESB). The Sonic product line delivers a distributed, standards-based, cost-effective, easily managed infrastructure that reliably integrates applications and orchestrates business processes across the extended enterprise

Velocitis

Velocitis provides the next level in agile software development tools with Flywheel, a code centric tool for rapidly designing, visualizing and refactoring Visual Studio .NET code. Fine grained synchronization allows the Flywheel and Visual Studio tools to be used in any order, at any time in the development process. With Flywheel developing in code or visually is the same because, the code is the model.

Westbridge Technology
Education Sponsor

Westbridge Technology provides Web Services Management solutions that enable enterprises to secure and manage XML Web Services networks. The Westbridge XML Message Server (XMS), enables organizations like the FTC, Danish Government and Travelers Property Casualty to seamlessly connect, secure and manage XML Web Services networks. For more information, visit www.westbridgetech.com.

WEB SERVICES EDGE CONFERENCE PAST SPONSORS & EXHIBITORS

Actional · AltoWeb · Antenna House · Covasoft · CTi · Dirig Software · EvolveWare · eXcelon · FatWire Software · webGain · attachmate · Cacheon · Connotate · Apress

Microsoft · Corda · atg · Forum Systems · Full Tilt · Gamexpress · Sun microsystems · webputty · Iopsis Software · Rational · TIBCO

Seagull · aligo · OneWebSystems · bea · Bestning XML Security · Cape Clear · CodeHound.com · Compuware · Data Junction · DataMirror · Entegrity Solutions · Vincera

HiT Software · ParaSoft · InCert · Inferdata · infoShark · ReportMill Software · LearningPatterns.com · Merant · Novalabs · pingtel · PyBiz · Rogue Wave Software

spiritsoft · Data Qua · Oracle · Borland · xml spy · eltegra · Matisse · IPEDO · PolarLake · Vordel · Web Age · Altaworks · Automatic Design

bowstreet · CSC · hp invent · iam · Information Laboratory · pingtel · iWay Software · McCabe & Associates · Nextel · Object Design · Pramati · Infravio · Kildara

XML Global · Mercury Interactive · isavvix · Jinfonet Software · KeyLabs · macromedia · Melissa Data · Metaserver · Micro Focus · Vultus · webMethods · Mongoose Technology · oakgrove

OpenDemand Systems · Our World Live · PointBase · Sybase · Trivera · Wakesoft · ObjectVenture · Optimyz · plumtree · serena · teamshare · Trellix

LOOK WHAT'S COMING NEXT MONTH

BEA WebLogic Enterprise Security

Enterprises have traditionally tried to resolve the security dilemma by hard-coding security into individual applications. BEA has taken a distributed architecture and service-oriented approach that is designed to allow for centralized security policy visibility and control without sacrificing enterprise performance, scalability, and reliability. Learn about BEA's new application security infrastrucuture solution from the people who helped to develop it.

EMALL: The Security Architecture

EMALL is a DOD procurement and collaborative commerce portal based on J2EE, WebLogic Integration, Web services, and the ebXML 2.0 protocol. The systems used here interact with each other using ebXML, under stringent security requirements. We will identify the major types of security threats and vulnerabilities and explain how we used specific parts of WebLogic security to reduce these threats.

ebXML and XML Digital Signature in WebLogic Integration 8.1

The data exchanged in business-to-business transactions is often sensitive and requires protection. Secure Socket Layer (SSL) provides protection at the transport level through confidentiality of data exchanged between two end points. This article offers an introduction to XML Digital Signature and describes how it can be used with ebXML in WebLogic Integration 8.1.

Analyzing Java Thread Dumps

Java thread dumps taken from a running Java application provide useful information about the state of the application at any point in time. This article explains when and how to take Java thread dumps, how to analyze them, and how to arrive at conclusions. It also explains the most common states of Java threads and their significance in analyzing the problem.

Handling System Core Files

A system core file is usually indicative of an error in some native code. This could be from the application code of a user (if you are using native code [JNI] in your application), an error in the Java Virtual Machine version you are using, or in BEA WebLogic Server itself. This article offers some useful tips for debugging your BEA WebLogic Server applications when a system core file occurs.

BEA WebLogic DEVELOPER'S JOURNAL

WebLogic Clustering on Intel-Based Blade Servers

A robust, reliable platform for complex implementations

BY **NITIN SINGHVI & ARAVIND PAVULURI**

AUTHOR BIO

Nitin Singhvi is a senior performance analyst at Intel Corporation where he has worked for the past 8 years. His focus has largely been in the area of enterprise server systems analysis using various server workloads. His work has spanned designs starting from some of the early Intel server components to current high-performance products.

Aravind Pavuluri is a software performance engineer with the Managed Runtime Environments group at Intel. He has been involved with optimizing Java Runtimes as well trying to understand their impact on the Intel platforms. He holds a master's degree in information networking from Carnegie Mellon and a bachelor's in computer science from West Virginia Tech.

CONTACT...

nitin.singhvi@intel.com
aravind.pavuluri@intel.com

This article presents the clustered implementation and performance scaling of SPECjAppServer2002 on dual processor Enterprise Blade Servers based on Intel architecture using BEA WebLogic Server clustering support in a Linux environment (Red Hat AS 2.1).

A brief introduction to Blade architecture is provided, along with an overview of WebLogic Server clustering capabilities, including the steps involved in setting up a clustered implementation of the workload. This is followed by the characterization of the performance and scaling (scale out) of the workload as the cluster size is increased from single to multiple nodes.

SPECjAppServer2002

The SPECjAppServer2002 benchmark measures the performance of Java Enterprise application servers based on J2EE, gauging the scalability of J2EE servers and containers. SPECjAppServer2002 stresses the ability of Enterprise JavaBean (EJB) containers to handle the complexities of memory management, connection pooling, passivation/activation, caching,and so on. It models manufacturing, supply chain management, and order/inventory business environments. There are four defined business domains: manufacturing, supplier, customer, and corporate. Performance results for the benchmark are influenced primarily by hardware configuration, J2EE application software, and database software. The setup consists of the driver and supplier emulator, the application server, and the database server. The run rules require that the driver and the supplier emulator must be run outside the system under test. Hence a network and at least two systems are required. The benchmark can be run in either a centralized or a distributed mode (see www.spec.org/jappserver2002 for description).

Results reported using this workload fall into one of three categories:

- ***Single node systems:*** A single node with a coherent memory system runs both the application server and database instances
- ***Dual node systems:*** Two nodes with coherent memory systems, one running the application server and the other running the database
- ***Multiple node systems:*** Contain three or more nodes (systems running a distributed operating system cluster or non coherent memory system fall into this category)

The primary performance metric is TOPS (Total Operations Per Second). This is the number of customer order transactions plus the number of manufacturing work orders divided by the measurement period in seconds. Among the constraints to meet for compliant runs are response-time constraints that affect all categories of transactions and limit the maximum rate at which transactions can be injected into the system by the driver, hence impacting throughput. Getting good results for this workload requires tuning the different run parameters to get increased throughput via maximum utilization of the application server (indicated by high CPU utilization numbers) while satisfying the run rules and constraints.

Our SPECjAppServer2002 Configuration

Our study focuses on the centralized mode of the benchmark and our category is a multiple node system consisting of a driver, a database server, and a cluster of application servers. However, the purpose of our study is not to report top performance in a category but rather to implement and study the effects of clustering. Furthermore we include Failover capabilities (explained later), which provide a realistic clustering scenario and do reduce the performance but are not a prerequisite for performance publications. Hence the performance reported is not optimal from a publication perspective.

Blade Architecture

Blade servers (see Figure 1) form the hardware foundation for a modular computing paradigm. While the pedestal and rack-mounted system approach increases complexity, modular computing reduces complexity. Modular computing enables greater automation and virtualization of resources, creates reliable systems essentially with no single point of failure, and helps reduce IT costs while increasing capacity, functionality, and agility. An emerging technological approach to datacenter architectures, modular computing integrates chassis-based modular hardware resources that are easily removed and replaced. Chassis-based systems aggregate interconnect, cabling, switching, power supplies, cooling, and other resources, simplifying the infrastructure and its management and service. Modular computing allows administrators to customize computing resources with a variety of processing resources – one-way, two-way; 32-bit, 64-bit, etc. – on server boards that are easily installed into a chassis, much like chassis-based switches currently used in large data centers. Modular switching and management components might also be part of the chassis, both to integrate functionality and to abstract monitoring and control of the entire system. Chassis components communicate across a modular backplane (or midplane, depending on their location in the chassis) and can be designed to provide full redundancy for high availability and protection against a single point of failure, as well as to allow easy serviceability.

As modular computing matures, future virtualization capabilities will treat resources as pools, regardless of their location, that can be quickly reconfigured to meet changing business needs. Management components will automate much of the work done manually by administrators today, giving them more time to focus on other critical tasks and initiatives. Blade servers and the modular computing paradigm give datacenters flexible resources, while simplifying the architecture to positively impact business agility and operating costs.

Our Blade Server Cluster Configuration

The blade server cluster we used allows for scaling up to 14 blade-server bays. It includes an integrated and hot-swappable Gigabit Ethernet switch module (scalable up to two modules). We did not use an external disk system for the cluster, but up to two integrated and hot-swappable Fibre Channel Switch Modules are possible. Each blade has two Intel Xeon processors running at 2.8 GHz with a 512K Cache. The front-side bus in each blade runs at 533MHz, and each blade is configured with 2.5GB of DDR SDRAM. This can be increased to a maximum of 8GB.

Clustering Using WebLogic Servers

A BEA WebLogic Server cluster consists of multiple instances of the WebLogic Server cooperating with each other in order to provide increased application scalability and reliability. These multiple instances of the WebLogic Server constituting a cluster may run on the same physical system or multiple physical systems, but for all the client requests they appear as a single, coherent unit.

As mentioned above, the biggest advantages of a cluster include scalability and availability. Provisioning additional capacity for an enterprise application just requires enabling more instances of WebLogic Server in the cluster.

In a cluster with failover enabled, the application components are deployed on multiple instances of the cluster. In case of either a node or a WebLogic Server instance failure, the other servers in the cluster pick up the work where the first server left off, thus maintaining availability and reliability.

Some of the objects that can be clustered using WebLogic Server include servlets, JSPs, and EJBs.

Architecture of WebLogic Server Clusters

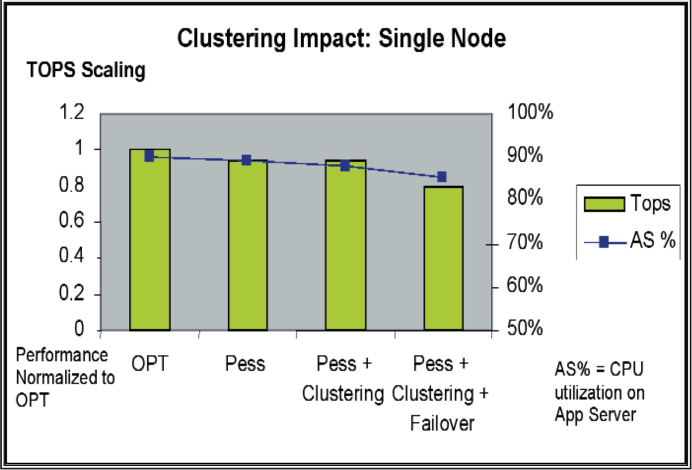
A cluster should be part of a domain. A domain can consist of multiple instances of WebLogic Server, some of which may be clustered while others are not. A domain logically groups these multiple instances of WebLogic Server based on some criteria.

Every domain needs to have a single administrative server whose responsibilities include configuring and managing the other instances of WebLogic Server in the domain. The admin server is configured with the information regarding the other instances in the cluster (their names/IP addresses/listen ports, queue sizes and other options, the cluster multicast address/port they should use, etc.) as well as with the application files intended for deployment in the cluster.

FIGURE 1

Blade server

FIGURE 2



Clustering performance for a single node

A cluster, in addition to the admin server, also hosts other instances of WebLogic Server that are the real work horses. These are known as “managed servers” and are started with information regarding how to contact the admin server. The managed servers then establish communication with the admin server and download their configuration information as well as the applications to be deployed. After retrieving this data from the admin server, they synchronize with the other managed servers in the cluster.

Clustering EJBs

SPECjAppServer2002 is the workload used in our study because this application makes extensive use of EJBs for its business logic, stressing the EJB infrastructure of an application server. Consequently, discussion of the clustering of EJBs is appropriate.

Failover and load balancing in a cluster are achieved through “replica-aware stubs” (RAS). When a client initially contacts one of the managed servers, it gets back a replica-aware stub. This stub contains information regarding all the server instances on which the particular object that the client has requested is hosted. The client, using the load balancing algorithm information stored in the stub as well the information of all the hosts of that object, then routes its request to the appropriate host.

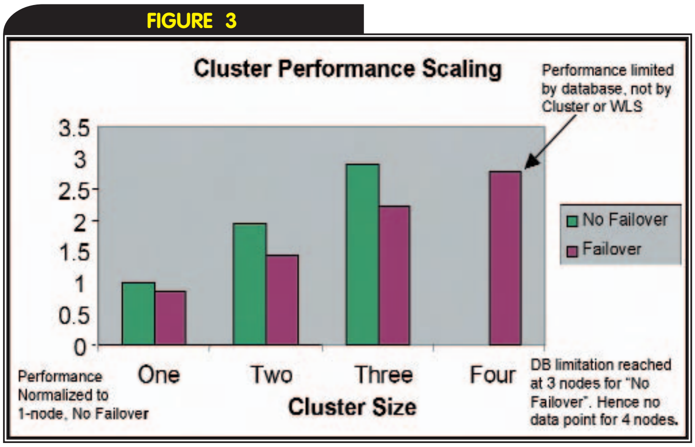
The three main types of EJBs, all of which are implemented and used by SPECjAppSever2002, include stateless, stateful, and entity EJBs.

In the case of stateless EJBs, the RAS contains information regarding all the WebLogic Server instances hosting the bean. The client can then choose from these instances the one to which it shall route its call. Failover in this scenario is just moving to the next instance of WLS hosting that bean.

Stateful EJBs need to maintain the state of the transaction. The WebLogic Server instance on which it is initially created, based on a client’s request, is referred to as the primary server (PS). The PS selects one other server in the cluster on which to replicate the state of the bean, and this chosen server is referred to as the secondary server (SS). The RAS that is sent back to the client contains information regarding both the PS and the SS, and in case of the failure of the PS, the client can switch to the SS.

An entity EJB’s state is stored in persistent storage, like in a database. The data store that holds the entity EJB’s state is shared among all the servers in the cluster. In the case of an entity EJB the RAS contains information about all the servers hosting the entity bean. If one of the servers were to fail, the request could be failed over to any of the remaining servers (as indicated by RAS), which then picks up the last committed state of the bean from the data store.

FIGURE 3



Cluster performance sealing

Setting up SPECjAppServer2002 for Clustering

We’re now ready to discuss the actual implementation of SPECjAppServer2002 using clustering.

1. Creating the Clusterable EAR File

The two mainWebLogic Server concurrency choices for a user are:

- ***Optimistic concurrency:*** The application server optimistically processes transactions without locking the records in the database and rolling them back if necessary.
- ***Pessimistic concurrency:*** Every transaction involves locking the records in the database.

SPECjAppServer2002 in a clustered WebLogic Server environment shows better performance with the pessimistic concurrency model than with the optimistic concurrency model. Thus we enabled clustering and the pessimistic concurrency model in the creation of the SPECjAppServer.ear file.

Failover for stateful session beans (SFSB) in WebLogic Server is enabled through “in memory replication” (IMR) since these objects are replicated on the primary and secondary servers in their memories. SPECjAppServer2002 has two SFSBs, “CartSes” and “BuyerSes”. Enabling IMR for these beans was done by adding the changes shown in Listing 1 in their deployment descriptors.

Once compiled and created after these changes, you have the specjappserver.ear file that can be deployed in a clustered failover environment.

2. Creating the Cluster

BEA WebLogic Server provides an intuitive GUI-based wizard called “BEA WebLogic Configuration Wizard” (activated by running the script /bea/weblogic81/common/bin/config.sh) that guides you through the process of configuring a cluster, the administrative and managed server instances in the cluster, physical systems in the cluster, associating the instances to the cluster as well as to the physical systems, and finally the domain of the cluster.

In the location of the created domain directory, a user finds the scripts startAdminServer.sh and startManagedServer.sh, which are used to run the WebLogic Server instances. In these scripts, make sure that the option --DWeblogic.ProductionModeEnabled =true.

Once the domain is configured, the EAR file created in the previous section is transferred to an applications sub-directory, and in the config.xml, the user points the application to be deployed to this EAR file.

To bring up the cluster, the administrative server is started. Once it is up and running, the other managed servers in the cluster are started up and then the emulator (the supplier emulator hosting servlets) is launched along with the driver (which generates requests to the application servers) instances on a different node.

Cluster Performance: Clustering Overhead on a Single Node

Once the cluster is configured and functional, the workload can be run and cluster performance measured. As discussed earlier, enabling clustering involved three main configuration changes in WebLogic Server: (1) changing to a clustering EAR file; (2) moving from an optimistic to a pessimistic concurrency model for database access; and (3) enabling failover. The performance impact of these changes in a single-node application-server configuration seems to be within reasonable limits. Note that CPU utilization on the application server remains about the same when moving from optimistic to pessimistic concurrency even though there is a slight drop in performance. Enabling “clustering” in WebLogic Server does not seem to have much impact on either application server CPU utilization or throughput (TOPS), indicating low overhead. The remaining drop in performance comes from enabling failover (see Figure 2).

Cluster Performance: Multiple Nodes

The next phase is to scale cluster size and measure the scaling in performance as shown in Figure 2. Cluster configurations both with and without enabling failover are shown. Scaling without failover enabled is almost linear from 1–3 nodes. We see a constant performance increase of about 94% of single-node performance with each node added to the cluster (i.e., 2 nodes = 1.94 x 1 node and 3 nodes = 2.88 x 1 node). We were unable to further scale the cluster in the "No Failover" configuration since the database becomes the performance limiter at this point due to high CPU utilization. Cluster performance scaling with failover enabled is somewhat lower at about 80% of single node performance for each node added, going from 1–3 nodes. Apart from additional failover-related processing necessary at each node, failover communication must now go across nodes in the cluster, adding to the overhead. With failover enabled, it was possible to add a fourth node since there was headroom available in database CPU capacity, but we must note that performance at four nodes does not truly represent workload scaling since at that point it is extraneously constrained by database CPU utilization having reached limiting levels. CPU utilization in each application server in the cluster remains fairly constant in the 80–81% range going over 1–3 nodes. For a four-node cluster, database limitations prevent CPU utilization in the application servers from going beyond ~ 61%. In all the configurations discussed, WebLogic Server achieved a very balanced load across the cluster and total network traffic was low enough for the Gb cluster backplane so as not to be a performance issue.

Summary

The clustered implementation of SPECjAppServer2002 on Intel architecture-based Enterprise Blade Servers using WebLogic Server clustering support was discussed. The Blade Server cluster was found to be a robust, convenient, and reliable platform that helped ease the effort involved in a complex implementation. WebLogic Server clustering support provided ease of workload configuration and good performance scaling over the 1–3 node range. The performance impact of enabling failover seems to be within reasonable limits. Performance scaling with clustering seems very encouraging and highlights the potential usefulness and practicality of a clustered approach.

Listing 1

```
<Weblogic-enterprise-bean>
         <ejb-name>CartSes</ejb-name>
                    .
                    .
                   <stateful-session-clustering>
                             <home-is-clusterable>
                                       <!--CLUSTERABLE-BOOLEAN-->
                             </home-is-clusterable>
                             <replication-type>
                              InMemory
                             </replication-type>
                   </stateful-session-clustering>
                    .
                    .

  </Weblogic-enterprise-bean>
```

UNDERSTANDING THE WEB SERVICES VISION
EVALUATING and ADOPTING WEB SERVICES
A pragmatist's view
Scalable Web Services
WebServices JOURNAL
.NET or J2EE

Mercury Interactive's LoadRunner

THE J2EE TRANSACTION BREAKDOWN DIAGNOSTICS MODULE, A PART OF MERCURY'S OPTIMIZATION SUITE FOR J2EE

Reviewed by
JASON SNYDER

Corporate Information
Mercury Interactive
Building A
1325 Borregas Avenue
Sunnyvale, CA 94089
(408) 822-5200

Customer Support:
Support@merc-int.com
+1-877-TEST HLP (toll free)
+1-877-837-8457 (toll free)

Requirements:
Pentium 350 Mhz or higher; Windows NT service pack 6a, 2000, or XP; Internet Explorer 5.x or higher or Netscape Navigator 4.x, 6.x

Mercury Interactive's LoadRunner is a leader in the performance-testing market. Its ability to create large volumes of data is legendary, and its ability to monitor the systems being tested provides great value. The J2EE Transaction Breakdown Diagnostics Module, LoadRunner, brings detailed J2EE transaction analysis into the mix.

Figure 1 illustrates at a high level the degree of monitoring and analysis available to the performance tester. The J2EE Transaction Breakdown allows each tier to be viewed, with detail to the method level for the application and Web servers. In addition, specific database queries can also be viewed.

The module provides for true integration of performance testing and performance monitoring. This allows greater diagnosis capabilities, improved control over the J2EE testing scenarios, and superior ability to pinpoint specific performance issues. For this review, I tested the WebLogic J2EE sample, Avitek Medical Records, using my machine as both the application server and the testing environment. My script exercised the three applications (patient, doctor, and admin), each of which have slightly different application architecture designs so I could expect to see some differences on similar calls.

The first step was to update the startup file for the WebLogic domain. I had to make the following updates to the startup file (startMedRecServer):

```
set
MERC_TXBD_HOME=c:\Progra~1\Mercur~1\
J2EE_TXBD_Probe
set CLASSPATH=%WEBLOGIC_CLASS-
PATH%;%CLASSPATH%;%MERC_TXBD_HOME%\d
at;

set PATH=%PATH%;MERC_TXBD_HOME%\bin

"%JAVA_HOME%\bin\java" %JAVA_VM%
%MEM_ARGS% %JAVA_OPTIONS%
-
Xbootclasspath/p:%MERC_TXBD_HOME%\cl
asses\boot
```

When the server starts up, look for "Initializing Transaction Break Down Probe" to confirm that the changes had an impact and the Mercury Probe was initiated. After making these changes, I went into the Virtual User Generator and generated a script of the steps to take during processing. These scripts are necessary during performance testing, and the interface was simple to use. I called the URL Web interface for the Avitek Medical Records sample and began recording. Transaction segments are user defined, so you can create transactions to precisely meet your needs.

Once the script was created, I created a Controller Scenario, which utilizes the script to simulate the number of users. The license associated with the copy of the product I had permitted only one user. The next step was to start the scenarios and then monitor the results.

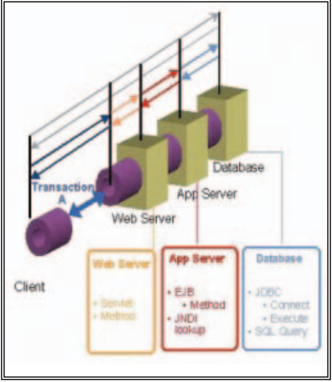
The LoadRunner Controller provides real-time monitoring for users to analyze J2EE component metrics. The metrics are gathered by an agent installed on the application server to collect information on the J2EE components. These measurements are sent from the application server back to the LoadRunner Controller.

The LoadRunner Controller application consists of three tabs: Design, Run, and J2EE Diagnostics. The Design tab allows various scenarios to be grouped and scheduled.

The Run tab (Figure 2) allows a variety of statistics and measurements from the running scripts to be captured and analyzed as graphs and tables. Many of these real-time measurements can be captured and stored as HTML, so any moments that could illuminate can be reviewed later. The viewer determines the number and type of measurements to display at any one time. Of particular interest to J2EE testers and developers are the WebLogic measurements and the Java Performance graphs. Although the setup for the WebLogic measurements required updates to some configuration files (I am using WLS 8.1 and JDK 1.4, not currently expected defaults of the product), the effort proved worthwhile.

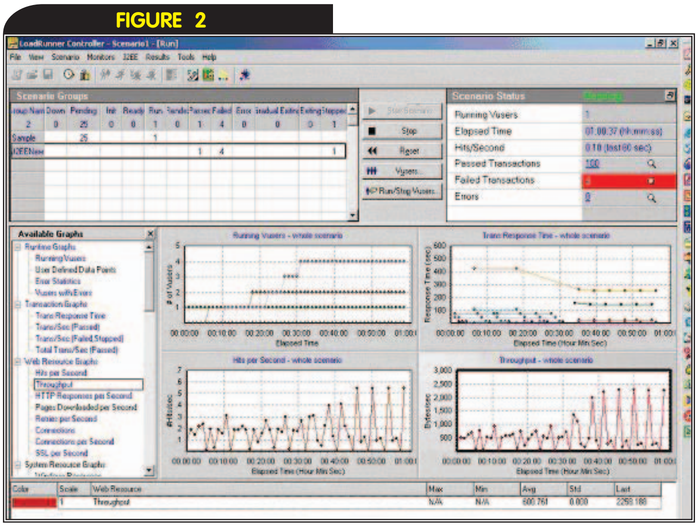
The WebLogic measurements use JMX to access application server–specific detail. The measurement alternatives are vast, and include the ability to review measurements about logging, the JVM itself, servlets, EJB components, JMS connections, and the JDBC connection pool. The Java Performance graphs provide method-level detail on average response times and method call per seconds. Evaluating this information for Avitek showed me that my limited load was hardly phasing the application and that my longest-running, server-side transactions were logging in as an admin, denying Avitek patient requests for enrollment. Each had an average response time of about 0.3 seconds and involved the admin application, so I was curious if there was a common cause for their slower (but not slow) response times.

FIGURE 1



High-level view of available monitoring and analysis

FIGURE 2



Group Name	Down	Pending	Init	Ready	Run	Rendez	Passed	Failed	Error	Gradual Exiting	Exiting	Stopped
2	0	25	0	0	1	0	1	4	0	0	0	1
Sample		25			1							
J2EENew							1	4				1

Running Vusers	1
Elapsed Time	01:00:37 (hh:mm:ss)
Hits/Second	0.10 (last 60 sec)
Passed Transactions	100
Failed Transactions	4
Errors	0

Color	Scale	Web Resource	Max	Min	Avg	Std	Last
🟥	1	Throughput	N/A	N/A	600.761	0.000	2258.188

Transaction Chain of Calls

The final tab shows detailed J2EE Diagnostics. The J2EE monitor provides the following information for each J2EE component:

- Average response time per method/query
- Number of method calls per second

With such coverage of the J2EE architecture, users can get an overview of the entire activity within the system. They can very easily correlate the end user response time with the Web server activity (servlets and JSP data), application server activity (JNDI and EJBs), and back-end activity of database requests (JDBC methods and SQL queries). All of these measurements can be drilled down to the specific method level.

I was curious about the Deny Request transaction, so I brought up a Transaction Chain of Calls (see Figure 2). This report shows the measurement diagnostics for a transaction by method, on one screen, providing an immediate breakdown as to the longer parts of the transactional call. Looking at this report, it became apparent that the JDBC SQL execution for getting the patient took about 32% of the total transaction time. The actual update to the patient itself took only 4% of the time. A similar transaction from the Patient application took much less time. The reason: the method by which the Patient EJB was generated. The Admin transaction used the findByUserEJB method while the Patient looked up by primary key. The difference was the time it took for the query to retrieve, as the following data illustrates:

```
Patient - Update Profile
Transaction

EjbFindByPrimaryKey Query:

SELECT WL0.id , WL0.address_id ,
WL0.dob , WL0.email ,
WL0.first_name , WL0.gender ,
WL0.last_name , WL0.middle_name ,
WL0.phone , WL0.ssn FROM patient
WL0 WHERE ( WL0.id = ? )

Average Response Time: 0.01 seconds

Admin - Deny Request Transaction

EjbFindByUserEJB Query:

SELECT WL0.id , WL0.address_id ,
WL0.dob , WL0.email ,
WL0.first_name , WL0.gender ,
WL0.last_name , WL0.middle_name ,
WL0.phone , WL0.ssn FROM patient
WL0 WHERE ( WL0.email = ? )

Average Response Time: 0.06 seconds
```

This is obviously just the tip of the iceberg for the type of the analysis that can be performed.

The final part of the toolset that I evaluated was the LoadRunner Analysis tool. This product uses the results of the scripts from the Controller Run efforts for additional review and analysis. The J2EE analysis reports provide transactional response time for the server, time spent in each server element, method calls per second, and average method response time, among others. Multiple scenarios and transactions can be statistically analyzed to determine if any correlation of behavior exists. In addition, all of these reports can be drilled down or broken into their component Transaction Chain of Calls, which can then be printed or stored as HTML.

Conclusion

Mercury Interactive has created a unique product that will greatly accelerate J2EE performance-testing efforts. By integrating the performance-testing tool with advanced J2EE diagnostics measurements and capabilities, performance problems can be simply diagnosed before production. This was an extremely easy tool to set up, and I was able to monitor any application of my choosing. It should be noted that Mercury Interactive also provides products for J2EE production monitoring and deep diagnostics, but these tools were not reviewed. Many thanks to Richard O'Connell of Mercury Interactive, who helped get me set up and running.

AUTHOR BIO

Jason Snyder is the product review editor for *WLDJ* and an architectural expert for CSC Consulting in Boston, and has served as the lead architect for several J2EE development projects. He has over 10 years of experience in software development, OO design, and application architecture.

CONTACT...

jsnyder1@csc.com

Transactions: How Distributed Are Yours?

TALK TO YOUR PEERS

BY **PETER HOLDITCH**

AUTHOR BIO

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

CONTACT...

peter.holditch@bea.com

REPRODUCED WITH PERMISSION FROM BEA SYSTEMS.

Another discussion based on a weblogic.developer.interest.transaction posting this month. It's a newsgroup that always proves to be a good source of information for the world at large when it comes to transactional behavior (and a good source of inspiration for me when the article time of the month rolls around again).

This particular posting is a great illustration of how an apparently simple assumption can pitch you into the bowels of the infrastructure, if you're not carefully guided by the architecture you're building against, and the possibilities it offers. The original posting ran:

We have to use distributed transactions between an XADataSource configured at the Web tier level in Tomcat 4.1.27 and an XADataSource configured in WebLogic Server 8.1. As a proof of concept, I built a small servlet sample that uses the XADataSource in Tomcat and uses the transaction manager of WebLogic.

When enlisting the XAResource with the javax.transaction.Transaction object, I keep receiving the following exception:

```
javax.transaction.SystemException: Not implemented at
weblogic.corba.j2ee.transaction.TransactionManagerImpl.
enlistResource (TransactionManagerImpl.java:370)
   at
com.myco.jta.TestServlet.addSubscription(TestServlet.ja
va:103)
   at
com.myco.jta.TestServlet.doGet(TestServlet.java:46)
   at
javax.servlet.http.HttpServlet.service(HttpServlet.java
:740)
   at
javax.servlet.http.HttpServlet.service(HttpServlet.java
:853)
```

We use XAPool from ObjectWeb as the XADataSource implementation. I'd like to know if we are using JTA properly and if we are missing a configuration of some sort at the WebLogic level.

On the face of it, what could be simpler: a distributed architecture, and we want distributed transactions so let's start coding… But these exceptions quickly put a stop to that, and give pause for thought… Let's think this through.

A first observation, at a high level (and at a tangent to the whole point of this article), is the implicit physical splitting of tiers in this posting. Okay, there may be very good reasons in this case why Tomcat is used for the JSP/servlet logic – maybe it's already in production and is being extended – but WebLogic has a perfectly good servlet/JSP capability. It is amazing how many people decide that the fact that Sun describes a Web tier and an EJB tier means that these tiers must be physically separate. Even people using WebLogic for both the Web tier and the EJB tier often decide that they should deploy different WebLogic instances for both. This is not necessarily the case. Calling between processes is costly and a source of complexity in development, configuration management, and production maintenance. Yes, you *can* make the tiers physically separate, but even if all the tiers are deployed in a single WebLogic instance they are no less *logically* distinct! Sorry about that rant, this is a subject that really gets my goat. Just be thankful that I fought the temptation to type the whole of this paragraph in bold caps!

That Really Gets My Goat!

So, Ommmmm… Calm, calm, calm… and back to the plot.

At the high level, transactions flow around our J2EE system quite transparently; you start them, make calls, commit them, and take it for granted that the transaction context flowed around with your calls, and the commit processing touched all the resources you ended up using. This sounds like magic, and – of course – like everything else in the world, it isn't. There is a rational explanation. To find it, you just need to look beneath the J2EE surface you're coding to. What happened under the covers is that every time you made a call, the runtime discovered a Transaction object associated with your thread. This discovery prompted the runtime to piggyback the data about the transaction with the request. The piggybacked transaction information has, in turn, caused the infrastructure on the receiving end to do the "right thing" in terms of associating a transaction with its thread of control, and piggybacked data about the resources that got touched to make sure all the bits of transaction know about each other, so when the time comes the transaction coordinator can wander by, tell all the resources (which it has collected a list of) to commit, and life is good. All this apparent magic is wonderful, but it does imply a pretty close coupling between the application server's runtime infrastructure and the transaction manager, and this implication is not for nothing – the two are pretty inextricably tied together. So when you start talking about the scenario laid out in the newsgroup, where two different infrastructure implementations are involved, things are going to get a bit hairy if you carry on trying to make the simple assumption that transactions will flow between the containers and "just work" when it comes to commit time.

What it boils down to is that BEA WebLogic cannot make any assumptions about the nature of a client that is calling it but, as I discussed way back in November 2002 (***WLDJ,*** Vol. 1, issue 11), it tries to smooth over the cracks by allowing a client to demarcate transactions whose completion will be delegated to a server-side transaction manager. A Tomcat client is no exception to that; as far as WebLogic is concerned, it is every bit as incapable as an applet.

What's Needed Is a Peer Relationship

What's needed to flow transactions between implementations in the way that the original poster wanted is a peer relationship between them, not a client/server one. If the two containers can be confident that each can look after itself in terms of transaction logging and recovery, then all that is needed is a shared understanding of how the transactions in one environment map on to those in the other. One way of doing this is via RMI/IIOP. If the containers are both speaking that protocol (as they are mandated to be able to do by J2EE 1.3) then the CORBA OTS specification provides a standard mechanism and interfaces for the transactions to flow through. Each container can handle its own internal implementation, with OTS providing a gateway between them. Internally to WebLogic, this gateway is implemented via a concept called an "Interposed Transaction Manager" – if WebLogic needs to import a transaction via OTS, that actually entails listening for "prepare", "commit", and "rollback" instructions from the remote system and responding appropriately to them. The BEA WebLogic transaction manager simply sits between these instructions and the actual resources participating in the transaction on the WebLogic side – it is interposed between the TM that controls the transaction and the local resources that it manages. If you think about it, responding to begin/prepare/commit/abort commands is exactly what any resource manager does through the XA interface. Wouldn't it be a neat trick if an XA interface could be exposed to WebLogic's transaction manager; then any foreign transaction manager could propagate transactions into WebLogic via XA, as if it were a database or any other xa resource, irrespective of support for any given network protocol.

The good news is that WebLogic's transaction manager offers just such a capability via the getServerInterposedTransactionManager and getClientInterposedTransactionManager methods provided by the weblogic.transaction.TxHelper. These methods give you the ability to get pretty exotic with the propagation of transactions into and out of WebLogic, to help you stitch together the end-to-end propagation of transactions.

As for the bad news… Well, until Tomcat grows a transaction manager of its own as a peer to WebLogic's, the scenario we started from just isn't going to work. Not that that's necessarily a bad thing – client code demarcating transactions can get some transaction architects foaming at the mouth and ranting, but that's a story for another month!

'StackOverFlow' Issues in BEA WebLogic Server

DETERMINE THE CAUSE BEFORE YOU MAKE THE PHONE CALL

BY **STEVE POZARYCKI**

A "StackOverFlow" message is usually indicative of an error in the application code of the user, an error in the Java Virtual Machine, or in BEA WebLogic Server itself.

This message is usually seen right before a Java Virtual Machine core dump or the WebLogic Server process just "goes away." It is because of either an unintentional recursive call in user/application code or a scenario where arrays of arrays of Objects can cause the stack to overflow (there are bug reports on http://java.sun.com about these types of issues). This is unfortunate, as it may require programmers to think about the implementation details of the Java Virtual Machine on which they are running. In order to investigate a "StackOverFlow" Error more thoroughly to determine the exact cause, first go through the items discussed here.

Look at any recent application code changes and see if anything could possibly be called recursively. If there is no stack trace produced for the "StackOverflow", try to add debug statements where there is suspect, code. If some application code is suspect, then you can make modifications and add the following:

```
catch ( StackOverflowError e ) {
         System.err.println("Exception: " + e );
         // Here is the important thing to do
// when catching StackOverflowError's:
         e.printStackTrace();
         // do some cleanup, destroy the thread or
unravel if possible.
}
```

The next thing you can look at involves JSPs (if you're using them). A few issues involving recursive problems have been resolved by using the following information for JSPs:

- Check within your jsp_error page to see if you have the tag <%@ page errorPage="jsp_error"%> as this would cause an infinite recursion. Please remove this and simply print the stack trace of any exceptions that could occur. This way you can find the problem in the error page. Once this is resolved you can look at the original error that gets sent to this page.
- Feedback provided to BEA Customer Support about the JSP tag was that another problem was due to bad jsp/servlet login/auth/error reporting code. Fixing this code fixed the recursion and therefore the crash.
- If you are using BEA WebLogic JSP Form Validation Tags, make sure that for the <wl:form> you don't set the action attribute to the same page containing the <wl:form> tag because this will create an infinite loop resulting in a "StackOverFlow" exception. For an example, see http://e-docs.bea.com/wls/docs61/jsp/validation_tags.html#67370.

A strange "StackOverFlow" was caused by the following code snippet within some application code. Please check to see if you are doing this in your code since a known Sun Issue (#4906193) addresses it. Instead of doing this in the application code:

AUTHOR BIO

Steven Pozarycki is a backline developer relations engineer with BEA Systems, Customer Support Group. He specializes in troubleshooting and solving complex customer issues with their mission-critical applications on BEA products. Steve holds a bachelor's degree in computer science.

CONTACT...

spoz@bea.com

REPRODUCED WITH PERMISSION FROM BEA SYSTEMS.

```
Properties p = new Properties(System.getProperties());
```

do the following:

```
Properties p = new Properties();
p = System.getProperties();
```

in order to avoid the recursive call stack trace that was observed.

A suggested “possible” workaround for some “StackOverFlow” messages is to increase the size of the thread stacks with the -Xss argument to the JVM. However, if a recursive call truly has caused this, then this option will not really help at all and will only delay the inevitable. Some background on this argument to the JVM: each Java thread has two stacks, one for Java code and one for C code. This option sets the maximum stack size that can be used by C code in a thread to the value specified. For a complete definition of the “-Xss” flag see “Non-Standard Options” at http://java.sun.com/j2se/1.3/docs/tooldocs/solaris/java.html.

If the first couple of suggestions don’t help pinpoint the problem, then try periodically collecting thread dumps of the JVM as it is running when you think the problem may occur (if it occurs at a specific time or a specific sequence of events causes the problem), usually about 5–10 seconds apart. Using this information, you may be able to find the recursive code and correct it or this may enable BEA Customer Support to have a better idea of what could be causing the problem. To collect thread dumps you need to do the following on the Java Process ID (PID):

- On all “Unix-like” platforms you can do “kill -3” before expecting the crash. “kill -3 <jvm-pid>” dumps Java threads.
- On all “Windows” platforms you can do a “kill –3” on the JVM PID. You can also do a “<CTRL> <BREAK>” in the window where the JVM is running.

The same applies to the BEA JRockit JVM as well as all platforms. You need to make sure that you do this on the root Java process. To get a tree structure of the processes on Linux, use the “--forest” option. For example, to find the processes started by user “wlsuser”, execute this: “ps -lU wlsuser –forest”. For a specific example of getting thread dumps with the BEA JRockit JVM on Linux, see http://e-docs.bea.com/wls/docs70/cluster/trouble.html#602852.

If you cannot “time” the thread dumps to get a thread dump right before the “StackOverFlow” happens, you can set the following flags to allow a thread dump to be taken of the server right before a core happens to get the state of the threads at that moment. The option is “-XX:+ShowMessageBoxOnError” option on the Sun JVM (which is not officially documented on Sun’s Web site). When the JVM crashes, the program will prompt: “Do you want to debug the problem?” You can then take a thread dump of the JVM. This option will be available on the 8.1 SP2 version of the BEA JRockit JVM when it is released. However, in that version the corresponding option will be “-Djrockit.waitonerror”.

If a binary core file is produced from a “StackOverFlow”, then you can run a debugger on the resulting core file to get a stack trace. This may help in pointing out the offending code to you. If you are unsure, then contact BEA Customer Support with this information so they can investigate the stack trace more thoroughly. If you are on a Windows platform, then a “Dr. Watson” file may be produced so please send this file to BEA Customer Support when opening a case. Otherwise, check the following “Unix” operating system values to make sure that they have already been properly set in order to generate a core file:

1. Check the “ulimit -c” (configured size of the core file) at a system and user level to make sure that it is set and that the value is not set too low to produce a meaningful core file.
2. Check the available disk space for the user. For example: Is there a disk quota?
3. Check the following parameter, which on Solaris is usually in /etc/system file and can be used to disable core files:

```
set sys:coredumpsize=0
```

4. On Linux, the coredump is turned off by default on all systems. In RedHat Advanced Server 2.1 it should be under “/etc/security”. There should be a self-explanatory file called limits.conf; within that file look for the word “core”. If it is set to “0”, then coredump files are disabled.
5. On HP-UX check the setting called “kernel parm maxdsiz” (max_per_proc_data_size, which increases the User Process Data Segment Size) from the old value of, say, 64m, to something higher like 134M.

Please get a stack trace (or back trace) from your debugger. For example, here are the commands needed when using “dbx” or “gdb”.

dbx

- ***$ java -version:*** Need to use right version of JDK
- ***$ ls /opt/bin/dbx:*** Need to know dbx location or “which dbx”
- ***$ export DEBUG_PROG=/opt/bin/dbx:*** Wherever “dbx” is located
- ***$ <path to java command>/java corefile***

Now you are in the debugger. Execute the following commands:

- ***(dbx) where:*** Shows a summary of the stack)
- ***(dbx) threads:*** Shows the state of the existing threads
- ***(dbx) quit***

gdb

- ***$ java -version:*** Need to use right version of JDK
- ***$ ls /usr/local/bin/gdb:*** Need to know gdb location or “which gdb”
- ***$ export DEBUG_PROG=/usr/local/bin/gdb:*** Wherever “gdb” is located
- ***$ <path to java command>/java corefile***

Now you’re in the debugger. Execute the following commands:

- ***(gdb) where:*** Shows a summary of the stack
- ***(gdb) thr:*** Switch among threads or show the current thread
- ***(gdb) info thr:*** Inquire about existing threads
- (***gdb) thread apply 1 bt***: Apply a command to a list of threads, specifically the back trace to thread #1
- ***(gdb) quit***

Further Information

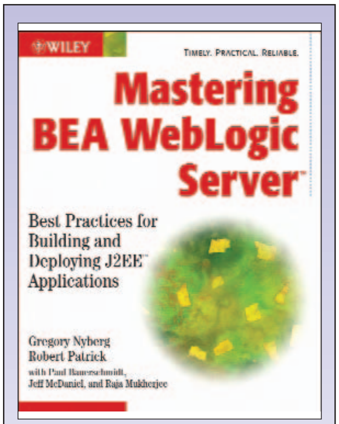
For additional information you can also go to http://support.bea.com and find some published solutions on “StackOverflows”. In the “Question” field type **“S-19795”** or **“S-19361”** to display the information from those solutions.

If none of these hints direct you towards a solution or an identifier in your application, then contact BEA Customer Support for further diagnosis. You can open a case with a valid support contract by logging in at http://support.bea.com/login.jsp

BOOK EXCERPT

Building Web Applications in WebLogic

FROM CHAPTER 1


WILEY
Timely. Practical. Reliable.
Mastering BEA WebLogic Server
Best Practices for Building and Deploying J2EE Applications
Gregory Nyberg
Robert Patrick
with Paul Bauerschmidt, Jeff McDaniel, and Raja Mukherjee

***Mastering BEA WebLogic Server, Best Practices for Building and Deploying J2EE Applications*.**

ISBN:
0-471-28128-X.
Published by Wiley

Authors:
Greg Nyberg, Robert Patrick with Paul Bauerschmidt, Jeff McDaniel, and Raja Mukherjee

Web applications are an important part of the Java 2 Enterprise Edition (J2EE) platform because the Web components are responsible for key client-facing presentation and business logic. A poorly designed Web application will ruin the best business-tier com-ponents and services. In this chapter, we will review key Web application concepts and technologies and their use in WebLogic Server, and we will provide a number of rec-ommendations and best practices related to Web application design and construction in WebLogic Server.

This chapter also provides the foundation for the discussion of recommended Web application architectures in Chapter 2 and the construction and deployment of a com-plex, realistic Web application in Chapters 3, 4, and 5.

Java Servlets and JSP Key Concepts

In this section we will review some key concepts related to Java Servlets and JavaServer Pages. If you are unfamiliar with these technologies, or if you need addi-tional background material, you should read one of the many fine books available on the subject. Suggestions include *Java Servlet Programming Bible* by Suresh Rajagopalan et. al. (John Wiley & Sons, 2002), *Java Servlet Programming* by Jason Hunter (O'Reilly & Associates, 2001), and *Core Servlets and JavaServer Pages* by Marty Hall (Prentice Hall PTR, 2000).

Characteristics of Servlets

Java servlets are fundamental J2EE platform components that provide a request/response interface for both Web requests and other requests such as XML messages or file transfer functions. In this section, we will review the characteristics of Java servlets as background for a comparison of servlets with JavaServer Pages (JSP) technology and the presentation of best practices later in the chapter.

Servlets Use the Request/Response Model

Java servlets are a request/response mechanism: a programming construct designed to respond to a particular request with a dynamic response generated by the servlet's spe-cific Java implementation. Servlets may be used for many types of request/response scenarios, but they are most often employed in the creation of HyperText Transfer Pro-tocol (HTTP) responses in a Web application. In this role, servlets replace other HTTP request/response mechanisms such as Common Gateway Interface (CGI) scripts.

The simple request/response model becomes a little more complex once you add chaining and filtering capabilities to the servlet specification. Servlets may now partic-ipate in the overall request/response scenario in additional ways, either by prepro-cessing the request and passing it on to another servlet to create the response or by postprocessing the response before returning it to the client. Later in this chapter, we'll discuss servlet filtering as a mechanism for adding auditing, logging, and debugging logic to your Web application.

Servlets Are Pure Java Classes

Simply stated, a Java servlet is a pure Java class that implements the javax.servlet .Servlet interface. The application server creates an instance of the servlet class and uses it to handle incoming requests. The Servlet interface defines the set of methods that should be implemented to allow the application server to manage the servlet life cycle (discussed later in this chapter) and pass requests to the servlet instance for pro-cessing. Servlets intended for use as HTTP request/response mechanisms normally extend the javax.servlet.http.HttpServlet class, although they may imple-ment and use the Servlet interface methods if desired. The HttpServlet class implements the Servlet interface and implements the init(), destroy(), and service() methods in a default manner. For example, the service() method in HttpServlet interrogates the incoming HttpServletRequest object and for-wards the request to a series of individual methods defined in the HttpServlet class based on the type of request. These methods include the following:

99.9% of the world won't find these screen shots terribly exciting. But if you're in the other 0.1%, yeehaw.

Use Crystal Reports 10 with your J2EE applications

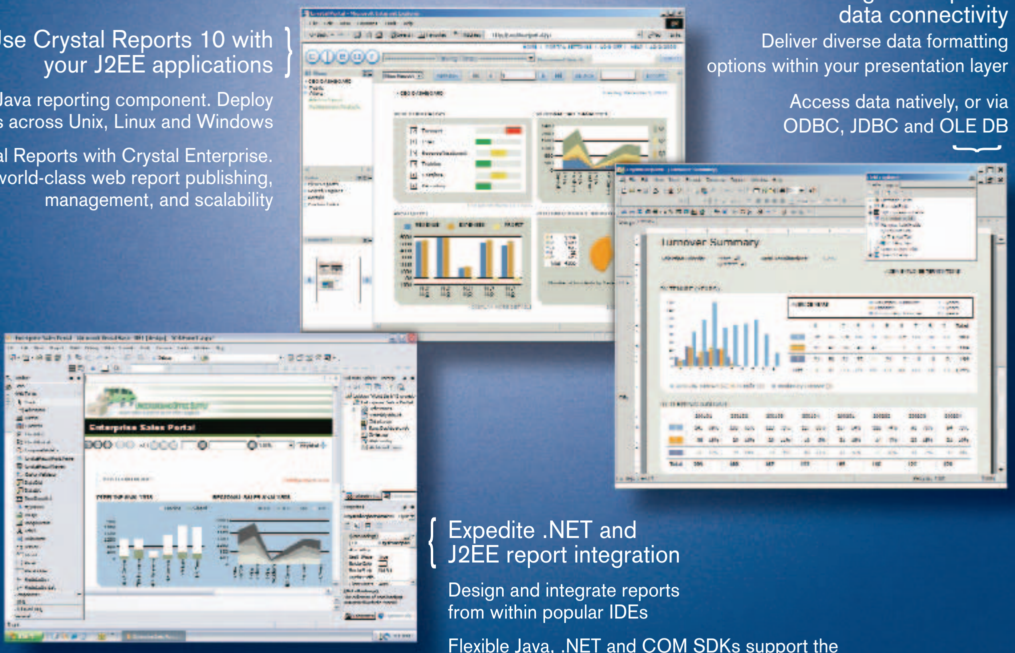
New 100% Java reporting component. Deploy reports across Unix, Linux and Windows

Extend Crystal Reports with Crystal Enterprise. Get world-class web report publishing, management, and scalability

Visual Designer simplifies data connectivity

Deliver diverse data formatting options within your presentation layer

Access data natively, or via ODBC, JDBC and OLE DB



Expedite .NET and J2EE report integration

Design and integrate reports from within popular IDEs

Flexible Java, .NET and COM SDKs support the tight integration of report interactivity including: group tree navigation, exporting, printing, and drill down

New Crystal Reports 10.

The best in business intelligence now offers the best in business reporting. New Crystal Reports® 10 is a faster and simpler way for developers to integrate dynamic data into applications and implement high-quality viewing, printing, and exporting. Learn more about Crystal Reports 10 and Crystal Enterprise™ 10, and access technical and evaluation resources at www.businessobjects.com/v10/059/ or contact us directly at 1-800-877-2340.

BUSINESS OBJECTS®

–doGet() for handling GET, conditional GET, and HEAD requests – doPost() for POST requests

–doPut() for PUT requests

–doDelete() for DELETE requests

–doOptions() for OPTIONS requests – doTrace() for TRACE requests

The doGet(), doPost(), doPut(), and doDelete() methods in HttpServlet return a BAD_REQUEST (400) error as their default response. Servlets that extend HttpServlet typically override and implement one or more of these methods to gen-erate the desired response. The doOptions() and doTrace() methods are typically not overridden in the servlet. Their implementations in the HttpServlet class are designed to generate the proper response, and they are usually sufficient.

A minimal HTTP servlet capable of responding to a GET request requires nothing more than extending the HttpServlet class and implementing the doGet() method.

WebLogic Server provides a number of useful sample servlets showing the basic approach for creating HTTP servlets. These sample servlets are located in the samples/server/examples/src/examples/servlet s subdirectory beneath the Web-Logic Server home directory, a directory we refer to as $WL_HOME throughout the rest of the book. We will examine some additional example servlets in detail during the course of this chapter. These example servlets are available on the compan-ion Web site for this book at http://www.wiley.com/compbooks/masteringweblogic.

Creating the HTML output within the servlet's service() or doXXX() method is very tedious. This deficiency was addressed in the J2EE specification by introducing a scripting technology, JavaServer Pages (JSP), discussed later in this chapter.

Servlets Have a Life Cycle

A servlet is an instance of the servlet class and has a life cycle similar to that of any other Java object. When the servlet is first required to process a request, the application server loads the servlet class, creates an instance of the class, initializes the instance, calls the servlet's init() method, and calls the service() method to process the request. In normal servlet operation, this same instance of the servlet class will be used for all subsequent requests.

Servlets may be preloaded during WebLogic Server startup by including the <load-on-startup> element in the web.xml file for the Web application. You can also pro-vide initialization parameters in this file using <init-param> elements. WebLogic Server will preload and call init() on the servlet during startup, passing the specified initialization parameters to the init() method in the ServletConfig object.

An existing servlet instance is destroyed when the application server shuts down or intends to reload the servlet class and create a new instance. The server calls the destroy() method on the servlet prior to removing the servlet instance and unloading the class. This allows the servlet to clean up any resources it may have opened dur-ing initialization or operation.

Servlets Allow Multiple Parallel Requests

Servlets are normally configured to allow multiple requests to be processed simultane-ously by a single servlet instance. In other words, the servlet's methods must be thread-safe. You must take care to avoid using class- or instance-level variables unless access is made thread-safe through synchronization logic. Typically, all variables and objects required to process the request are created within the service() or doXXX() method itself, making them local to the specific thread and request being processed.

BEST PRACTICE: Servlets that allow multiple parallel requests must be thread-safe. Do not share class- or instance-level variables unless synchronization logic provides thread safety.

Servlets may be configured to disallow multiple parallel requests by defining the servlet class as implementing the SingleThreadModel interface:

```
...
public class TrivialSingleThreadServlet
extends HttpServlet implements SingleThreadModel
{
public void init(ServletConfig config) throws
ServletException
{
super.init(config); System.out.println("Here!");
}
...
```

This simple change informs the application server that it may not process multiple requests through the same servlet instance simultaneously. The application server can honor this restriction in multiple ways: It may block and queue up requests for processing through a single instance, or it may create multiple servlet instances as needed to fulfill parallel requests. The servlet speci-

Book Review

Mastering BEA WebLogic Server, Best Practices for Building and Deploying J2EE Applications is a new book for developers and architects. It covers many common areas experienced in the development and rollout of applications through J2EE and the use of BEA WebLogic Server. While the main audience is towards application developers and architects that have had some exposure to J2EE concepts and experience in development, beginners will find resource references and design patterns that will save a lot of extra work down the road.

The book covers Web applications, EJBs, and several other areas. There are three chapters on EJBs. Great care and time is spent discussing them in the BEA WebLogic Server context, including strategies in use, features, how to build, and typical packing and deployment of them. The remaining chapters cover JMS, security; administration/deployment; optimizing WebLogic Server performance, development, and production environment best practices; and developing/deployment of Web service applications. The first five chapters are on Web applications. Those covered focused on key concepts, best practices, architecture, package and deployment, and a sample walk-through.

Each section in a chapter normally contained best practice tips that are highlighted. Sample code was throughout the book, but there wasn't such an amount that you were reading code for the majority of a topic. Downloads are available to look through that exemplify the concepts the section is trying to portray. I like the fact that the authors bring out sections of BEA WebLogic Server that really aid in development or optimize performance, like the discussion of wl:cache.

In all, I found this book easy to read. I wished that there was more content on the security and deployment sections, but the information that was there established a strong basis for any team working with BEA WebLogic Server. I highly recommend this as a resource for any architect working with WebLogic Server. It will be a definite piece in my library.

by Eric Ballou (iknowbea@yahoo.com

fication does not dictate how application servers should avoid parallel processing in the same instance.

WebLogic Server satisfies the single-threaded requirement by creating a small pool of servlet instances (the default pool size is five) that are used to process multiple requests. In older versions of WebLogic Server, multiple parallel requests in excess of the pool size would block waiting for the first available servlet instance. This behavior changed in WebLogic Server 7.0. The server now creates, initializes, and discards a new instance of the servlet for each request rather than blocking an execute thread under these conditions. Set the pool size properly to avoid this extra servlet creation and ini-tialization overhead.

You can configure the size of the pool at the Web application level using the single-threaded-servlet-pool-size element in the weblogic.xml deployment descriptor. If you choose to employ single-threaded servlets in high-volume applications, consider increasing the pool size to a level comparable to the number of execute threads in the server to eliminate the potential overhead required to create extra servlet instances on the fly to process requests.

Although instance variables are safe to use in single-threaded servlets, class-level static variables are shared between these instances, so access to this type of static data must be thread-safe even when using the SingleThreadModel technique. Deploy-ing and executing this TrivialSingleThreadServlet example verifies this pool-ing behavior in WebLogic Server. The first servlet request causes WebLogic Server to create five instances of the servlet, as evidenced by five separate invocations of the init() method and the subsequent writing of five “Here!” messages in the log.

WLDJ ADVERTISER INDEX

ADVERTISER	URL	PHONE	PAGE
BEA Systems	http://dev2dev.bea.com/medulla	800-817-4BEA	2,3
Compuware	www.compuware.com	800-COMPUWARE	57
Crystal Decisions	www.businessobjects.com/v10/059	800-877-2340	55
Cyanea	www.cyanea.com/wldj/underpar.html	877-CYANEA8	9
Edge East 2004	www.sys-con.com	201-802-3069	27-42
H&W Computer	www.hwcs.com/wsd1.asp	800-338-6692	60
HP	http://devresource.hp.com/d2d.htm	800-752-0900	19
Java Developer's Journal	www.sys-con.com	888-303-5282	43
Mercury Interactive	www.mercuryinteractive.com/optimizej2EE	800-831-8911	15
NetIQ	www.netiq.com/solutions/web	408-856-3000	11
PowerBuilder Developer's Journal	www.sys-con.com/pbdj	800-303-5282	26
Quest Software	http://java.quest.com/jprobe/wldj	800-663-4723	59
SalesForce.com	www.sforce.com	800-NO SOFTWARE	21
SYS-CON Reprints	www.sys-con.com	201-802-3026	51
WebLogic Developer's Journal	www.weblogicdevelopersjournal.com	888-303-5282	43
Web Services Journal	www.sys-con.com	888-303-5282	47
Wily Technology	www.wilytech.com	888-GET-WILY	5

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

LINUX

– continued from page 10

Sometimes tuning is necessary as the workload has changed. Maybe it’s more complex as updated code or design has been migrated into production, or perhaps the application now supports a larger user base. Whatever the reason, tuning requires careful validation and, as is often the case, only performance monitors can show the overall impact to the whole application.

Start at the operating-system level and work up through the different stacks. Review the current performance measurements and use tools like Transaction Tracer to quickly show what set of components are responsible for the majority of elapsed time within a given request.

Review the load average, runnable tasks, and disk and swap activity reported per interval by the operating system. Consider reallocating files if disk activity is only on one device.

Perhaps the number of concurrent processes has increased (additional instances on the same machine). If the load average or runnable tasks are high, review what other processes are competing for resources. Maybe deploying application instances on separate machines would allow for the workload to be distributed across many machines, thus lowering the resource usage of an individual machine.

When tuning the JVM, look at the memory usage and garbage collection that is being used. The JVM tuning document, http://edocs.bea.com/wljrockit/docs81/tunelnux, is a good resource that outlines the garbage collection and thread options that are available. The WebLogic application must also be reviewed. The article “Turning WebLogic Server,” http://edocs.bea.com/wls/docs81/perform/WLSTuning.html, is an excellent starting point. Then use the data collected to validate the performance.

Review the execute queue and thread pools within WebLogic. Are requests waiting to execute? Are enough connections available in the JDBC poll to process the expected workload, without over allocation?

Resources

Talk to your vendors about your deployment plans. Often they see multiple approaches to solving issues and can sometimes share insights based on experiences. The following Web sites will also help in building a Linux WebLogic deployment:

- www.wilytech.com
- www.bea.com/framework.jsp?CNT=index.htm&FP=/content/solutions/resource/linux
- http://e-docs.bea.com

Summary

I hope this article has provided the background for a production BEA WebLogic and Linux deployment within your environment. The application server will only perform as fast as WebLogic can receive requests and retrieve data from the back end, so tuning is critical.

The touch points we outlined and tuning considerations are a starting point. Your application and environment will have other touch points. But know that you are not alone on your Linux and BEA WebLogic deployment!