SYS-CON

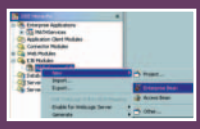
CONTENT MANAGEMENT • EXTENSIONS • PROFILES

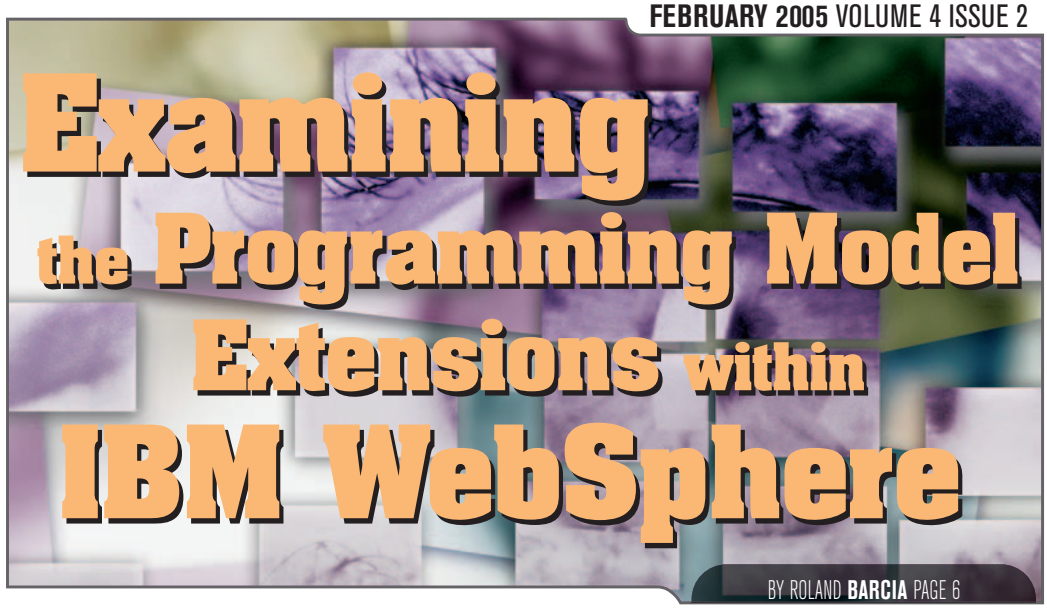
WebSphere® JOURNAL

www.SYS-CON.com/WebSphere

The World's Leading Independent WebSphere Resource

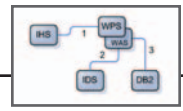
FEBRUARY 2005 VOLUME 4 ISSUE 2





Examining the Programming Model Extensions within IBM WebSphere

BY ROLAND **BARCIA** PAGE 6



Open Up!

Portals–Get what you want easily

company, product and service names maybe trademarks or service marks of others.

Ask the Experts

Prolifics is 1 of only 3 WebSphere Service Providers retained by IBM. Open up with portals! With 26 years of industry expertise, Prolifics' certified WebSphere portal experts deliver a personalized, secure, unified view to your enterprise assets.

Our portal solutions build agility into your business — enabling you to reach new markets, achieve customer satisfaction beyond expectation, and ease collaboration within your enterprise. Our teams of specialists work together with your business experts from gathering the business requirements, translating them to a technical specification, to delivering the final production application.

Portal Executive Assessment

Prolifics experts help you articulate the value of portals to your business sponsors and perform a quick business review identifying portal benefits as it applies to your organization.

Portal Workshops and Proof of Concepts

Prolifics demonstrates valuable portal features and translates the breadth of your portal objectives into a step-by-step roadmap to meet your current and future needs.

Portal Jump Start

Prolifics' consultants work with you from the start to install, configure and jump start your team on using WebSphere Portal Software.

Portal Development and Mentoring

Prolifics' innovative approach to building portals drives down your overall cost and transforms the way you conduct business. We mentor and arm your team with the skills to manage, maintain and extend the portal environment.

Prolifics

Winner of 2003 IBM BUSINESS PARTNER LEADERSHIP AWARD

116 John Street, New York, New York 10038
800.675.5419
solutions4websphere@prolifics.com
www.prolifics.com

Prolifics. IBM Business Partner

Deliver better, safer software

in less time, with fewer resources.

Parasoft® Automated Unit Testing & Coding Standards Analysis: specifically designed for under-staffed, over-committed development organizations like yours.

Parasoft products ensure that your entire development organization is meeting requirements and adhering to a common standard. Parasoft products automatically identify security violations, memory errors, runtime exceptions, common defect patterns, and violations of best coding practices.

Increase development effectiveness by allowing:

- **Developers** to correct problems early and reduce time spent chasing bugs
- **Architects** to ensure that development is adhering to a common standard
- **Managers** to improve project visibility, decision making and predictability

The Parasoft product family supports all your major development environments including Java, Web Services, C/C++, .NET and more.

For details go to: www.parasoft.com/Products

Email: Products@parasoft.com Call: 888-305-0041 x3307

PARASOFT®
We make software work.™

Founded in 1987, Parasoft's clients include IBM, HP, DaimlerChrysler and over 10,000 companies worldwide.

Are Application Servers, XML, and Web Services Fads?

BY JACK MARTIN

I received a lot of feedback last month on my "Will Every Child Be Left Behind?" editorial. A lot of you related your collective disappointment in the current state of our public schools. What surprised me was the number of readers who thought that application servers, XML, and Web services were current buzzwords, not core foundations of modern technology. One reader in particular thought that application servers, XML, and Web services were about as important as black holes and string theory, which in his view were unnecessary for the basics of knowledge. I think they are also important.

What surprises me is that ***WebSphere Journal*** is all about application servers, XML, Web services, and other related technologies that make up the world of modern information technology and its future. I wouldn't have anyone believe that the basic underpinnings of the world economy were anything less than a building block for all young people – not just those planning on entering the economics field – so they have a basic understanding of how things work, and how they may effect them now and in the future.

While I know that technologies will always come and go, it seems to me that application servers, XML, and Web services are now so widely deployed across all industries (and the rollout continues into every nook and cranny) that most people in the field would view this as a long-term trend, not the latest technology *du jour*. Maybe I'm wrong – I've been wrong before – but I think we'll be living with these technologies for a very long time to come and people should know about them.

Another reader felt that needing to know about application servers, XML, and Web services was like needing to know about catalytic converters and how they operate. Catalytic converters are an excellent example of a key technology that most people don't understand but should.

What comes out of catalytic converters is the leading cause of cancer in most neighborhoods, not cigarette smoking. I know that all of the nonsmokers will howl that cigarettes are the bane of the modern world. Your government (for that matter, everyone around you) will tell you if you are a smoker that the single best thing you can do is to quit. There are countless people who are so intolerant of cigarette smoke they will throw a fit if you light up in front of them, and go on and on about the dangers of secondhand smoke. They're right – cigarette smoking and secondhand smoke are not good for you, but what comes out of a catalytic converter is actually a lot worse, and there is a lot more of it for you to breathe everyday.

The fact is that when you burn anything all you are doing is converting a solid or liquid into a gas. It weighs the same; it is just a gas and no longer solid or liquid – nothing more, nothing less.

A pack of cigarettes weighs just about one ounce including all of the packaging. Most smokers do not smoke the packaging, but I'll throw it into the equation. So if you smoke one pack of cigarettes in its entirety, you have converted one ounce of solid matter to gas – simple. That single ounce goes into the air, people breathe it, and it gets on everything because it is a gas. Breathing cigarette smoke is very bad for your health as it causes cancer.

One gallon of gasoline weighs six pounds and three ounces, which equals 99 ounces. When we use gasoline, we burn every drop of it; it has no packaging. So if you burn one gallon of gasoline in its entirety, you have converted 99 ounces of liquid matter to gas – simple. That 99 ounces goes into the air, people breathe it, and it gets on everything because it is a gas. Breathing gasoline smoke is very bad for your health as it causes cancer.

I know that some of the gasoline and a much smaller part of the cigarette are either used for power conversion or are absorbed, but let's put that aside.

So in simple terms a car that gets 20 miles to the gallon and is driven 12,000 miles a year burns 600 gallons of gasoline, which is 59,400 ounces. You would have to smoke almost 163 packs of cigarettes a day to equal one car's worth of air pollution for the same year. Which is worse for you? If I could only choose one, I would choose to smoke cigarettes. This is why young people should know about catalytic converters and application servers.

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

WebSphere® JOURNAL

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

EDITOR-IN-CHIEF JACK MARTIN
EXECUTIVE EDITOR GAIL SCHULTZ
CONTRIBUTING EDITOR PATTI MARTIN
PRODUCT REVIEW EDITOR JAY JOHNSON
EDITOR NANCY VALENTINE
ASSISTANT EDITORS NATALIE CHARTERS, SETA PAPAZIAN

WRITERS IN THIS ISSUE
Chae An, Roland Barcia, David Beveridge, Larry Carvalho, Brian Garr, Sonny Hastomo, Kevin Haverlock, Joseph R. Marques, Jack Martin, Ellen Matheson McKay, Aditya Rustgi, Barry Searle, Thomas Stober

SUBSCRIPTIONS
For subscriptions and requests for bulk orders, please send your letters to Subscription Department.
888-303-5282 201-802-3012
SUBSCRIBE@SYS-CON.COM
Cover Price: $8.99/Issue Domestic: $149/YR (12 Issues)
Canada/Mexico: $169/YR Overseas: $179/YR
(U.S. Banks or Money Orders)
Back issues: $15 U.S. $20 All others

PRESIDENT AND CEO FUAT KIRCAALI
PRESIDENT, SYS-CON EVENTS GRISHA DAVIDA
GROUP PUBLISHER JEREMY GEELAN
SENIOR VP, SALES & MARKETING CARMEN GONZALEZ
VP, INFORMATION SYSTEMS ROBERT DIAMOND
VP, SALES & MARKETING MILES SILVERMAN
PRODUCTION CONSULTANT JIM MORGAN
FINANCIAL ANALYST JOAN LAROSE
CREDITS & ACCOUNTS RECEIVABLE STEVE MICHELIN
ACCOUNTS PAYABLE BETTY WHITE
ADVERTISING DIRECTOR ROBYN FORMA
ADVERTISING MARKETING DIRECTOR DENNIS LEAVEY
ADVERTSING SALES MANAGER MEGAN MUSSA
ADVERTISING SALES MANAGERS KRISTIN KUHNLE, DOROTHY GIL, KIM HUGHES
LEAD DESIGNER LOUIS F. CUFFARI
ART DIRECTOR ALEX BOTERO
ASSOCIATE ART DIRECTORS RICHARD SILVERBERG, TAMI BEATTY, ANDREA BODEN
WEB DESIGNERS STEPHEN KILMURRAY, MATTHEW POLLOTTA
ONLINE EDITOR MARTIN WEZDECKI
CIRCULATION SERVICE COORDINATORS EDNA EARLE RUSSELL, LINDA LIPTON, MONIQUE FLOYD
CUSTOMER RELATIONS MANAGER BRUNI STAROPOLI

EDITORIAL OFFICES
SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645
Telephone: 201 802-3000 Fax: 201 782-9637
SUBSCRIBE@SYS-CON.COM
WebSphere® Journal (ISSN# 1535-6914) is published monthly (12 times a year).
Postmaster send address changes to:
WebSphere Journal, SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645

WORLDWIDE NEWSSTAND DISTRIBUTION
CURTIS CIRCULATION COMPANY, NEW MILFORD, NJ

NEWSSTAND DISTRIBUTION CONSULTANT:
GREGORY ASSOCIATES, W.R.D.S.
732 607-9941, BJGASSOCIATES@CS.COM

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

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

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

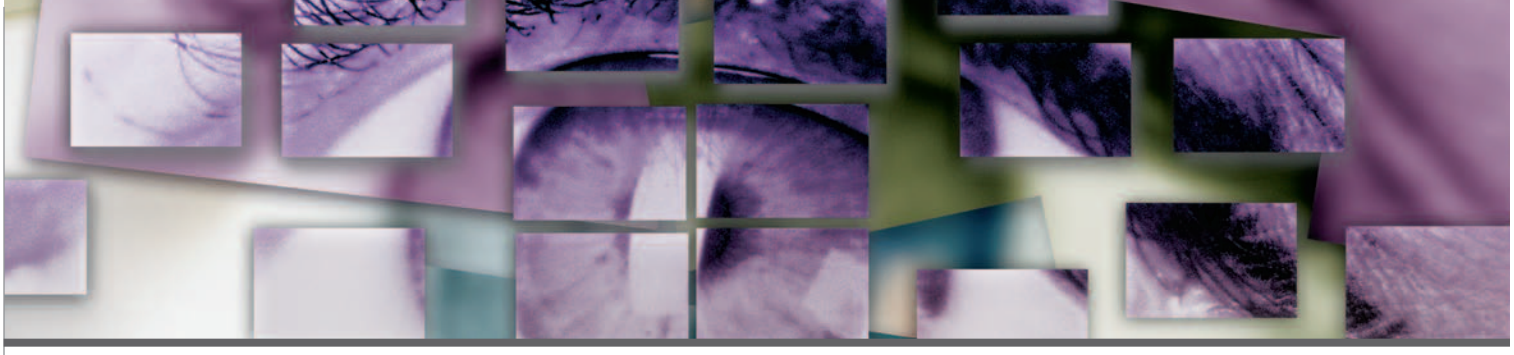
Unless of course, you've discovered the power of Wily. No other software offers our level of insight. Which means you'll be able to monitor transactions and collaborate with colleagues in real-time. Even more important, you'll be able to optimize performance across the entire application—end-to-end.

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

Get Wily.™

1 888 GET WILY | wilytech.com

wily technology



Examining the Programming Model Extensions

Within IBM WebSphere Application Server version 6

BY ROLAND BARCIA

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

Back in version 4 of IBM WebSphere Application Server (WAS), IBM released the Enterprise Edition of WAS. In version 5, IBM continued to release the Enterprise Edition (later renamed IBM WebSphere Business Integration Server Foundations in v5.1) with even more functionality. In version 5, the heart of the Enterprise Edition functionality was process choreographer, a Java 2 Enterprise Edition (J2EE)-based workflow engine. However, the Enterprise Edition also contains a slew of other technology sometimes referred to as the WAS Programming Model Extensions (PME). The goal of these programming model extensions is to provide solutions to problems that are difficult and sometimes impossible to solve using existing J2EE technology.

In WAS v5, customers needed to upgrade to the Enterprise Edition to benefit from WAS PMEs, leaving a large portion of WAS developers out of luck. However, in version 6 most of the PMEs have now been included in all editions of WAS. This means that all WebSphere version 6 users can benefit from these extensions without upgrading to a higher end workflow engine.

In this article, I will go over some of these WAS PMEs. I will discuss each PME in terms of the problem it solves. In addition, I'll talk about the work being done to add some of these extensions to future versions of the J2EE specification.

Overview of Extensions

In order to put some structure around this article, I am going to categorize the extensions by their problem space:

- Multi-threading and scheduling extensions
- Transactional enhancements and EJB entity bean extensions
- Other extensions

Multi-threading and Scheduling

In this section, I am going to discuss some of the WAS PME features in the multi-threading and scheduling space.

Asynchronous Beans and CommonJ Work Manager

The J2EE platform shields developers from low-level system coding such as multithreaded programming. For that reason, there are many restrictions in the J2EE programming model when it comes to multithreading. For example, in J2EE 1.4, it is illegal to create your own Java threads in a J2EE container. WAS has no knowledge of these threads so it cannot propagate things such as a security context. In addition, because plain Java threads are random they cannot be managed by a thread pool; without someway to manage these threads, applications can lose control of machine resources. However, there are certain use cases that could benefit from the ability to do work asynchronously. Traditionally in J2EE such use cases are implemented with the Java Messaging Service (JMS). JMS forces you to define destinations, create messages, and code message driven beans to consume messages. JMS is a great platform for doing asynchronous work, especially for integration scenarios, but, in certain situations all a developer wants to do is accomplish concurrent processing. For example, you may have a use case where an application needs to retrieve data from two or three back end independent data sources. In order to speed up performance, the application could run these queries concurrently. With JMS, this is quite complex to write. However, WAS provides a framework (called asynchronous beans) of classes that allow programmers to have work executed on separate threads. In addition, asynchronous beans can also have activities performed in

response to alarms or events. At the same time, it ensures that these activities are executed using the "logically correct" J2EE context. This model represents a very interesting compromise between the loosely coupled approach used with messaging and the tightly coupled approach of traditional J2EE programming that requires a single thread of execution.

WAS developers can create asynchronous beans instead of Java threads. Asynchronous beans allow a developer to code the work to be done on a separate thread. (It is important to note that because of work being done to standardize asynchronous beans, the asynchronous bean framework is also referred to as CommonJ WorkManager. This means that there are two APIs, one in com.ibm package referred to as asynchronous bean API, and the other in the commonj package referred to as CommonJ WorkManager API. For the purpose of this article we will refer to asynchronous beans and CommonJ WorkManager interchangeably; they are both configured the same way.) There are three types of asynchronous beans:

- ***Work objects:*** Instances of classes that implement a special work interface. Essentially, the Work interface is the same as the Java runnable interface. However, Work instances can be submitted to a WorkManager (discussed below) for asynchronous execution. Unlike a plain runnable, Work inherits the J2EE context of their creator rather than the J2EE context of the thread pool where they actually run. Works allow a J2EE component to run operations in the background and asynchronously. The code that runs in the background behaves as if it were part of the J2EE application that instantiated the work.
- ***Event listeners:*** These listeners are also asynchronous by nature. An application can create an event listener and subscribe to events. When the event occurs, the listener will be notified and will run the logic to handle the event. The problem with listeners in a J2EE container is that, since the listener runs asynchronously in its own thread, it won't be able to use the J2EE context of the subscriber. The asynchronous beans provide a generic notification framework that allows a J2EE component to subscribe a listener to generic events and have the listener execute its logic under the same J2EE context of the subscriber.
- ***Timer listeners:*** (Not to be confused with EJB timers described later.) Special listeners that will be fired when a timer goes off. In the asynchronous bean framework, a timer can be created by a J2EE application. The listener will be invoked on a separate thread and it will inherit the same J2EE context as the timer creator. The code sample below shows an example of a timer.

```
public class StockQuoteTimerListener implements
TimerListener {
  String context;
  String url;
    public StockQuoteTimerListener(String context,
String url){
        this.context = context;
        This.url = url;
    }
    public void timerExpired(Timer timer) {
        System.out.println("Timer fired. Context ="+
((StockQuoteTimerListener)timer.getTimerListener()).
getContext());

  }
     public String getContext() {
      return context;
     }
  }
```

Timers are not persistent in WAS. If you need persistent services, the WAS scheduler or EJB timer service should be used. (In version 5, the Enterprise Edition provided alarms instead of timers. Timers are a replacement for alarms but the alarms are still supported.)

As I mentioned, the J2EE context can flow from one J2EE component to another as long as they are called on the same thread. The asynchronous bean framework makes it possible to transfer all or some of the context to separate threads. The administrator can also configure which parts of the context are going to be transferred when a WorkManager is created. The following contexts can optionally be configured to propagate:

- Security (J2EE security context)
- Work area (another WAS PME explained later in this article)
- Internationalization context (another WAS PME explained later in this article)
- Application profiling definitions (another WAS PME explained later in this article)

In addition, the java:comp/env local namespace is always passed along with asynchronous beans.

Due to the asynchronous nature of this programming model, transactional contexts are not passed to the asynchronous beans. The asynchronous beans get their own local transaction, or they can start global transactions if XA resources are involved.

Asynchronous beans should stay away from caching connections. They can cache connection factories and then get a connection, use it, and release it at the end of the asynchronous execution. That way connection pooling is used correctly.

FIG 1: TIMER AND WORK MANAGERS

The WorkManager is the heart of the asynchronous bean framework. A WorkManager

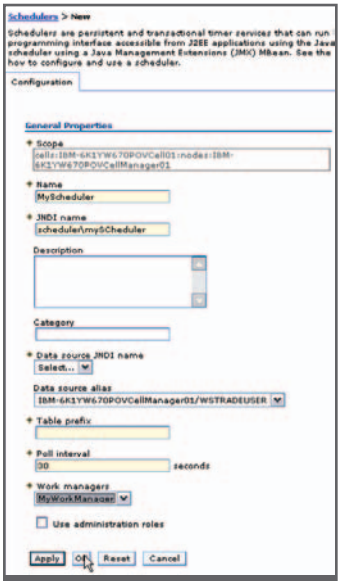


FIG 2: SCHEDULER

FIG 3: EJB TIMER SETTINGS

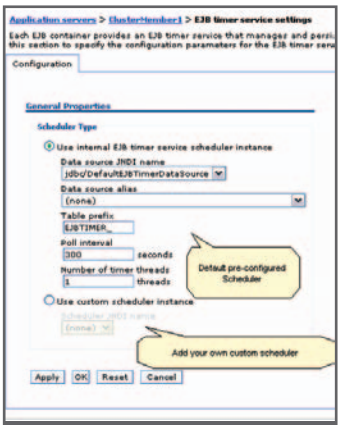


FIG 4: SCHEDULER TYPE

is essentially a thread pool. It runs Work instances asynchronously and controls the transfer of J2EE contexts to those threads. The WorkManager is the anchor point that allows developers to create asynchronous beans. Applications will look up the WorkManagers using JNDI. Once an application has obtained a WorkManager, it can use it, for example, to submit Work instances. The WorkManager returns a WorkItem to the application, which can use it to monitor the progress of the work. Applications can define resource references to work managers.

For timers, timer managers are configured on the server much the same way as work managers (see Figure 1).

The asynchronous bean framework is the basis for two new JSRs which have been submitted jointly by IBM and BEA. These are JSR 236 (Timers for Application Servers) and JSR 237 (Work Managers for Application Servers).

For further information on asynchronous beans, see the following link: http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/welc6tech_asb.html.

Scheduler Service

Prior to J2EE 1.4, the J2EE specification defined no solution for scheduling. Applications that required scheduling to run batch jobs or scheduled events needed to do so outside the application server. In version 5, WAS introduced the scheduler service in the Enterprise Edition. Applications could now write batch jobs and scheduled events that ran inside an application server. Batch applications could now benefit from container managed transactions and J2EE security. A developer just had to provide a stateless session bean class and use the provided home and remote interface. A scheduler would then be configured administratively. Tasks could be scheduled using the scheduler API either in Java code or wsadmin. The scheduler service continues to function in WAS Application Server version 6. Figure 2 shows the administrative console screen for creating a scheduler.

Applications can then treat the scheduler as a resource and create tasks as shown:

```
//lookup Scheduler as a Resource
context = new InitialContext();

Object schedulerObj =
context.lookup("java:comp/
env/scheduler/MyScheduler");

scheduler = (Scheduler) schedulerObj;

//Lookup Task Handler as an EJB
Object taskHandlerObj = context.lookup("java:comp/
env/ejb/StockTickerPublisher");
taskHandlerHome =
(TaskHandlerHome) PortableRemoteObject.narrow(
taskHandlerObj,
TaskHandlerHome.class);
//Create Task
taskInfo = scheduler.createBeanTaskInfo();
//Associate TaskHandler with Task
taskInfo.setTaskHandler(taskHandlerHome);
taskInfo.setStartTimeInterval(startInterval
+ "minutes");
taskInfo.setRepeatInterval(repeatInterval + "min-
utes");
taskInfo.setNumberOfRepeats(repeats);
taskInfo.setName(taskName);
//Create Task in Scheduler
TaskStatus status = scheduler.create(taskInfo);
```

WAS provides the home and remote interface for the task handler object. Developers only need to provide a bean class with a special process method:

```
//Example task handler
public void process(TaskStatus taskStatus)
{
  try
  {
        publisherBO.publishStockTicker();
  }
  catch (StockException e)
  {
        throw new EJBException(e.getLocalizedMessage
(),e);
  }
}
```

In J2EE 1.4, the EJB timer service is introduced to the specification. The EJB timer service allows any EJB component that implements the TimerObject interface to be called by some scheduling component. Any EJB type can be used (session bean, entity bean, or message driven bean) as a timed object. The timer service also defines a standard interface called TimerService for scheduling events. In version 6, the EJB timed objects is a special type of WAS TaskHandler. So the EJB timed components build upon the same scheduling infrastructure provided in version 6. By default, an internal scheduler instance is

used to manage those tasks, and they are persisted to a Cloudscape database associated with the server process.

The EJB timer service can be configured on the desired server (see Figure 3).

You can choose to use the default pre-configured scheduler or your own (see Figure 4).

Any EJB type can implement the TimedObject interface and implement the ejbTimeout() method:

```
import javax.ejb.Timer;
import javax.ejb.TimedObject;
import javax.ejb.TimerService;

public class MyBean implements SessionBean,
TimedObject {

  // This method is called by the EJB container upon
Timer expiration.
  public void ejbTimeout(Timer theTimer) {

    // Any code typically placed in an EJB method
can be placed here.

     String whyWasICalled = (String) Timer.getInfo():
     System.out.println("I was called because of"+
whyWasICalled);
  } // end of method ejbTimeout
```

Keep in mind that you cannot use the scheduler API to schedule tasks for EJB-based timers or use the TimerService API to configure WAS TaskHandlers.

For further information on the scheduler service, see the following URL: http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/welc6tech_sch.html.

Transactional Enhancements and EJB Entity Bean Extension

In this section I will discuss some extensions that deal with transactions, such as single phase resources inside an XA transaction. I will also discuss some extensions to entity beans, such as dynamic query and application profiles.

Last Participant Support

One of the major motivations behind J2EE application servers is the ability to provide enterprise applications with a transaction manager that can coordinate transactions across multiple resources. J2EE application server can coordinate transactions across multiple resources as long as they support the XA specification. In a perfect world, all back end resources would have support for XA. However, since this is not a perfect world, this is not the case. There are some back end systems that do not support XA. For example, distributed platforms often communicate with legacy systems that do not support XA, such as IBM, CICS, systems.

Ordinarily, all participants in a two-phase commit process must be two-phase commit-capable. With the last participant support (LPS), you can use a single one-phase commit (1PC)-capable resource with any number of two-phase commit (2PC)-capable resources in the same global transaction. The global transaction commit processing still takes place in two phases. In phase one, all the two-phase commit resources are prepared using the two-phase commit protocol. During phase two, the one-phase commit resource is called to commit first if all the two-phase resources successfully prepared. This way, if the 1PC resource fails, WAS can simply roll back the other resources. On the other hand, if the 1PC commit resource succeeds, WAS can commit the remaining resources. Figure 5 illustrates the process.

Keep in mind that LPS is not a complete substitute for using a resource that fully supports 2PC. It should only be used when you want to commit a one-phase capable resource (that has no 2PC support) with other 2PC resources and you want to get as close to 2PC as possible. LPS introduces a hazard of a mixed commit result because 1PC resources can't completely participate in the 2PC protocol. For example, if the WAS doesn't get a response

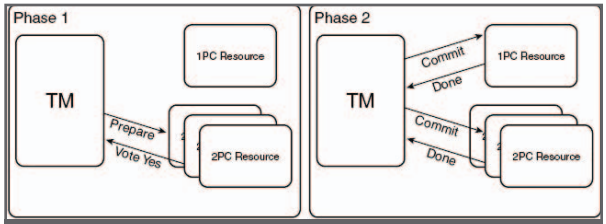


FIG 5: TWO-PHASE COMMIT PROCESS

from the 1PC resource, it has no way of knowing the status. Thus, WAS can't know whether the 1PC resource committed or not and thus does not know how to proceed. In this case, manual intervention will be required to resolve the state of the data in the 1PC resource. No special programming is needed to use LPS; it only needs to be enabled using WebSphere Application Server administration.

For more information on last participant support, see http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/lao/ref/rla_rlinks.html.

Activity Sessions

With LPS, you can only have at most one nonphase capable resource in your XA transaction. However, if you have multiple one-phase capable resources and you need some level of coordination between them, WAS provides the activity session service. Activity sessions extend the JTA transaction model by providing an alternative unit of work to deal with 1PC. Activity sessions provide several features:

1. ***Client-side demarcation of and coordination of multiple one-phase units of work:*** The activity session service captures the commit operations of each local and global transaction and upholds them. At the end of the activity session, the container will go back and commit (or roll

back) each individual transaction. Keep in mind that the activity session, like LPS, is no substitute for two-phase commitment control, when it comes to data integrity. An activity session may result in a mixed outcome, if some of the single phase resources successfully get committed before another resource fails to commit. In that case, the activity session service will allow the programmer to retrieve the list of resources that were committed and those whose state is uncertain.

2. ***Distributed context distribution using the OMG activity service:*** In order to support a declarative model, activity sessions define propagation of activity sessions from one component to another. The activity session context propagates according to the standards defined by the OMG activity service. See JSR 95 description at www.jcp.org/en/jsr/detail?id=95.
3. ***Extensions to local transaction support:*** WebSphere Application Server supports the notion of local transaction containment (LTC), which allowed the container to deal with unresolved resource manager local transactions and the resources associated with them. An LTC is created by default when an XA transaction is absent. Activity sessions can be used to extend the life of an LTC.
4. ***Extended entity bean life cycle support:*** The entity EJB can then be configured to be activated at the activity session boundaries, rather than at the transaction boundaries. The container will then keep the entity EJB active for the duration of the activity session – which corresponds to the duration of the unit of work – but no global transaction is needed to achieve this result. We often refer to this scenario as the "Commit option C plus" in that it offers a nice compromise between option C and option A by allowing the client to closely control the activation boundaries without resorting to XA transactions all the time.
5. ***Association of activity session with HttpSession to scope contexts to the life cycle of HTTP clients:*** You can scope an activity session to an HTTP session to help scope the life of activity sessions in Web applications. This is a powerful concept that allows you to mimic user level transactions which are made up of multiple resource transactions. Combined with the extended dntity bean life cycle support, you can interact with entity bean data without them reloading for the life of an activity session. This technique makes it possible to implement some long-running transaction semantics.

There are two approaches to implementing activity sessions in your applications: the declarative (or container-managed) approach and the programmatic (or bean managed) approach. The programming model for activity sessions and the terminology which relates to the activity session service closely resemble the standard definitions that apply to transaction management, making it simple for a J2EE programmer to become familiar with the activity session semantics. Activity session scopes can be defined either through declarations that apply to EJB methods invocations (in a very similar way as for declarative demarcation of transactions) or servlets – or programmatically, using the UserActivitySession APIs – which are semantically very similar to the standard UserTransaction APIs.

Figure 6 shows an example of a scenario. Here you have two different entity beans each mapped to a different 1PC resource. The EJB methods are demarcated much like you would regular transactions only using ActivityService demarcations.

Activity sessions are not just bound to local resources. They can even be used to group multiple 2PC.

Activity sessions are being worked on for future versions of J2EE as *JSR 95: J2EE TM Activity Service for Extended Transactions (URL)*.

For more information on the activity service, see: http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/welc6tech_as.html.

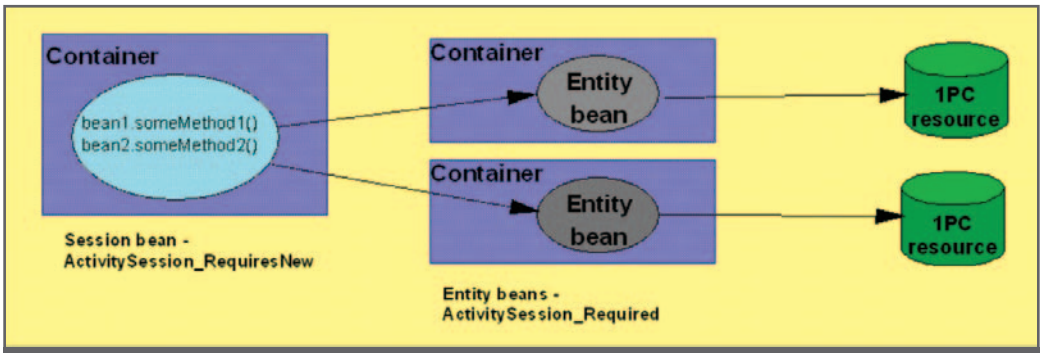


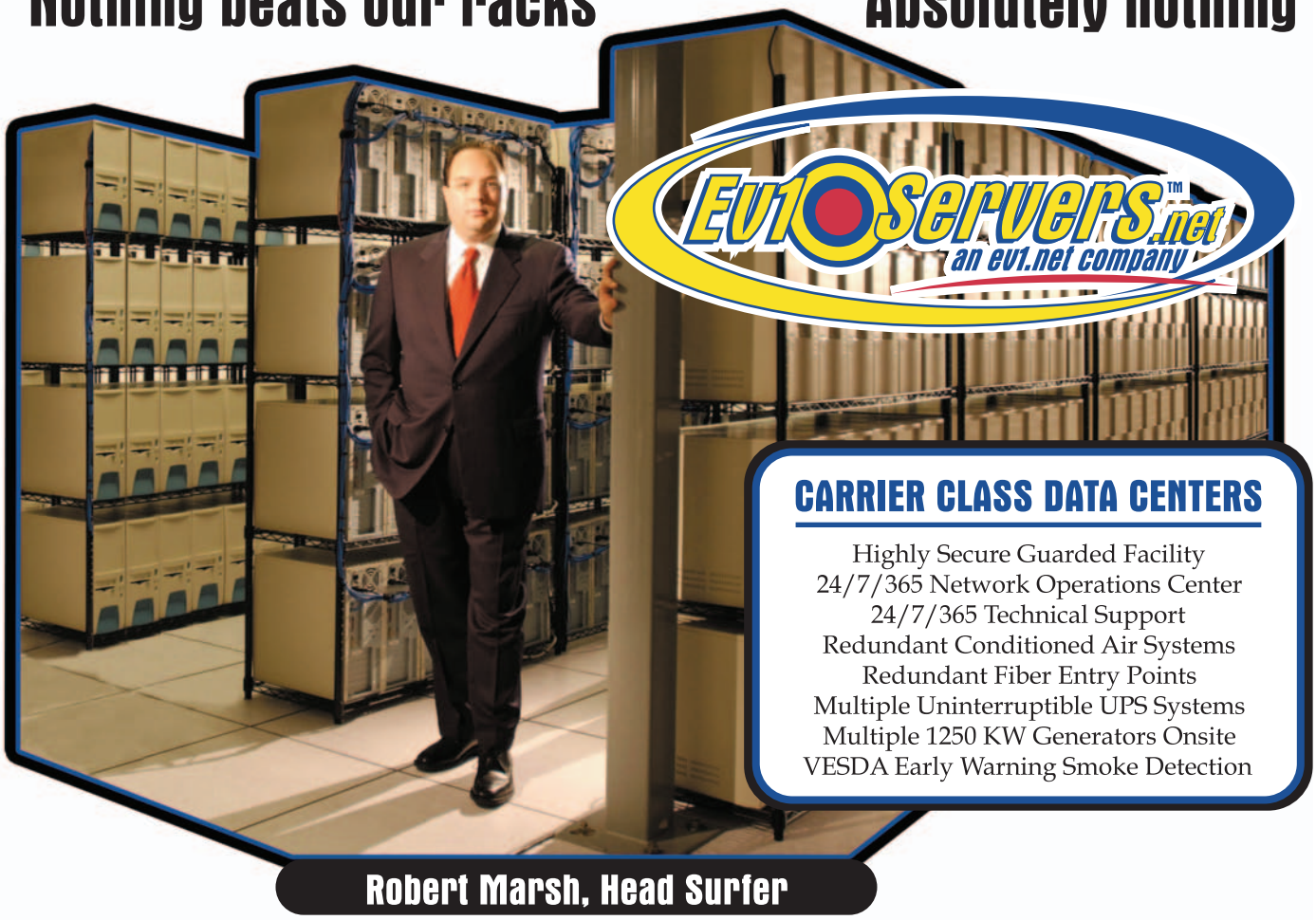
FIG 6: ACTIVITY SERVICE

Other Entity Bean Enhancements

Websphere Application Server v6 adds some enhancements to entity beans:

1. ***Dynamic query support:*** The current EJB 2.1 specification only allows you to define EJB QL statically in the EJB deployment descriptor. However, there are times when an application needs to construct a query dynamically. The dynamic query service provides an API you can use to pass EJB QL at runtime. The dynamic query service is automatically deployed on WAS as a stateless session bean with a local and remote interface. Dynamic queries is being worked on as part of EJB 3.0.
   For more information on dynamic query, see: http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/welc6tech_que.html.
2. ***Extensions to EJB QL:*** WAS version 6 provides several extensions to EJB QL to meet real application requirements. Some examples include: querying business bean methods, dependent values, and multiple element selects.

Nothing beats our racks Absolutely nothing



CARRIER CLASS DATA CENTERS

Highly Secure Guarded Facility
24/7/365 Network Operations Center
24/7/365 Technical Support
Redundant Conditioned Air Systems
Redundant Fiber Entry Points
Multiple Uninterruptible UPS Systems
Multiple 1250 KW Generators Onsite
VESDA Early Warning Smoke Detection

START YOUR OWN WEB HOSTING BUSINESS TODAY!

from **$299*** Instant Activation!

Dedicated Server

Dual Xeon 2.4 GHz
2 GB RAM • 2 x 73 GB SCSI HD
Remote Console • Remote Reboot
2000 GB Monthly Transfer Included

Over 20,000 Servers!

1-800-504-SURF | ev1servers.net

PLESK7 RELOADED
Preferred Control Panel

IP Compliant. Price subject to change. Quantities Limited.
*Per month. Set-Up fees apply. See web site for complete details.

3. ***Application profiles:*** WAS uses the notion of access intents on entity beans to define things such as isolation level and lazy loading of collection-based data. In WAS v5, an entity bean method could only be assigned one access intent. If an entity bean needed to be accessed by different session beans that had different access intent requirements, you needed to define multiple methods that access the same data or sometimes even create new entity beans mapped to the same data with different access intents. Application profiles allow me to create multiple access intents on the same entity bean method and mark the calling session beans with a particular task. Based on the container task of the session bean, the entity bean will use the correct access intent for the particular use case. This allows entity beans to be more reusable across various data access requirements.

For more information on application profiles, see: http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/welc6tech_appprof.html.

Other PMEs

We will briefly review some more WAS 6 PMEs and provide links where you can find more information.

WorkArea Service

The J2EE specification defines a component style of development. Components share data by passing information through well defined interfaces. Certain information, however, gets passed automatically. For example, transactional components such as Enterprise JavaBean components pass a transactional context to each other. Another example is a servlet that can propagate a security context to a stateless session bean. Developers do not have to pass transactional or security handles from one component to another, even when calling remote session beans. J2EE application servers pass this information along transparently. However, most other data traditionally needs to be passed through interfaces. Certain data, however, could benefit from automatic propagation. Usually any data that is needed by many components (for example, profile data about a user) is a good candidate for automatic propagation. Rather than populating every interface with the same information, it would be nice if you could propagate this frequently used information using some J2EE context.

WAS version 6 provides this function through the WorkArea service. By using a simple API, components can pass context information from component to component without passing it directly through programmatic interface. The WorkArea service will propagate to local components using thread local storage and it will pass data to remote components through IIOP. Work areas can be nested as well to allow flexibility of layering your code correctly.

A client component can create a UserWork area:

```
//Lookup UserWorkArea
UserWorkArea workArea =  (UserWorkArea) ctx.
lookup("java:comp/websphere/UserWorkArea");

//create workArea
workArea.begin("Profile");
workArea.set("Name","Luke Skywalker");
workArea.set("Age",new Integer(28));

//call component
myEJB.callService(params);

//complete workArea
workArea.complete();
```

The target component can then access the UserWorkarea to obtain the information.

```
void callService() throws SomeException
{
InitialContext ctx = new InitialContext();
  //lookup WorkArea
UserWorkArea workArea =
(UserWorkArea) ctx.lookup("java:comp/websphere/
UserWorkArea");

  //Access Data
System.out.println("Work Area");
  System.out.println("Name : " + workArea.
get("Name"));
  System.out.println("Age : " + workArea.
get("Age"));
}
```

IBM is leading JSR 149 for adding WorkArea support to the J2EE specification.

You can find more information about the WorkArea service at: http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_wa.html.

Internationalization Service

Many applications require Internationalization(i18n) support. In most cases, developers can rely on using resource bundles to display data in different languages or regions. However, some i18n requirements are more than just displaying data; applications may need to alter business logic depending on the locale of the user. For example, based on what country you logged in on, an application may need to provide different tax rules. In order to be able to do this, an application would need to get some information about the locale or time zone of the user to properly handle the logic correctly. One could use the WorkArea service to pass i18n, however, having i18n data passed in a special way can benefit greatly. Consider another case where a company needs to develop an internationalized application that is meant to be deployed in various countries. Depending in which country the

application is deployed, applications require certain portions of the application to run logic differently. Rather than maintaining different code bases, the i18n service can also be used to determine the locale of the server. This allows applications to deal with both their local i18n context and the caller's i18n context separately. The i18n service allows applications to access both the caller's i18n information as well as the local servers transparently.

The i18n service in WAS 6 allows applications to access the i18n context of the caller:

```
UserInternationalization i18nService = (UserInternat
ionalization) ctx.lookup("java:comp/websphere/UserInt
ernationalization");

Internationalization callerI18n = i18nService.get-
CallerInternationalization();
```

The i18n service also allows applications to access the i18n context of the running environment:

```
UserInternationalization i18nService =
(UserInternationalization) ctx.lookup(
"java:comp/websphere/UserInternationalization");

InvocationInternationalization invI18n =
i18nService.getInvocationInternationalization();
```

IBM is working on standardizing the i18n service through JSR 150.

More information can be found at: http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_in.html.

Object Pools

J2EE application servers provide pooling for Enterprise JavaBean components as well as resources such as JDBC connections. However, there are times when applications need to pool plain Java objects. Many times developers spend time developing their own object pools or invest in some third party library. WAS provides an object pool that can pool any arbitrary set of objects. The object pool can be configured through the WAS administrative console or through any supported WAS administrative client.

More information about object pools can be found at: http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_objp.html.

Startup Beans

Most applications require some sort of initialization. For example, an application may need to cache some read only data for fast access. This data can be initialized at start up. In the Web container, this can be done by marking a servlet for start up in the web.xml file. However, in the EJB container, start up logic can be challenging. The J2EE specification does not define a way to handle start up logic in the EJB container. WAS provides startup beans that can be used to accomplish start up logic in the EJB container.

More information can be found about startup beans at: http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_sub.html.

Dynamic Cache

The dynamic cache service is a whole caching infrastructure. It is a large topic and requires its own paper. My book (referenced at the end of this article) has a whole chapter on this topic. In short, the dynamic cache service improves performance by caching the output of servlets, commands, Web services, and JSP files. Dynamic caching features include replication of cache entries, cache disk offload, Edge-Side Include caching, Web services, and external caching.

More information on dynamic cache service can be found at: http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_dyn.html.

Conclusion

This article provided an overview of many of the WAS programming model extensions. There are certainly other extensions not addressed in this article. For example, there are more extensions in the area of SOA, such as the Web Service Gateway or Service Data Objects (SDO). This article tries to provide the PMEs that were previously only available in the Enterprise Edition (WBI SF) of WAS. Now, all WebSphere Application Server customers can use these enterprise class PMEs to fulfill their enterprise requirements.

Acknowledgments

Thanks to Billy Newport, Chris Johnson, and Bruce Clay for reviewing this document.

References

- *IBM WebSphere: Deployment and Advanced Configuration* (by Roland Barcia, Tom Alcott, Bill Hines, and Keys Botzum) www.amazon.com/exec/obidos/tg/detail/-/0131468626/qid=1106108458/sr=1-1/ref=sr_1_1/103-0795483-2226208?v=glance&s=books
- *Migrating WebLogic Startup Code to WebSphere Application Server* www-106.ibm.com/developerworks/websphere/library/techarticles/0401_beaton/0401_beaton.html
- *Specifications: service data objects, work managers, and timers* www-128.ibm.com/developerworks/java/library/j-commonj-sdowmt/index.html
- *Introduction to service sata objects:* www-128.ibm.com/developerworks/java/library/j-sdo/index.html
- *Static and dynamic caching in WebSphere Application Server:* www-128.ibm.com/developerworks/websphere/techjournal/0405_hines/0405_hines.html

Automated Deployment of Enterprise Application Updates

Part 2 – Run randomly built collections

BY BARRY SEARLE &
ELLEN MATHESON McKAY

Barry Searle is the architect for WebSphere Tools for Automated Build and Deployment. A professional engineer, he has worked at the IBM Canada Lab for over 15 years on various application development tools. Prior to that he worked on developing command and control systems, and leading complex communications development projects.
searle@ca.ibm.com

This two-part article discusses application deployment, particularly automated updates, to IBM WebSphere Application Server in a large-scale enterprise environment. It applies to Application Server versions 5.0, 5.1, and 6.0, and also includes an introduction to a few version 6.0 enhancements. This article is not intended to be used as a reference for all the details of Application Server administration, but it does describe the key concepts used and contains a list of references. Although the beginning of the article reviews some fairly basic base server and managed server concepts and operations, much of the remainder of the article will discuss certain complex concepts or operational considerations that will be new even to very experienced enterprise application server administrators.

Part one of this article discussed wsadmin deployment to base and managed servers. It examined why phased deployments are needed to maintain applications in an Application Server Network-Deployment managed cell, and how to maintain high availability in such an environment.

This article discusses pre- and post-deployment validation as well as gradual deployment of incompatible versions. It also discusses the design and implementation of a downloadable Automated Deployment example program that illustrates how to automate the deployment of randomly built collections of enterprise applications or updates, and how to automatically target those applications or updates to the correct servers, including stage-specific application setup.

Pre- and Post-Validation to Maximize Availability

As in all of the deployment scenarios mentioned in Part 1 of this article, it makes no sense to start deploying an application update if it contains problems that are likely to cause the deployment to fail.

Typical pre-validations can include validating that (see Figure 1):

- The application (EAR) appears to be complete and well formed.
- The deployment targets and the required application settings are known.
- The deployment target nodes and servers, or clusters, are valid.
- The deployment target nodes and servers, or clusters, are running and are accessible.
- The application settings are valid.
- The application is already installed, in the case of an application update.
- The application is not already installed, in the case of an application installation.
- Dependencies (prerequisites, resources, or interdependent application versions) are met.

Typical post-validations can include validating that (see Figure 2):

- The application installed correctly.
- The application started.
- The application appears to be the correct application, and is running correctly.

– One or more Web page HTTP requests return the expected HTTP response.

Note: This is not a complete function test of the application, just validation that it is running.

Gradual Deployment (Incompatible Version Migration) Using Versioned Cells

Sometimes a set of interrelated application updates has significant interrelated changes, and the rollout of all those updates must be done at the same time. Of course, enterprise application availability must still be maintained. There is a similar version incompatibility problem if the application update changes the user experience, uses different database schemas, or uses different HTTP session-persistent data. All of these major updates typically require a gradual and carefully controlled rollout of the new version. Many large organizations (financial institutions, hospitals, etc.) require that all updates be handled this way.

An incompatible version update is typically accomplished using two sets of independent production cells. The current production cell is running version=N of all applications, and handling all user requests. Another independent cell then receives deployment updates for version=N+1 of all applications, but it is not active. Network Dispatchers (IP sprayers) provide session affinity, which means that multiple request-response HTTP messages from the same user will be routed to the same Web server and application server for processing. Thus the old version=N cell can be quiesced (allowed to continue processing ongoing HTTP sessions), but the Network Dispatchers (IP sprayers) route new requests to the new version=N+1 cell.

The transition to the new version=N cell can be done either slowly, by gradually increasing the work management load on the new cell, or relatively quickly. Some organizations will initially route selected risk-tolerant users (from selected source HTTP addresses) to the new version=N+1 cell, and only after a successful operation will they reroute the balance of incoming requests. If there is a problem, the work can be routed back to the old cell. Once the transition to the new version=N+1 cell is complete, the old cell is then typically updated to the same version=N+1, or it is used to start a new version=N+2 rollout.

There are many variations on the preceding approach. Some organizations run the parallel cells on separate machines. Others install parallel (but independent) nodes on the same machines. Some organizations just use a single cell with redundant (but independent) clusters, and do a gradual deployment one cluster at a time, using the Web HTTP server (configured with session affinity) to carefully quiesce and reactivate workflow to each of the clusters.

Of course, nearly every organization first extensively tests the new updates in a pilot-production stage, including functional testing and performance/stress testing, before the production rollout is even started.

For an excellent case study of how the New York Stock Exchange (NYSE) and the Securities Industries Automation Corporation (SIAC) achieve extreme availability using gradual rollouts, see the article "Extreme Availability with WebSphere and DB2" in the reference section.

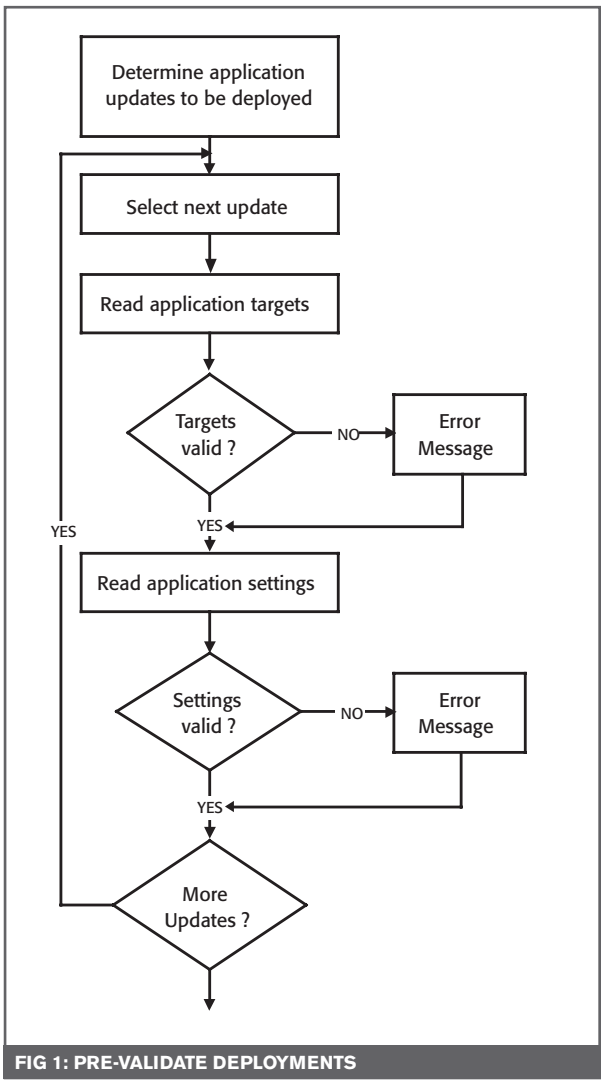


FIG 1: PRE-VALIDATE DEPLOYMENTS

Automating Enterprise Application Update Deployments

The downloadable Automated Deployment example program shows a way to simplify the automated deployment of enterprise updates while helping to maximize enterprise application availability. The Automated Deployment example is designed and implemented to handle the typical enterprise concept of deployment Stages (see Figure 3).

The Automated Deployment example may be invoked manually or by some regularly scheduled system program (see Figure 4). The invocation must specify the input Distribution Directory, the deployment Action to perform, and the deployment Stage qualifier:

The stage name is just a text qualifier to the application Target and Settings files, and may represent quality stages (integration, pilot, production, etc.), platforms

Ellen Matheson McKay is an information developer for IBM Canada Ltd. She writes online help and publications for Rational Application Developer.
ecmckay@ca.ibm.com

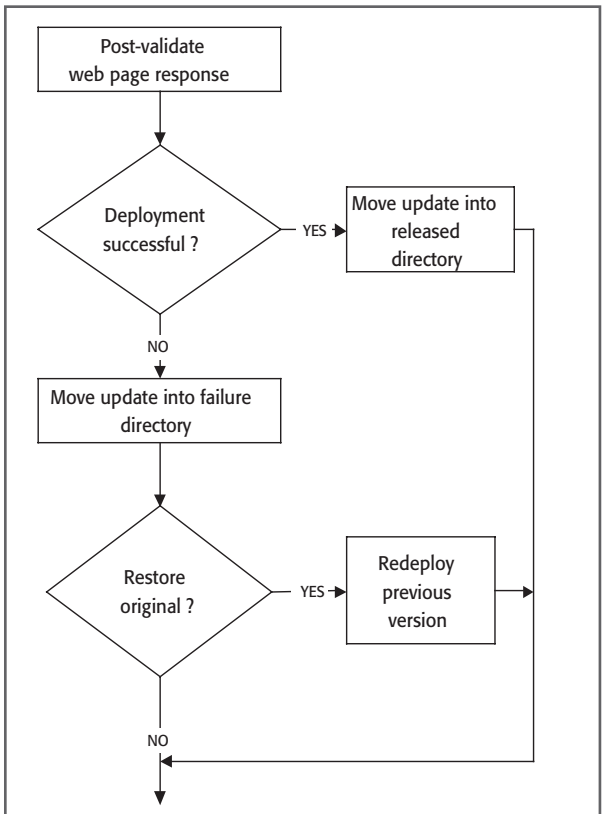


FIG 2: POST-VALIDATE DEPLOYMENT

(Windows, Linux, z/OS, etc.), organization departments (sales, finance, etc.), or whatever names or combinations a particular enterprise uses.

If an updated application myApp.ear is to be deployed into the pilot stage, then a typical MyApp-pilot.targets server targets property file (or XML file or database entry, etc.) might be:

```
# multiple entries: nodeserver=nodeName,serverName
# multiple entries: cluster=clusterName
nodeserver = myNode,myServer1
cluster = myCluster

# multiple entries: testURL = URL
# multiple entries: testResult = resultString
testURL = http://myHost:9081/myAppWeb/AdderTest.jsp
testResponse = Java adder(3,4) = 7
testURL = http://
myHost:9085/myAppWeb/
AdderTest.jsp
testResponse =
EJBsessionbean
adder(5,6) = 11
```

Similarly, a typical MyApp-pilot.settings application settings property file might be:

```
# myApp application EAR
ApplicationName=myApp
startingWeight=9
warClassLoaderPolicy=SINGLE

# myApp webmodule WAR
ModuleName=myAppWeb.war
startingWeight=12121
classloaderMode=PARENT_LAST
```

The overall automated deployment sequence is (see Figure 5):

1. Invoke the automated distribution program
   a. This is typically done from an automated system cron job, but may be manually invoked
   b. The invocation command also specifies the stage to be deployed to
2. Read the distribution directory to determine the new application updates to be deployed
3. For each application update, read its stage-specific server targets and application settings
4. From the total set of affected nodes and servers, calculate the subset of unique affected nodes and unique affected servers
5. Pre-validate that the applications and targets and settings are valid
6. Save and then disable AutoSync on all affected nodes
   a. Optionally, you can save and disable SyncOnStartup
7. Install the applications into the Deployment Manager repository:
   a. Set the stage-specific application settings
   b. Set the stage-specific target servers or clusters
8. Sequentially, for each affected node, phase distribute the updates:
   a. Optionally, quiesce all its affected servers (reroute new work requests)
   b. Stop all its affected servers
   c. NodeSync that node to retrieve all updates and install them into the affected servers
      Note: Wait to ensure the EAR expansion is complete
   d. Restart the affected servers
      Note: Test and wait to ensure the server is running
   e. Optionally reactivate the affected servers (to process new work requests)
   f. Optionally validate the installed application operation and request manual confirmation

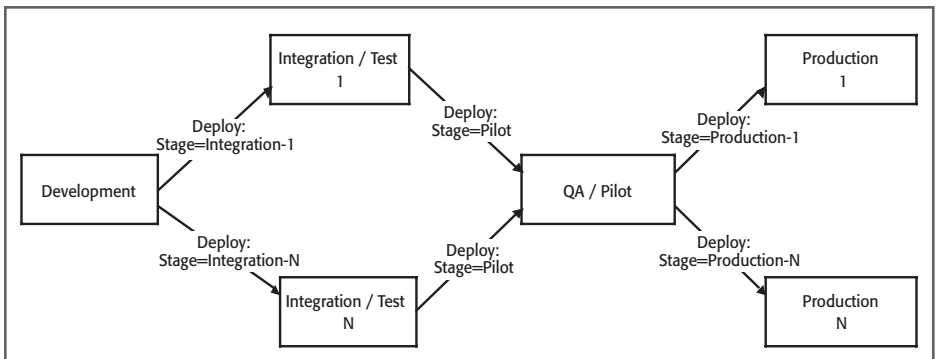


FIG 3: TYPICAL STAGED DEPLOYMENT

NEW!

CodePro AnalytiX™

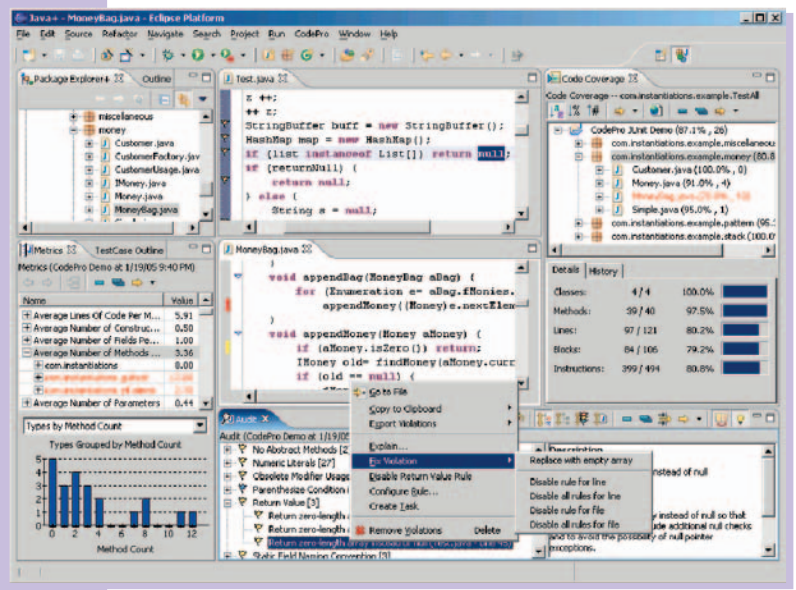
Continuous Collaborative Code Analysis for Rational®, WebSphere® & Eclipse

Imagine your software team creating higher quality, more reliable, Java systems... with less time and money.

Sound like a fantasy?

...*CodePro AnalytiX* lets you do exactly that, or

Your money back.



***AnalytiX* automates the process of building quality into your software as it's developed. Using the latest software development technology *AnalytiX* monitors Java code development in real-time, as its written. It identifies and automatically fixes hundreds of issues that would reduce quality and reliability.**

Powerful wizards generate unit and regression tests and analyze code coverage, helping drive out problems early in the development process.

Less experienced developers code like experts, experts create tight, maintainable systems.

***AnalytiX* also allows you to define, distribute, and enforce corporate coding standards and quality measures across your development teams...no matter where they're located.**

Early identification and correction of code quality issues dramatically improves your application development process and reduces the time and money it takes to develop your high performance Java systems.

From the development tool experts at Instantiations. No one outside of IBM has longer or deeper experience in creating Eclipse-based Java development tools.

Put our experience to work for your team.

www.instantiations.com

1-800-808-3737

Key Product Capabilities

- **Detect and correct code quality issues ...*Automatically***
- **Define, distribute and enforce code quality standards across development teams**
- **700+ rules and metrics/300+ Quick Fixes**
- **Code metrics with drilldown and triggers**
- **Custom rules and rule sets**
- **JUnit test case generation**
- **Code coverage analysis**
- **Dependency analysis and reporting**
- **Integrated team collaboration**
- **Javadoc analysis and repair**
- **Seamless integration with IBM Rational Developer, WebSphere Studio and Eclipse**

Risk Free Trial Copy

www.instantiations.com/analytix

eclipse READY · IBM Business Partner · eclipse FOUNDATION MEMBER

instantiations

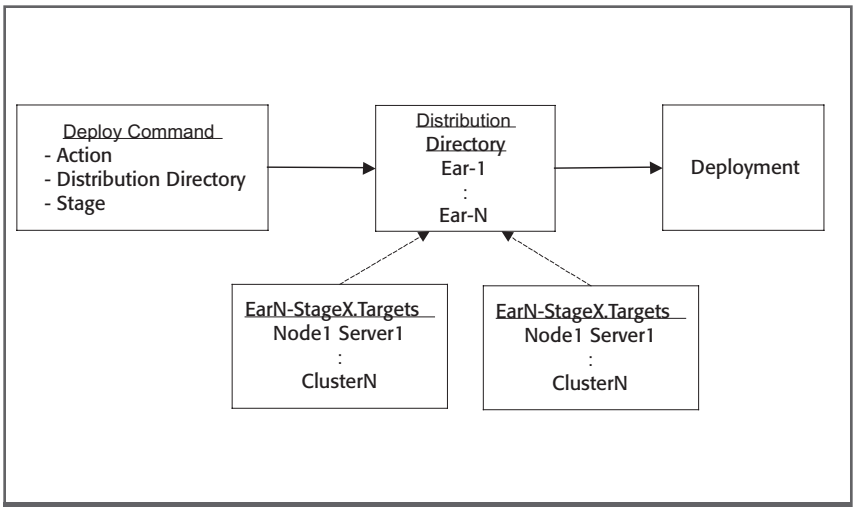


FIG 4: AUTOMATED DEPLOYMENT INPUTS

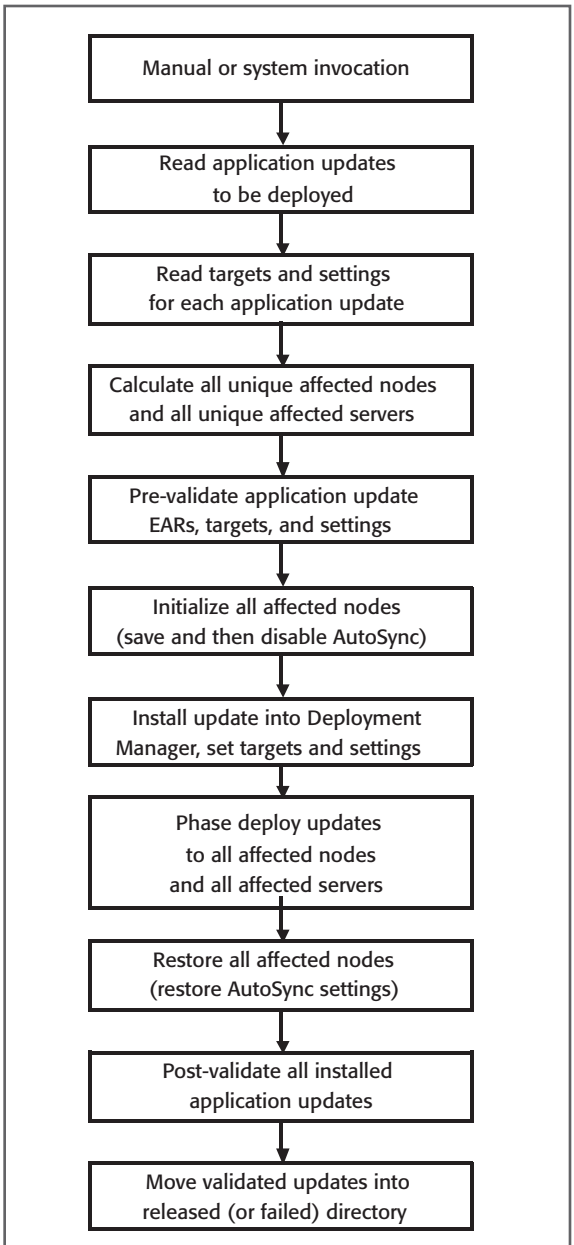


FIG 5: AUTOMATED ENTERPRISE UPDATE DEPLOYMENT

9. Restore the previous AutoSync settings for all affected nodes
   a. including SyncOnStartup if it was optionally disabled
10. Post-validate all applications
11. Move validated applications into the released directory
    a. If it failed, you can attempt to restore the previously released application
12. Optionally e-mail the deployment log to a notification list

As before, there are two special notes in the above steps. First, after performing the NodeSync, the application update (EAR) has been distributed down to the node, but the EAR file must still be expanded into the server installed application directory. Until this EAR expansion is complete, attempting to start the server may produce indeterminate results. There is an IBM Problem Report about this and in the future there may be a downloadable Application Server Interim Fix to allow scripts to test for the completion of the EAR expansion. Second, after returning from the wsadmin startServer command, the command has been processed by the Node Agent, but the actual server startup may not yet be complete. Scripts need to test that the server has completed startup and is running.

Review of Requirements and Benefits of Automated Update Deployments

Automated update deployments are intended to:

- Automatically detect current updates to be deployed
- Automatically read stage-specific targets and application settings
- Optionally, read application requirements (prerequisites, resources, interdependent versions, etc.)
- Minimize failures by pre-validating as much as possible
- Only update affected nodes and affected servers
- Only cycle affected nodes and servers once each (even in the case of many concurrent updates)
- Only cycle one affected node at a time to maximize application availability:
  - Optionally, quiesce affected servers (to allow complex in-progress HTTP sessions to complete)
  - Stop affected servers (to allow server failover and session recovery)
  - NodeSync to retrieve all updates and install them into the affected servers
  - Restart all affected servers
  - Optionally, reactivate affected server if quiesced

– Minimize the effect of failures by post-validating

Deployment Actions

The primary focus of this article has been the deployment of application updates since that is by far the most complex deployment problem. Generically, the most common deployment actions are:

1. Install (a new, nonexistent application)
2. Update (a rebuilt application, or rebuilt application components)
3. Reconfigure (change application settings or targets)
4. Uninstall (remove an existing application)

Automated Deployment Example Program Updates and Limitation

It is expected that any updates to the current version 1.1 (December 2004) Automated Deployment example program may be available from the Application Server "Sample Scripts" Web page in the Developer Domain Application Server Library Samples (www.ibm.com/developerworks/websphere/library/samples/SampleScripts.html).

Some limitations in the current example program include:

1. Limited error detection and exception handling (not production quality code)
2. No exploitation of Application Server security (no userid nor password passed to wsadmin)
3. No handling of nested or complex attributes (for application settings)
4. Errors and Warnings are logged and summarized, but no e-mail notification is generated
5. No handling of incompatible applications or versions (no interdependency analysis)
   a. Today, many customers handle this by having version=N and version=N+1 cells

References

WebSphere Application Server Administration

- *IBM WebSphere Application Server Network Deployment V5.1: System Administration:* ftp://ftp.software.ibm.com/software/webserver/appserv/library/wasv51nd_admin.pdf *Ch-4: Welcome to Scripting, Ch-5: Deploying and Managing Using Scripting*
- Williamson, Leigh; Chan, Lavena; Cundiff, Roger; (et al.) (2004). ISBN-0131446045. *IBM WebSphere System Administration*
- Barcia, Roland; Hines, Bill; Alcott,Tom; Botzum, Keys (2004). ISBN-0131468626. *IBM WebSphere: Deployment and Advanced Configuration*

WSADMIN Scripting

- *WebSphere Application Server Information Center: Deploying and managing using scripting:* http://publib.boulder.ibm.com/infocenter/wasinfo/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm.html
- *WebSphere Application Server Technical Library, Sample scripts:* www.ibm.com/developerworks/websphere/library/samples/SampleScripts.html
- *RedBook: WebSphere Version 5 for z/OS – WSADMIN Primer:* www.ibm.com/support/docview.wss?uid=tss1wp100421

WebSphere Availability and Workload Management

- *Extreme Availability with WebSphere and DB2:* www.ibm.com/websphere/developer/zones/hipods/
- *High Performance On Demand Solutions:* www-128.ibm.com/developerworks/websphere/zones/hvws/library.html
- *RedBook: WebSphere V5 Performance, Scalability, and High Availability*: www.redbooks.ibm.com/abstracts/sg246198.html
- *RedPaper: Server Clusters For High Availability in WebSphere Application Server Network Deployment V5:* www.ibm.com/support/docview.wss?uid=swg27002473
- *Maintain continuous availability while updating WebSphere Application Server enterprise applications:* www.ibm.com/developerworks/websphere/techjournal/0412_vansickel/0412_vansickel.html
- *Using WebSphere Application Server V5 for Load balancing and Failover:* www.ibm.com/developerworks/ibm/library/i-wasldbal/index.html

A detailed list of resources pertaining to this article can be found online at http://sys-con.com/websphere.

Integration and standardization

A Step Toward Virtual Portals

BY THOMAS STOBER

Dr. Thomas Stober joined IBM's Pervasive Computing Division in 1998. During his IBM career he worked on smart cards, data synchronization, and mobile computing. Today, Thomas is based in the IBM development lab in Germany. As a member of the Portal's architecture team, he is responsible for virtual portals.
tstober@de.ibm.com

WebSphere Portal is all about the integration of users and administration, a common look and feel, and a standardized application programming model. With Portal, various back-end systems can be integrated to a common user experience. However, there are various cases where you will have several independent user populations and will want to provide a unique and distinguished look and feel to each of them. Each community will want to work with its own logical partition of WebSxphere Portal independent of all others.

Examining Some Typical Usage Scenarios

Before you start to set up a portal server with logical partitions for different communities, it is important to be clear about your specific needs and priorities. This section details three typical usage scenarios and their typical requirements.

Multiportal Enterprise

In the first scenario, a single enterprise owns and operates multiple and different logical portals on a single installation. For instance, there will be independent portal instances for different organizational units or brands. Usually there will be a strong emphasis on the sharing of applications, centralized administration, and a common user repository.

Workgroup Service Provider

The second scenario is very similar to the previous one, but there are significantly higher numbers of such portal instances. There can be dedicated portal services for individual teams, departments, or projects. Easy creation of additional portal instances as well as self-administration is essential to achieving a reasonable total cost of ownership (TCO). Prerequisites for this scenario are minimal memory footprint per portal instance and good scalability.

Hosted Enterprises

In this scenario a service provider hosts and operates independent enterprises on the same portal installation. A specific portal instance can be provided for each tenant. A major focus will be on self-administration by the tenant. Secure isolation between portal instances is essential. Each instance typically has its own user repository. Applications as well as data must be shielded properly between tenants. Quality of service (QoS), reliability, availability, and failure protection will be other strong requirements.

The following questions and considerations will point out that some of the key requirements in the described scenarios are orthogonal to each other.

- Is it most important that the portlets and applications of different logical portals are shielded from each other? Do you need a strong isolation of your applications? This can be achieved by having a separate Java Virtual Machine (JVM) for each portal instance. Nevertheless, the downside of this is the significant memory footprint of running multiple JVMs in parallel.
- Do you want to share data? Or is it mandatory to have a very strict isolation of your data and portal configuration? Based on this decision, either a separate or a common portal database may be required.
- Are you considering a few or hundreds of portal instances on a single installation? The answer will determine if you need to stick to a single JVM in order to minimize memory consumption.

There are two alternatives to implementing multiple portal server instances on a single hardware instance (see Figure 1). Based on the previous scenarios and questions, you need to decide which of these two options fits best for your situation.

- True portals are full parallel portal and application server installations on the same hardware instance. Each of the portal servers will run its own JVM and have its own portal configuration database. The benefit of this option is the best possible isolation of applications and data. The downside is that it is not possible to share applications between true portals.

"Virtual portals are peers of each other"

Using multiple JVMs will also have an impact on memory consumption and limit the number of true portals that can reside on the same hardware.

- Virtual portals are logical partitions within a single portal and application installation. Many such partitions are possible since this approach is highly scalable. Sharing of applications and data is possible. Concepts for isolation between virtual portals are introduced. Nevertheless, failure protection between applications is limited, due to a common JVM, which is shared among all virtual portals.

Virtual portals have been introduced as a new feature in WebSphere Portal 5.1. The following sections will explain the concepts of virtual portals in more detail.

New in WebSphere Portal 5.1: Virtual Portals

Virtual portals can be created easily within a single Websphere Portal 5.1 installation whenever you need additional logical portal instances. You do not need to repeat the portal or application server installation over and over again. Instead, you will reuse the existing Portal server on your system and avoid redundant memory footprints.

Virtual portals are based on the following key concepts:

- Can be customized to expose their unique look and feel
- Are accessed through friendly URL mappings
- The notion of scoped and non-scoped portal resources is introduced
- Each virtual portal can have its own distinct user population
- For administration of virtual portals, the delegation model of Portal Access Control is leveraged
- A new administration portlet will expose a user interface to manage virtual portals

Individual Look and Feel

Virtual portals are peers of each other. Each virtual portal has its own navigation with its own context root (represented by a label), its own instances of administration portlets, and its own set of pages (see Figure 2). These pages include a login page, favorites, My Portal and the anonymous pages. You can assign a particular theme to the pages of a virtual portal in order to expose a unique look and feel to the users.

In Portal 5.1, it is now possible to use Login and Enrollment/Self-care Portlets instead of the screens that have been used in the previous releases. These Portlets and their pages can be customized for each virtual portal.

Accessing a Virtual Portal

A user can access a virtual portal using a URL mapping, which is defined during the creation of a virtual portal. The mapping points to the root page of the virtual portal that the user wants to access. Such a friendly URL for a virtual portal can look like this: www.ibm.com:9081/wps/portal/aVirtualPortal.

Internally, the URL mapping will be translated into a representation, which uses the object identifier of the virtual portal as a parameter in the URL. For instance, such an internal URL can look like this (18_0_35 is the virtual portal's object ID): http://www.ibm.com:9081/wps/portal/!ut/p/vp/18_0_35.

Based on the used URL, the Portal will determine which virtual portal a user wants to access. The requested virtual portal will be selected as the current context. All of the user's activities will be executed within in the context of that virtual portal.

Scoping and Sharing

The requirement of isolating the content of individual virtual portals, but still allowing the sharing of applications, has been addressed in Portal 5.1 by a new concept: depending on the type of portal resource, the resource will be either be scoped to a particular

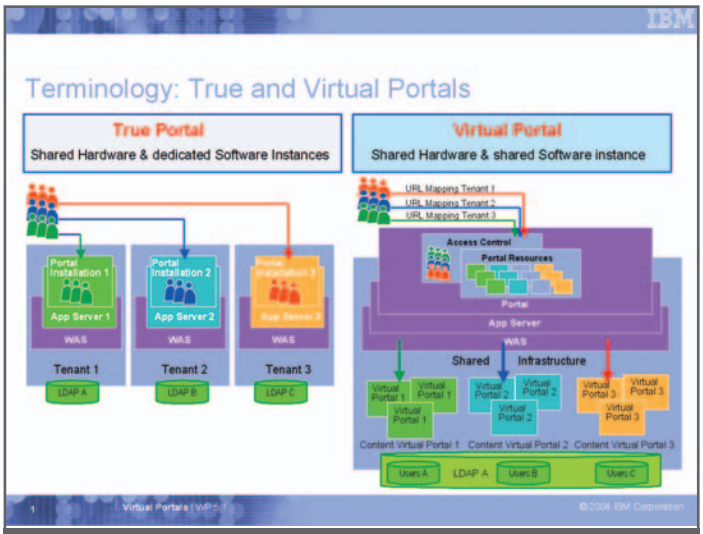


FIGURE 1: TRUE PORTALS VERSUS VIRTUAL PORTALS

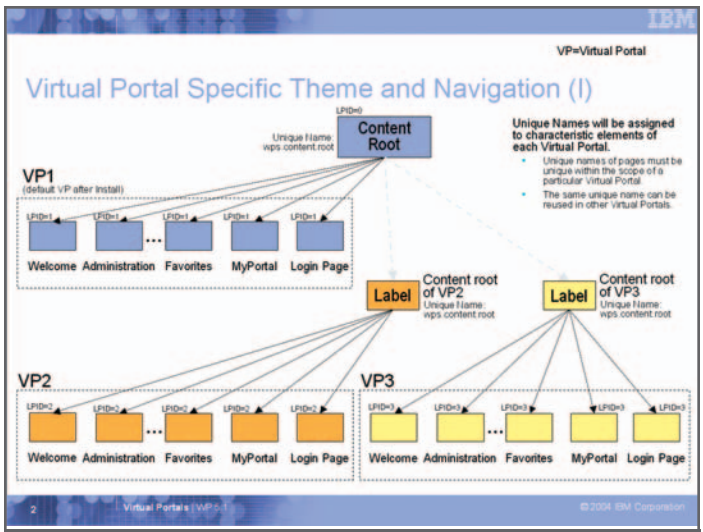


FIGURE 2: VIRTUAL PORTAL–RELATED NAVIGATION

"Each community will want to work with its own logical partition of WebSphere Portal independent of all others"

virtual portal, or shared among all virtual portals of an installation.

Scoped Resources

Scoped resources belong to exactly one virtual portal. They are associated with the virtual portal in which they have been created. To use such a scoped resource, you must access the associated virtual portal. These resources are not visible or accessible from any other virtual portal and, therefore, cannot be shared among virtual portals.

Pages, portlet entities, and search indexes are scoped resource types. The scoping of these resource types is enforced by the portal. You cannot modify or override this behavior, e.g., using access control.

Shared Resources

Shared resources are available in the entire portal installation and can therefore be used by all virtual portals. Portlet definitions, portlet applications, Web modules, URL mappings, themes, and skins are shared resource types. Nevertheless, it is possible to apply portal access control and limit the usage of these shared resources to certain virtual portals. This is done by granting permissions to a shared resource only to the users of the virtual portal of choice.

Unique Names

Unique names can be either scoped or shared, depending on the resource to which they are assigned:

- Unique names of scoped resources are scoped
- Unique names of shared resources are shared

This implies that only the unique names of shared resources must be unique within the portal installation. When used for a scoped resource, the same unique names can be reused in multiple virtual portals.

Portlet Definitions and Portlet Entities

When you try to share portlets across virtual portals, one important impact of scoping and sharing surfaces: since portlet applications are shared resources, their properties will be common across the entire installation. This means that the configuration preferences of the portlet definition will have the same values for all virtual portals. If you need to define different configuration preferences for your virtual portals, you will have to clone the portlet application. Each virtual portal can use its own clone with its own individual configuration preferences.

On the other hand, the portlet entity is scoped. Any preference of portlet entities, which you have defined, will be specific to a single virtual portal. Obviously, those properties cannot be shared across virtual portals.

Installation-Wide Settings

All portal property files apply to the entire installation. You cannot define a specific setting for individual virtual portals. The same is true for portlet services and the credential vault. In the credential vault, any public credential will be visible to all virtual portals.

Defining a User Population for a Specific Virtual Portal

In Portal 5.1, you can configure WebSphere Member Manager (WMM) as a custom user registry. WMM introduces the notion of a realm. A realm aggregates a set of users within a user repository, for instance, by selecting several nodes in an LDAP tree. Realms are an abstraction of physical user registry systems; they expose a distributed user population as a single coherent entity. It is important to emphasize that realms are defined within the WMM configuration and are not part of the user registry itself. This distinguishes realms from groups.

Realms can be applied to scope users to specific virtual portals; you can associate a virtual portal with a specific realm. The realm membership is validated during authentication in order to ensure that a virtual portal can only be accessed by members of the corresponding realm. Therefore, the realm membership can be used to have a strictly separated user population for different virtual portals. Even

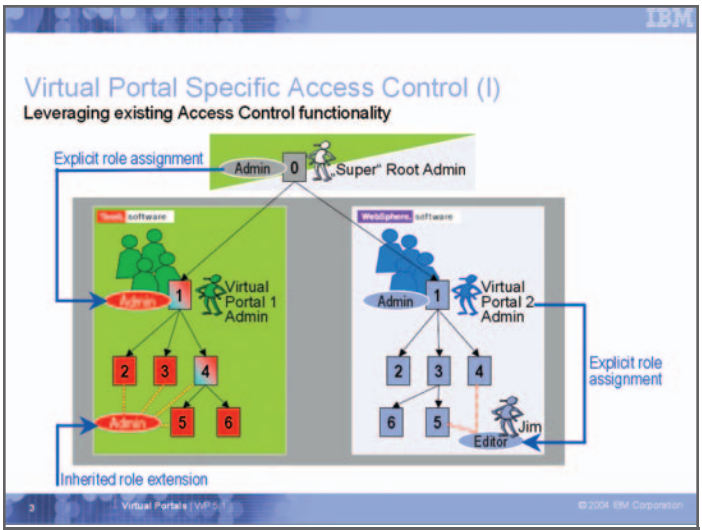


FIGURE 3: LEVERAGING DELEGATED ADMINISTRATION PROVIDED BY THE PORTAL ACCESS CONTROL

Ignorance isn't bliss.

Knowing about performance problems before they happen is.

It might seem like bliss to ignore application performance problems and deal with them only if and when they surface. But waiting until a problem arises in production only results in crises, stress, and wasted time. Experience real bliss by eliminating performance problems proactively with DiagnoSys.

From development through production, DiagnoSys targets potential problems by providing a wide breadth of information, collecting data not only from the application server but also the associated resources, such as the Web server, database, and network.

This breadth of data gives a more accurate picture of a developing problem. Other products only look at the app server and imply performance of the resources based on the performance of the connections, a method that often misses the real source of the problem.

DiagnoSys also provides in-depth, statistically correlated information so you see a more complete view. DiagnoSys gathers information down to the method on the app server as well as in-depth information for all resources down to the OS.

Even though you get depth of information, you won't be overwhelmed. DiagnoSys synthesizes the data and quickly identifies the root cause so you can correct the problem before it impacts you or your company.

diagnoSys

Improve Web app performance at your pace. Listen to expert Web seminars – it's a fast and easy way to keep up on the latest advice, tips, and best practices.

Visit: www.hwcs.com/wsj19.asp

SILVER ANNIVERSARY 1979 - 2004

H&W

www.hwcs.com | 1.800.338.6692

Bridging the Enterprise Computing Gap For 25 Years

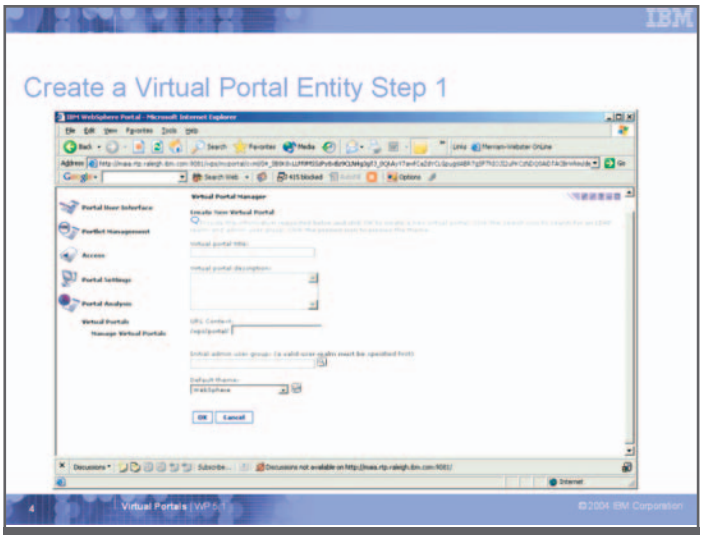


FIGURE 4: THE NEW "MANAGE VIRTUAL PORTALS" PORTLET

a wpsadmin will not be able to login into a virtual portal if he/she is not a member of the corresponding realm. This security check cannot be overwritten. Nevertheless, it is possible that multiple virtual portals share the same user population by specifying the same realm relationship. Realms can overlap, which allows a user to be a member of more than one realm.

Access Control

When setting up a virtual portal installation, you will have two different kinds of administrators:

- The master administrator
- The VP-specific subadministrator

The master administrator is responsible for the entire installation. He/she will have administration privileges for the portal. He/she will be able to create virtual portals and will typically manage the shared resources. The master administrator will also be responsible for the installation of portlets, themes, skins, etc.

During the creation of a new virtual portal, the master administrator will delegate the administration of the new virtual portal to a group of subadministrators. The subadministrators will manage the pages, portlet entities, and users within their virtual portal (see Figure 3). The flexibility of portal access control supports the concept of virtual portals very well. By inheritance, subadministrators implicitly have the administrative access rights for all the child pages underneath the context root label of their virtual portal. Access to shared resources can be granted explicitly.

The master administrator determines the delegation of access control roles to the subadministrator. By default, only the admin role to the VP's context root label and the editor role for the VP's portlet entities will be assigned automatically. Additional roles need to be set up by the master administrator. Typically, these additional role definitions will include editor and security administrator permissions for the users and groups of this virtual portal.

Some portlets, such as Web Clipping or Search, require additional role definitions for the virtual portal's subadministrator as well.

Since all virtual portals run in the same JVM, it is unlikely that subadministrators will be allowed to deploy their own code into their virtual portal, since there is a potential risk that uncertified, poorly written, or malicious code would impact the overall stability of the installation. If individual virtual portals need to integrate their private portlets, the preferred solution would be to aggregate remote portlet services using WSRP.

Regardless of role definitions in access control, it is never possible to access or manage a scoped resource in another virtual portal, nor is it possible to administer users if you are not a member of the user's realm. These principles are enforced by the portal.

Managing Virtual Portals

The "manage virtual portal" administration portlet provides the capabilities to manage virtual portals (see Figure 4). In order to create a virtual portal, a set of properties such as title, realm, default theme, and the designated group of subadministrators need to be specified. During the creation process, an XML Access script will be executed and will provision the initial content of the new virtual portal. Once out of the box, that script will deploy a set of basic administration portlets including login, self-care, search portlets, and the document management portlets. The default content can be modified easily by changing the provided XML Access script. The script can be found in the WAS directory in installedApps\<hostname>\wps.ear\wps.war\virtualportal.

Besides the "manage virtual portal" portlet, it is also possible to use a set of configuration tasks in order to manage virtual portals.

Summary

Virtual portal is an out-of-the-box feature of WebSphere Portal 5.1. The feature has a strong emphasis on an efficient implementation, which addresses the need for isolation as well as for sharing between virtual portals.

For further reading, you can find additional information in the virtual portal section of the InfoCenter at: http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/index.html.

"It is also possible to use a set of configuration tasks in order to manage virtual portals"

A WebSphere success story

Content Management at the Department of Energy

BY DAVID BEVERIDGE

David Beveridge has served in every aspect of Web development. He oversees all areas of project management and operations for Brook Group. Prior to joining Brook Group, David amassed extensive experience running Web shops for a large not-for-profit association and a dot-com startup. During 20 years at the *Boston Globe* and *National Geographic Society*, David entered the interactive age in the late '70s, cutting his teeth on early full-text search and retrieval systems for print publishing. He was named the Geographic's first full-time online staff member in 1994, running its sites on America Online and CompuServe, and served as a founding member of the National Geographic Interactive management team.
dave@brookgroup.com

When the Office of the Chief Information Officer at the U.S. Department of Energy (DOE) called for a renovation of the Department's signature Web site, energy.gov, the plans were ambitious: blow up and reorganize the site's organization and navigation, develop a new look and feel to bring the site a high level of sophistication and polish, and deploy a publishing system that DOE's content team can use to create and maintain the site easily and efficiently, at less cost.

DOE put out the call to Brook Group, a Web services firm, to rescue the Web site and asked them for a full redesign and re-architecture of the site content. As for the publishing system, Brook Group answered the call with Tacklebox, its Java-based Web content management system (CMS), running on an IBM server platform.

Web Publishing for the 1990s

In the "old" days of the Web, sites were hand-coded, page by laborious page, using now-familiar technologies such as HTML, FTP, and TCP/IP. These technologies looked then (and still look today) far too much like programming for most nontechnical people. It was unrealistic to expect people to acquire such specialized technical knowledge and training just so they could enter press releases onto the site or change a phone number on the Contact Us page. In most organizations, that meant the people closest to content and its presentation – writers, artists, product specialists, marketers, PR folks – had neither the training nor the inclination to do what was necessary to publish on what was quickly becoming the organization's most public forum – its Web site.

Worse yet, the people most likely to acquire that technical knowledge rarely possessed the background or position in the organization to make decisions about content presentation. The disconnect that resulted – what we call the Webmaster bottleneck – led to inefficiency and frustration at both ends of the equation. Content "owners" want to control how content is managed. They need the ability to make edits (often simply fixing typos or changing a few words) without waiting for their requests to cycle through the IT department's priority list of to-dos. The IT people called into service for Web site maintenance are frustrated by the mundane nature of much of their work, particularly when the content owners always seem to want the work done yesterday. Add to that the unavoidable culture clash between techies and nontechies, and it became clear that a new solution was required. Enter the CMS.

Content Management

Tacklebox and other such systems allow organizations to put content management in the hands of anyone by simply assigning them rights to manage content and requiring a minimum of training. A CMS automates many of the mundane tasks that make up the publishing cycle. Content is entered through a Web-based content creation and editing engine. The engine provides "what you see is what you get" controls, allowing nontechnical staff to maintain content in a recognizable, word processor-like interface. This means that those simple edits that need to be done "now" can be done, simply and immediately, by anyone in the organization. A workflow system allows site administrators to automate the editorial review and approval process by assigning staff members to step-by-step workflows that can be assigned to individual pieces of content. Each staff member has a specific role in the workflow. Once those workflow assignments are made, the system provides the right users access to the right content, and then routes that content through its appropriate approval path.

Workflows are created by site administrators to mimic their organization's real-life publishing flow, moving content from author to editor to reviewer until one of three outcomes is reached: the content is published on the live Web site; it is sent back to the author for changes and then iteratively through the pro-

cess until it is ready to be published; or it is not approved and is stored in the content archive as a "disapproved" item.

The system automates notifications by alerting workflow role players when to act. It provides content scheduling, an automated way for content to "go live" or be removed by simply applying a scheduled date to the content. Instead of the 1990s method of requiring Web staffers to sit at the keyboard waiting for midnight to strike so they could punch the Enter key to FTP a new page to the site, CMS system daemons constantly monitor workflows and content states, moving content onto or off of the site as scheduled. This self-monitoring allows authors to develop content well ahead of its needed publication date. The content then goes through the entire approval workflow, but is not posted live on the site until the time is right.

In addition to its main job of content management, full-feature CMS publishing provides two key site management features.

First, navigation management allows site managers to manipulate the actual site structure. Where pages and folders once used the server's file system to create a navigation tree, Tacklebox allows the DOE staff to add, delete, move, or rename sections, subsections, and subsites, or create page redirects to assist in marketing and in managing legacy URLs. Managers can hide or reveal sections of the site, allowing them to fill pages with content in the background, and only reveal them when they are fully ready for publication.

Second, all publishing activities related to a given piece of content, or a given user, are monitored and maintained in an audit table, allowing site managers to know precisely which content was updated, when and by whom, and maintaining those specific relationships so that content can be "recalled" if necessary. This function is crucial to federal government Web sites like energy.gov. Federal Web content is part of the public record of federal documents, subject to retention policies just as is any document produced by the Government Printing Office. Preservation of documents and an audit trail that preserves all published versions of a document, as well as recording the actions taken during the publication cycle are crucial features of a government CMS.

Building the Infrastructure

HTML sites were simple. They needed an Internet-connected Web server, a set of HTML files containing embedded static content, and a collection of image files that were referenced from within the HTML: that is, a Web server, a file server, and a DNS entry. While the technical infrastructure required to drive a CMS like Tacklebox is certainly within the capability of many organizations, it far exceeds that of a typical HTML site. The main Tacklebox engine is built in Java using the Struts framework. Its database back end contains all site content, graphics, page templates, navigation, and site-specific configuration options, and the front-end templates are sets of JavaServer pages (JSP).

Tacklebox requires an application server to manage interactions between the Web server and the application/database. The application is serving three masters at once: the set of users visiting the site over the public Internet and viewing HTML pages served by the system; the site managers and content folks who are working behind the scenes to dynamically change the site while it is in production; and the application functions that drive the system workflow, site management, user management, e-mail notifications, and automated publishing functions.

Brook Group knew that DOE would require a high level of reliability and industrial-strength performance from every tier of the site. For that reason, they recommended the IBM infrastructure: IBM Web Server at the front end and a WebSphere Application Server in the middle tier. DOE planned to use an existing Oracle 9i database server for the back end. Brook Group's Tacklebox development team uses WebSphere Studio Application Developer as its primary development tool. Given the DOE mandate that the site maintain maximum uptime, even under potentially heavy load, Brook Group decided that WebSphere Application Server (WAS) was the best choice. With large-scale clients such as America Online, Ameritrade, the Department of Justice, and the White House Office of Management and Budget,

"This means that those simple edits that need to be done 'now' can be done, simply and immediately, by anyone in the organization"

they know that their solutions have to be reliable.

CMS systems can be deployed in one of two modes. Clients such as DOE, who have the existing infrastructure and resources to host the application themselves, often prefer that method. It provides them with economies of scale through shared resources and ensures that all data and system management functions stay behind the firewall. Brook Group also provides a hosted CMS solution in which clients use a hosted version of the site. For that method, Brook Group also uses WAS as its application server.

"Stuff" Happens

The process of designing and building a department-level Web site for a federal agency is fraught with bureaucracy: endless meetings and conference calls, steering committees, listservs, and multiple layers of approval for every project element from color schemes to search engines to hardware acquisition plans to content types. The Office of the CIO at DOE led a 30-plus member steering committee of stakeholders throughout the organization. They shared two characteristics: they all wanted a new site, and they all disagreed on what that site should be.

The recommendation to use a WebSphere Application Server met with minor opposition. Although DOE already used the product, the specific staff members assigned to this project were unfamiliar with it. They had valid concerns about their ability to support a product they did not know well, and also were concerned that it was more power than this application and this one Web site needed. As it turned out, the recommendation paid off.

After a number of months developing design approaches and content maps, acquiring and installing equipment, gaining approvals, and conducting meetings, we finally had a working application in place. The DOE staff charged with developing content were hard at it, and the site took shape. The final element to be implemented, a non-IBM search engine, was the final technical piece to be deployed. As the launch date neared, the search engine was proving a particularly sticky issue. It was not well documented and the software company provided little technical support. A few days before final testing, the engine was finally configured and working. Final testing showed all systems working. Almost. The site, now running in a staging environment and ready for a public launch, began to bog down. Performance seemed to regularly slow to a crawl and the site began to drop. The application server was crashing and no one knew why.

When a large, public project begins to go south, there is no lack of finger-pointing and panic button-hitting. Various members of the team, and a few new faces who popped out of the bureaucratic woodwork for the first time, began taking pot shots. "The application has a memory leak!" "The search engine is no good!" "The testing was inadequate!" "The implementation was faulty!"

But amid the clamor, a few cooler heads began poring through the data, methodically analyzing the situation until it became clear that a minor component in the system, the Java drivers used by WebSphere and the search engine, were incompatible. Although the team had deployed the correct drivers, and the search engine documentation suggested that its driver would work with IBM's, it was not the case. The driver mismatch was causing sessions to spin out of control, eventually crashing the application server and the site along with it.

Happy Endings

Throughout the analysis, the project team made numerous calls to the search engine vendor and IBM. Once the problem was isolated, IBM stepped up with a solution, forwarding an about-to-be-released WebSphere patch that addressed the specific incompatibility. The patch was installed, the problem immediately disappeared, and eighteen months later, energy.gov is still running strong.

The DOE site garnered an Internet Best In Class award from *Content Week* magazine, a testament in part to the decision to go with a proven platform. As Brook Group prepares to launch the next version of Tacklebox at energy.gov this year, the site remains a WebSphere success story. From the use of an IBM SPC test lab during the development process, which proved that the site would withstand far more capacity than was expected on energy.gov, to the crucial last-minute driver fix, the power and reliability of the WebSphere-Tacklebox combination has proven a winner for energy.gov.

"When a large, public project begins to go south, there is no lack of finger-pointing and panic button-hitting"

LinuxWorld® CONFERENCE & EXPO

CONFERENCE: February 14 – 17, 2005 EXPO: February 15 – 17, 2005

HYNES CONVENTION CENTER • BOSTON, MA

Linux. Reliability. Efficiency. Cost Savings. If you're thinking about Linux or Open Source, the time is now. And LinuxWorld Conference & Expo is where you need to be. Expect the following cutting edge insights:

- Forward-thinking Keynote presentations by industry thought leaders from Novell, HP, Computer Associates, and MySQL.
- 10 hours of **FREE** educational programming on everything from the State of Linux and Open Source technology to Linux and desktop users.
- The world's leading hardware and software vendors as well as the hottest new start-ups giving you the opportunity to evaluate and test drive the latest technology.

And that's only the beginning. LinuxWorld Conference & Expo – where professionals come to do business better.

Register today at linuxworldexpo.com for your **FREE** Exhibit Hall pass.

LINUXWORLDEXPO.COM

Where OPEN MINDS Meet

Register Online With Priority Code: D2601

PLATINUM SPONSORS

BakBone Redefining Data Protection. · ca Computer Associates · hp invent · IBM · Novell · ORACLE · Sun microsystems

IDG WORLD EXPO

IBM's middleware innovates change

Automotive Industry Solutions

BY LARRY CARVALHO & CHAE AN

Larry Carvalho is market segment manager, Automotive Industry, IBM Software Group.

Chae An is director, Industrial Sector Solution Development, IBM Software Group.

You don't have to look any further than your own driveway to get a sense of some of the challenges facing the automotive industry.

First, is the type or brand of vehicle you're driving now different from your previous vehicles? The answer is likely yes. The days when buyers bought a Chevrolet, Ford, or another brand again and again are largely over. Today, many people also move between economy cars, sporty cars, mini-vans, SUVs, and luxury cars as their needs and desires change. This makes it very challenging for automotive companies to meet customer demands and to do so ahead of fierce competition.

Second, what happens when your vehicle needs maintenance? There's hardly any do-it-yourself anymore, and maintenance and repair is not simple for professionals. Increasingly sophisticated vehicles chock-full of intricate electronics and software have made it more difficult, and more essential, for service professionals to receive ongoing training. Parts suppliers also must keep pace with the ever-increasing complexity. If they don't, quality suffers, warranty issues and costs increase, and customer satisfactions drops.

Third, was your vehicle made in the U.S.? Whether or not your answer is yes, foreign-made vehicles have become so common that U.S. manufacturers are grappling with significant over-capacity in their manufacturing capabilities. That pressure on efficiency undermines their businesses.

In addition, automotive industry players need to compete on a global scale, offer costly incentives to customers, and deal with increasing government regulation. As a result of these pressures and more, the industry needs to keep in lock-step with rapidly changing customer demands and improve quality and efficiency from product design to vehicle service across their operations.

Many are looking to information technology (IT) to help them do so. They need to integrate their business and IT processes within and beyond their own organizations to allow them to respond with speed to customer needs, technology issues, market opportunities, and competitive threats.

IBM has worked closely with automotive customers for years. The top 10 worldwide motor vehicle and parts companies listed in the July 2003 *Global 500* use IBM WebSphere software. IBM Rational is number one in embedded software test automation tools, and seven of the top 10 worldwide automotive manufacturers use IBM Lotus e-learning solutions.

In addition, IBM has created software solutions designed specifically to address the challenges facing the automotive industry. The solutions are part of IBM's effort to deliver solutions based on customer preference for buying solutions designed for their industry. Each solution draws on the appropriate capabilities from IBM's WebSphere, Lotus, Tivoli, DB2, and Rational middleware brands, and IBM's industry-specific middleware combined with applications from independent software vendors (ISVs) and industry-expert services.

IBM automotive solutions can speed the design and development cycle, and improve plant operations and dealer communications. From design to ongoing service, the solutions help companies analyze data to improve quality and customer satisfaction. They also provide ways to meet varying customer expectations and respond faster to market shifts. Offering better collaborative tools and automating early warning business processes allows for cost-effective development and deployment.

"Each solution has multiple scenarios that address a range of industry challenges," said Larry Carvalho, market segment manager, Automotive Industry, IBM Software Group. "And they're based on open standards and industry initiatives so that companies can begin by addressing any challenge they choose and move on to the next when they're ready."

The solutions are also designed to allow customers to reuse existing and new applications and components. "The solutions are based on a services-oriented architecture

which means that each application (including legacy applications) can be constructed as a Web service to be reused and added to as needed," said Chae An, director, Industrial Sector Solution Development, IBM Software Group.

"This approach also enables the integration of back-office applications, such as Enterprise Resource Planning (ERP) systems, with plant floor applications or devices, such as those for scheduling, robotics and printing at a manufacturing facility," said An. "Also, because of the open architecture, the applications can integrate in today's heterogeneous IT environments," he added.

IBM's solution for automotive product design and innovation helps companies more effectively collaborate within and beyond their organizations. Automotive manufacturers have long depended on single-vendor CAD/PDM solutions to support their enterprise-wide engineering and design activities.

In recent years, however, industry trends including mergers and globalization, increased product complexity, more dynamic and volatile partnering on product value chains, and the need for tighter team integration and collaboration with primary partners and suppliers have created new challenges for OEMs (original equipment manufacturers), their primary suppliers, and their product line value chains. More and more, first tier suppliers are being forced to deploy, maintain, and integrate a widely heterogeneous environment of CAD/CAM/CAE and product data management systems, particularly if they wish to participate in multiple product value chains.

The design and innovation solution helps enable effective collaboration in these heterogeneous environments by providing seamless integration of PDM systems from multiple software vendors. This ensures that all parts of the system are well integrated and work together as one coherent enterprise PDM. The solution extends the basic capabilities in IBM's Business Integration (people, process, information) portfolio for federating data and processes across multiple heterogeneous PDM systems and data sources in a product design, development, and manufacturing environment. This enables real-time visibility and centralized access and management of product data, configuration, and process information throughout the product life cycle, across the partners working together in product design and development.

The solution also has an embedded system life cycle management offering that combines software, services, and consulting to improve the strategy, design, and support of in-vehicle software and electronics. These systems are expected to provide much of the innovation for automotive companies in the future and they also have a tremendous impact on vehicle warranties. The offering improves embedded system design, thus reducing warranty issues and improving product differentiation through enhanced features.

In addition, IBM has put together a set of investments, initiatives, and solutions that create a leading-edge, open standards-based hardware platform for computer aided engineering (CAE) simulation and analysis targeted for the automotive industry. Designed to help automotive companies find new ways to reduce product development time and cut costs for increased quality and competitiveness, the Automotive Engineering Innovation Framework (AEIF) initiative extends IBM's capabilities in Product Life Cycle Management (PLM). It also brings together IBM expertise in Grid and Deep Computing, virtual products, process innovation, and more.

Just as in the design and development process, automotive companies need to improve their manufacturing productivity. They face cost pressures and decreased profits due to excess and obsolete inventory, poor on-time delivery which has hurt customer satisfaction, difficulty implementing product innovations which hurts market share and limited ability to make the most of system capabilities such as electronic procurement.

IBM's solution for automotive manufacturing productivity addresses plant floor systems, ERP, and supply chain management.

For plant floor systems, the offering provides an integrated set of processes, tools, and services that help reduce the cost and complexity of IT for plant operations, improve plant management decision-making, and increase the flexibility of manufacturing operations.

The ERP offering helps companies integrate systems within and outside their organizations so they can make real-time, knowledge-based decisions within the supply chain. The solution allows an enterprise to extend data outward and collaborate effectively with partners, suppliers, and customers. The result is improved productivity and forecasting, as well as reduced cycle times and investments in inventory.

The supply chain management offering helps companies balance

"Understanding the industry's needs is key to the effectiveness of all the solutions"

demand and supply, improve customer service, and optimize inventory levels by continuously planning in real time across organizational boundaries. Its process, organization, and technology elements assist companies with production planning, profit optimization, and yield management. As a result, automotive companies can reduce planning cycle and lead times, as well as increase order fill rate, inventory turns, and throughput.

Another IBM solution for the automotive industry deals with sales and service issues. It consists of offerings to address quality insight, dealer collaboration, and consumer insight.

Quality issues facing the industry include recurring product recalls and rising warranty costs. IBM's quality insight offering is a process and technology infrastructure framework where the combination of information integration, analytics, and automation identifies and predicts defective products. It does so by creating an enterprise-wide view from manufacturing through customer ownership, thus providing capabilities for risk management, warranty cost containment, quality, and product improvement programs. The ownership experience can be improved and brand loyalty strengthened.

Automotive companies also need to improve their collaboration with dealers. Currently, they incur high costs due to inconsistent processes across dealerships. In addition, customer satisfaction is low and the relationship between dealers and parts manufacturers (OEM suppliers) is deteriorating.

IBM's offering for dealer collaboration, a workplace and business process infrastructure framework, focuses on improving the customer buying and ownership experience. It does so by enhancing collaboration and consistency of processes between the dealer and OEM partners. IBM's Workplace platform provides role-based capabilities for different personnel such as sales managers and mechanics at a dealership. Mechanics, for example, could view 3D drawings and instruction manuals using Workplace capabilities. Within this offering, automotive parts information is managed by the WebSphere Product Center.

The solution, therefore, helps provide dealers with integrated content, improved sales, purchasing, delivery of parts, accessories and service, issue management, and education throughout the aftermarket demand chain. The result is lower process costs, a better buying and ownership experience, and improved relationships between dealers and their suppliers.

Gaining insight into consumer needs is also important for automotive companies. In addition to declining customer loyalty and low customer satisfaction, they have had to absorb high sales incentives costs and have had difficulty implementing effective marketing programs. IBM's offering for customer insight includes enterprise and dealer customer relationship management capabilities, call centers, marketing, and customer intelligence telematics. The use of these capabilities can reduce call center costs and enable marketing programs through better intelligence.

IBM also has solutions for finance management and for human capital management. The solution for human capital management addresses the industry's high HR function costs and inconsistent HR processes. It also helps automotive companies speed up their responsiveness to HR service requests, and provide cost-effective training. The solution for Finance Management helps improve forecasting accuracy, shorten planning cycle times and the time between closing books and issuing financial reports, cut costs in the finance function and improve compliance with regulations.

"Understanding the industry's needs is key to the effectiveness of all the solutions," Carvalho said, and to ensure that this insight continues to be part of its offerings IBM is active in industry groups working on IT issues. Among them are the Automotive Industry Action Group, a U.S. organization similar to groups in Europe and Asia, which is working on industry standards for business-to-business integration, and the Automotive Open Systems Architecture (AUTOSAR) consortium, which is working on the creation and adoption of globally recognized, open-standards-based platforms for the electronics and software in vehicles.

"The pedal-to-the-metal pace of change in the automotive industry makes it a constant challenge for its players to be first to the finish line," said Carvalho. "But the right IT solutions can make a big difference in who wins the race."

"The days when buyers bought a Chevrolet, Ford, or another brand again and again are largely over"

CTIA WIRELESS 2005®

A Division of **CTIA**–The Wireless Association™

The Most Important Technology Event of the Year!

March 14-16, 2005 Ernest N. Morial Convention Center New Orleans, LA **www.ctia.org**

Register by Feb. 21st and receive a FREE pedometer!

Keynotes

Stay tuned for more exciting Keynote announcements...

Keynote Host
Steve Largent
President/CEO
CTIA-The Wireless Association

Day One

George Bodenheimer
President
ESPN Inc. and ABC Sports
Co-Chairman
Disney Media Networks

Day One

Daniel A. Carp
Chairman & CEO
Eastman Kodak Company

Day Two

John Chen
Chairman, CEO & President
Sybase, Inc.

Day Two

Jeong Han Kim
President
Samsung Telecommunications America

Day Three

Scott Ford
President & CEO
ALLTEL

Day Three

Len Lauer
President & COO
Sprint Corporation

Day Three

Stan Sigman
President & CEO
Cingular Wireless

Day Three

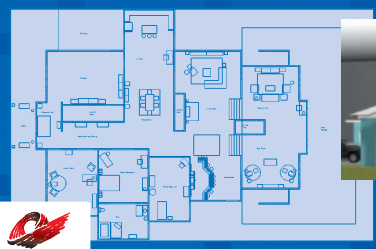
Robert Dotson
President & CEO
T-Mobile USA

CTIA WIRELESS 2005

is the most important global event where standards bodies converge to set the agenda for the future of the wireless industry. Network with the largest gathering of wireless engineers and technologists to find the tools you need to help advance the wireless industry.

CTIA WIRELESS 2005 offers endless opportunities for the wireless engineer and technologist:

- **CTIA Education** – *numerous sessions dedicated to wireless technology*
- **Tower Summit & Trade Show 2005** – *premiere wireless infrastructure event*
- **IEEE Wireless Communications Network Conference (WCNC) 2005** – *the industry's foremost conference for developing wireless standards and engineering*
- **Developers Conferences** – *CDMA University, Global LBS Challenge, WiFi VoIP Summit, and more*
- **Exhibits** – *a 400,000 sq. ft. floor displaying the latest in wireless technology*
- **Wireless News** – *More new wireless product announcements than any other show!*
- **CTIA Wireless Home** – *a 7,000 sq. ft. home that brings wireless technology to life*



COMSEARCH®

The Official Spectrum Manager and Wireless Home Integrator

Produced by

CTIA The Wireless Association™

Developing Standalone Client Applications

In a secure WebSphere 5 environment

BY ADITYA RUSTGI

Aditya Rustgi is a lead technical consultant at Perficient. He has been working with Java since 2000. He has worked with WebSphere 3.5.x, WebSphere 4.x, and WebSphere 5.1 since September 2001.
aditya.rustgi@perficient.com

In an enterprise WebSphere 5.1 deployment, the application business logic is developed as Enterprise JavaBeans (EJB). In many situations, there exists a need to execute standalone Java programs that run outside the container. To reuse the business logic present in the EJBs, the application client can use the EJB resources that are provided by the application server. However, in an enterprise deployment, these resources are protected from external use by a security system that requires clients to present appropriate credentials to access any of the application server resources. This article describes the programming model as well as the client and server configurations needed to develop a Java thin client application.

Programming Model

In order to connect to WebSphere Application Server 5.1 in a secured environment using an RMI/IIOP connection, an application needs to perform certain operations in a particular order. These operations are:

1. Initialize the Object Request Broker (ORB) inside the client JVM
2. Log into the application server providing the appropriate credentials
3. Asserting the identity of requester, obtain a reference to the appropriate resource
4. Execute the operations on the resource

This section describes these operations, their order, and the reason for executions.

Initialize the Object Request Broker

The first operation in the scheme above is to initialize the Object Request Broker. This is done by creating an InitialContext object passing in the information about the location of the target application server and the name of the InitalContextFactory implementation class. After performing an InitialContext creation, perform a default lookup so that the bootstrap host/port can be determined for the security server lookup.

```
Hashtable env = new Hashtable();
    env.put(Context.INITIAL_CONTEXT_FACTORY,
            "com.ibm.websphere.naming.WsnInitialCon
textFactory");
    env.put(Context.PROVIDER_URL, "corbaloc:iiop:
myhost.mycompany.com:2809");
    Context initialContext = new
InitialContext(env);
    Object obj = initialContext.lookup("");
```

If the target server is located at the same machine as the client, we do not need to specify the Context. PROVIDER_URL in the hashtable. If the target server is running on a remote machine, or if it is running on the same machine as the client but the IIOP listener port is not 2809, we have to provide that information to the InitialContext. In this situation, perform a new InitialContext programmatically ahead of the JAAS login. JAAS needs to know where the Security Server resides to verify that the user ID or password entered is correct prior to doing a commit(). By performing a new InitialContext in this way, the security code has the information needed to find the Security Server location. This can be performed in the following ways:

1. Define a system variable java.naming.provider.url at

the command line and set its value to the IIOP URL of the target WebSphere application server

```
java –Djava.naming.provider.url=corbaloc:iiop:
myhost.mycompany.com:2809
        -Djava.naming.factory.initial=com.ibm.web-
sphere.naming.WsnInitialContextFactory
```

2. Create a file named jndi.properties and place the file in the class path of the client JVM. The file will contain the following information:

```
java.naming.provider.url=corbaloc:iiop:myhost.mycom-
pany.com:2809
java.naming.factory.initial=com.ibm.websphere.nam-
ing.WsnInitialContextFactory
```

3. Provide it to the constructor of the InitialContext as shown in the example above

In a WebSphere 5.1 cell, the URL can point to any WebSphere application server running inside the cell. Since this initial context is only used to initialize the ORB running inside the client JVM, and no contact with the application server is made, we do not need to provide the credential during the creation of the InitialContext here.

Perform a JAAS Programmatic Login

A secured enterprise bean application requires the identity of the user that is looking up the EJB and invoking the method for authentication and authorization purposes. This information has to be collected in the client application and provided to the user. There are two different models for performing this, batch programs and user interactive programs.

Batch Programs

These are standalone programs that run without user interaction and use one identity to perform all the operations. In this case, the information can be provided statically in the sas.client.props properties file. The application client will then be written in such a way that the runtime uses this information rather than collecting from the user. In order to perform this mode of authentication, the following properties must be specified in the sas.client.props file:

- ***loginSource:*** This specifies how the credentials for user validations are obtained. For batch programs this value should be set to properties or keyfile. If set to properties, the value for the username and password are specified in this file as mentioned below. If the value is keyfile, then the username and password can be specified in a keyfile.

```
com.ibm.CORBA.loginSource=properties
```

- ***loginUserid:*** This specifies the username to be used to authenticate the client code and it will be the identity of the caller for EJB authorization checks on the application server. This must be specified in conjunction with the loginSource set to properties above.

```
com.ibm.CORBA.loginUserid=wasadmin
```

- ***loginPassword:*** This specifies the password for the username specified above

```
com.ibm.CORBA.loginPassword=wasadminpassword
```

After initializing the ORB (as specified above), the application can be written in a normal way. At locations where the client looks up the EJB resources, the application server will challenge the client for authentication information. This information will be gathered by the client runtime from the sas.client.props and passed onto the application server. The code in Listing 1 depicts this model.

The component of the test application in Listing 1 that accesses EJBs is shown to run both in a separate thread and the main thread. Test is an EJB that has authorization rules defined using declarative security.

User Interactive Programs

As opposed to the batch programs, user interactive programs collect the authentication information from the user. The client application then runs using the identity of that user. In this case, providing the authentication information inside the sas.client.props will not suffice. For this model, there are two options:

1. Use the WebSphere default information collection mechanism. WebSphere provides two user interfaces that can be used to collect the username and password from the user.
   a. ***GUI:*** This is a basic Swing window that prompts the user to specify the username and password.

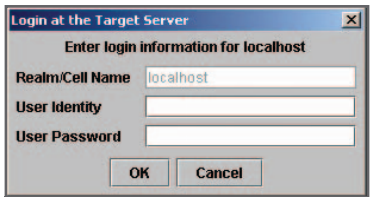


FIG 1: GUI INTERFACE

   b. ***Stdin:*** This is the command line standard input.

FIG 2: STDIN INTERFACE

Using this model, the application can be written in exactly the same way as the batch programs above. No special programming is needed except initializing the Object Request Broker.

2. Provide your own user interface and own the responsibility of collecting this authentication information and then use that information to programmatically log onto the application server. Once the authenticated Java subject is obtained from the logging in process, it is used to execute the actions that require access to the EJB resource on the application server (see Listing 2). (Listings 2 and 3 can be downloaded from sys-con.com/websphere/sourcec.cfm.) Additionally, the application can create an implementation of the CallbackHandler interface that creates the user interface to collect the information from the user.

To execute Listing 2 the sas.client.props would need to be modified to contain the following.

```
com.ibm.CORBA.loginSource=none
```

The things to note in this programming model are:

1. ***CallbackHandler:*** A CallbackHandler is a JAAS mechanism to collect authentication information from the user. Since we are taking up the responsibility of collecting that information using our custom interface, we will be using a nonuser interactive WebSphere CallbackHandler implementation:

```
com.ibm.websphere.security.auth.callback.
WSCallbackHandlerImpl
```

We can also use the following CallbackHandler:

```
· com.ibm.websphere.security.auth.callback.
WSGUICallbackHandlerImpl
· com.ibm.websphere.security.auth.callback.WSStdinCal
lbackHandlerImp
```

These two CallbackHandler implementations will produce a user interface identical to the GUI and Stdin interfaces discussed previously. In the sample code above, reflection has been used to create the CallbackHandler. The benefit of this is that you would not need to place that class in your build classpath. When the application runs (configured as shown in the next section), the runtime shall provide the definition of the class. The developer can choose to use the two-parameter constructor to create an object of that class.

2. ***LoginContext:*** A LoginContext is a vehicle in the JAAS mechanism to specify the Login Module alias (client application should use "ClientContainer" login module) and the CallbackHandler. The LoginContext is used to login, which in the background invokes the CallbackHandler to collect the authentication information. It then passes this information to the login modules that are specified using the alias.
3. ***Login action:*** Creates a subject that contains the principles corresponding to the logged in user.
4. ***WSSubject.doAs():*** In order to associate the current thread's identity with the authenticated procedure, the doAs method on the WebSphere WSSubject is invoked.
5. ***PrivilegedAction:*** The code to access the EJB resources on the application server should be wrapped inside the run method of a class that implements PrivilegedAction. The code running inside this block is associated with the identity of the subject passed into WSSubject.doAs().
6. ***Thread:*** If your client application creates threads, the client JVM runtime does not propagate the thread identity to the newly created thread. Therefore, your application will need to make sure that the authenticated subject is available to each thread, if that thread looks up EJB resources on the application server.

"The application server will challenge the client for authentication information"

Running the Client Application

In order to start and run the thin Java application client there are several settings that need to be provided to the Java runtime at the command line. Listing 3 shows a sample batch file that launches the TestJavaThinClient class.

The following items need to be noted about the patch file:

1. ***JVM:*** In order to use Java thin clients, we have to use the IBM JRE. The runtime that has been used in this example is from the WebSphere application client. The runtime that comes with WebSphere Application Server can be used instead.
2. ***Classpath:*** The Client JVM uses hierarchical class loaders to load all the relevant classes. There are three class loaders involved.
   a. ***Extension class loader:*** The classpath for this class loader is specified through the value of java.ext.dirs on the command line. This should contain the JRE extention classes as well as all the IBM WebSphere specific classes. If your application used JMS, then this is the location where you would put JMS specific jars.
   b. ***Bootstrap class loader:*** The classpath for this class loader is specified using the –Xbootclasspath/p: syntax on the classpath. This will contain the ibmorb.jar and the WebSphere properties.

eclipse CON™ 2005

2nd Annual Gathering of the Eclipse Community

Plug-in to Eclipse

This is your opportunity to get in-depth technical information from the Eclipse experts, learn the latest tips and techniques for using the tools, network with fellow enthusiasts and experience the breadth and depth of the Eclipse community. Attending EclipseCon™ will expand your knowledge and make Eclipse work better for you.


Tim O'Reilly


Urs Hoelzle


John Wiegand


Erich Gamma

Keynotes: Tim O'Reilly,
CEO of O'Reilly Media

Urs Hoelzle,
VP of Engineering, Google

John Wiegand and Erich Gamma,
Eclipse Platform PMC Leaders

Over 60 Sessions and Tutorials for beginner through advanced participants. Some of the conference sessions and tutorials include: Eclipse Platform · Rich Client Platform (RCP) · Test and Performance Tools · Debugging · J2EE Development · Web Development and Tools · Building Plug-ins · Modeling Framework (EMF) · C/C++ Development · Business Intelligence and Reporting · Embedded Development

February 28-March 3, 2005
Burlingame, CA

Register at www.eclipsecon.org

c. ***Application class loader:*** All the client application classes, utility JARs, and EJB client JARs are specified in this classpath. This is specified using the –class path syntax on the command line.

3. ***SAS client file:*** The location to the SAS client file is specified using com.ibm.CORBA.ConfigURL parameter on the command line. This URL points to the sas.client.props file.
4. ***STDIN cLIENT SAS:*** This is a SAS client file where the login source has been set to standard input.
5. ***SOAP client file:*** The location of the SOAP client file is specified using the com.ibm.SOAP.ConfigURL parameter on the command line. This URL points to the soap.client.props. The properties in this file dictate the SOAP communication between the application client and the application server. The client can make JMX administrative calls to the application server using the SOAP connector. This file dictates the security settings for that communication. If your application does not make any JMX administrative calls, you can point this to the default soap.client.props file located under the WebSphere/properties folder.
6. ***JAAS login configuration:*** The location to the JAAS login configuration file is specified using the java.security.auth.login.config. This file specifies the login modules for the JAAS login alias ('ClientContainer') that were used in our application.

Summary

To write a Java thin client to access EJB resources on an application server certain programming models and configurations need to be specified. This article discussed a few of these programming models and mentioned the property settings needed to configure the client JVM to successfully make the remote calls to a secured application server.

LISTING 1

```
package com.perficient.client;

import java.rmi.RemoteException;
import java.security.PrivilegedAction;
import java.util.Hashtable;

import javax.ejb.CreateException;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.rmi.PortableRemoteObject;

import com.perficient.test.Test;
import com.perficient.test.TestHome;

/**
 * @author aditya.rustgi
 *
 */
public class TestClient {

   public static void main(String[] args) {

         try {

//Perform an InitialContext and default lookup prior to logging
//in to initialize ORB security and for the bootstrap host/port
//to be determined for SecurityServer lookup.

                     Hashtable env = new Hashtable();
                     env.put(Context.INITIAL_CONTEXT_FACTORY,
                     "com.ibm.websphere.naming.WsnInitialContextFactory");
                     env.put(Context.PROVIDER_URL,
                     "corbaloc:iiop:localhost:3737");
                     final Context initialContext = new InitialContext(env);
                     initialContext.lookup("");

                     // Execute an action in the same thread.
                     new TestAction(initialContext).run();

                     // Execute an action in a separate thread.
                     Thread t = new Thread() {
                              public void run() {
                                       new TestAction(initialContext).run();
                              }
                     };
                     t.start();
                     t.join();
         }
         catch (NamingException ne) {
                  System.out.println("Naming Exception");
                  ne.printStackTrace();
         }
catch (InterruptedException inte) {
                  inte.printStackTrace();
         }
   }

   public static class TestAction {
         private Context ic;

         public TestAction(Context ic) {
                  this.ic = ic;
         }

         public Object run() {

                  Object o;
                  try {
                           o = ic.lookup("ejb/com/perficient/test/TestHome");
                           TestHome testHome = (TestHome)PortableRemoteObject.narrow(o,TestHome.class);
                           Test test = (Test)testHome.create();
                           System.out.println("Message from Test EJB:"+test.getHello());
                  } catch (NamingException e) {

                           e.printStackTrace();
                  } catch (RemoteException e) {
                           e.printStackTrace();
                  } catch (CreateException e) {
                           e.printStackTrace();
                  }
                  return null;
         }
   }
}
```

For Listing 2 and Listing 3, go to www.sys-con.com/websphere/sourcec.cfm

A Step in the Right Direction

IBM makes advances in the IBM software installation process

BY JOSEPH R. MARQUES

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

IBM is known in the industry for releasing software products that are tremendously powerful yet tremendously complex; unfortunately, because of the latter their software installations are sometimes quite difficult. Actually, in their defense, "are" is quickly becoming "were." IBM has been putting forth quite an effort to revamp and fix the installation challenges that developers and consultants have seen in the past. This Pro-Spective will attempt to touch upon the major recurring issues with IBM installations, both in the WebSphere suite of products and others, and look at how IBM is aggressively pursuing solutions so that these issues will be a mere thing of the past.

Fool's Gold

Everyone remembers the "worksheet" you had to fill out for WebSphere Portal Server v4 to prepare for the droves of configuration options that the installation would require. This step-by-step manual, which could be ordered on CD or downloaded (and which was supplementary to the regular Portal documentation), walked a user through the installation; it explained each and every option up front, gave examples for most options, and let the user create *the configuration* before attempting the installation. Due to its success, this worksheet quickly became a critical companion for most installers. It helped avoid many installation issues that would cause the lengthy installation process to have to be restarted.

Unfortunately, the procedures for correctly cleaning the machine after a failed installation were poorly documented. Granted, the Troubleshooting Guide did provide a list of potential errors you might receive during an installation, but it wasn't exhaustive. So it was *possible* to generate an invalid "worksheet" configuration, have the installation fail with an unknown error, and then have to recover via (mostly) trial and error.

The most common complaint, however, from end users and/or clients using Portal version 4 was that the installation tried to lay down a complete IBM solution: IBM HTTP Server (IHS), WebSphere Application Server (WAS), DB2, Informix Dynamic Server (IDS), and WebSphere Portal Server (WPS). Only after you installed everything (assuming there were no conflicts with similarly functioning existing software), could you run a couple of configuration scripts that could move parts of the infrastructure to other vendors, such as LDAP to Active Directory, or DB2 to Oracle. The point is that the initial installations of certain parts of the IBM stack were not useful because clients were going to move to other vendors for their business reasons anyway.

Another IBM product, Content Manager, shares some of the same installation issues as Portal as well as introduces new ones. Before Content Manager version 8.2, it, like Portal, also suffered from having panel after panel of configuration options, which was simply overwhelming for amateur users. The problem was exacerbated for new users because if they were unfamiliar with the product or its dependencies (such as DB2) it might have taken several hours to research the proper values for all the fields on all these panels. If a user was able to install the product, the values entered sometimes resulted in an invalid configuration; the installation succeeded, but the program didn't run.

Because of all these different factors the installation processes for certain products were sometimes reduced to little more than trial and error processes, which beg us as developers to ask the tough question: should products really require experts to install?

Rags to Riches

Those are just two examples; there are others. The emphasis should not be on things that were done wrong, but instead on what lessons can be learned to

enhance the experience for end users in the future. It is well understood that IBM has extremely complex products that need to be integrated into highly complex, heterogeneous environments – so it's no wonder that the installation process is overly complex and not necessarily intuitive. With experience, like that mentioned above, and critical user feedback, which is provided continuously, IBM has taken a serious look at this problem and has recently come up with at least one solution that will be making its way onto the stage soon enough. It is called Integrated Runtime (IR).

A Quick Fix, for Now...

IR is an impressively fast response to the problem that IBM faced, and that's because it operates primarily on the principle of data hiding; it does not attempt to, nor does it want to (at this point in time anyway), change the actual installation procedures. Its focus is to put the installation issues into the hands of experts for each product. These people have gathered knowledge and have experienced installing each product on different platforms. They are best able to aggregate all of this knowledge to produce reusable "wrappers" that sit on top of the existing procedures, providing customized installation scenarios; in other words, preconfigured solutions (with all the bugs and kinks worked out, hopefully).

IR's current version is 1.1, with an upcoming 2.1 release targeted for the end of Q1 2005. Figure 1 gives a high-level look at how IR is designed. It can be broken down into two primary units: application wrappers and solution wrappers. Each of the units work together in a cooperative fashion as a single solution: the application wrapper controls the execution of one or more existing installation scripts as well as possibly one or more pre- or post-installation procedures; the solution wrapper coordinates one or more application wrappers to provide a solution that can potentially integrate any number of products, of any version, on any set of platforms, with each other.

Application wrappers facilitate reuse within and amongst product teams. Looking forward, if all teams opened up their respective installations through wrappers (instead of having experts come in and fix the issues for them), it would reduce the time necessary to perform integration testing because the tasks would be broken down into smaller and more manageable pieces that could then be reused in this framework provided by IR for other solutions.

Not only does a framework like this one enable a clean and consistent interface to the installer, but it also abstracts away the concept of separate installation logs and installation error codes. Since everything now gets installed *through* this framework, everything would also report to it.

In fact, to reap the most benefits, it should be the responsibility of this expert team to break any tasks that are specific to integration out into separate wrappers,

"IBM has been putting forth quite an effort to revamp and fix the installation challenges that developers and consultants have seen in the past"

which could be included in different solutions as necessary. Figure 2 shows a product map where the numbered lines are these procedures, which integrate their respective endpoints.

According to the diagram, an integrator can presume that if IDS and DB2 were packaged as a single solution it would require only two application wrappers – one for each product, more clearly shown in Figure 3 as Solution Wrapper A. It is easy to see that none of the integration wrappers are needed because there aren't any lines *between* the products in this solution wrapper.

However, if a client wants to install Portal and have it integrated with either of the aforementioned prod-

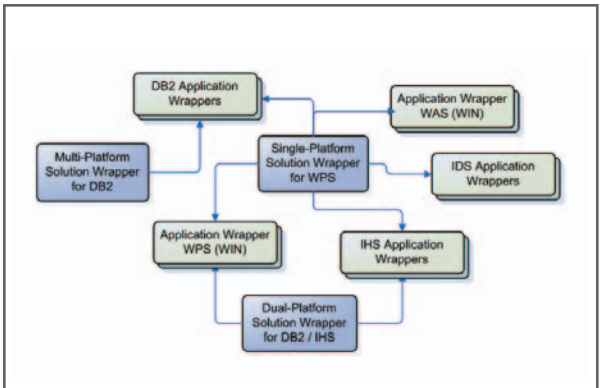


FIG 1: HIGH-LEVEL DESIGN FOR INTEGRATED RUNTIME

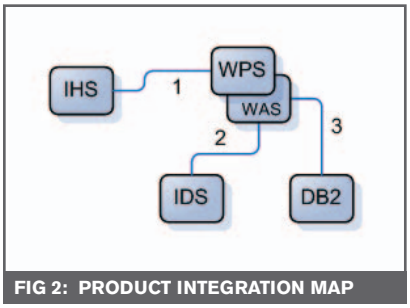


FIG 2: PRODUCT INTEGRATION MAP

"The ideal solution would be for the development teams to collaborate on the interfaces"

ucts, it would require four application wrappers: two for the WPS/WAS combination, one for the other product, and one more for the integration piece between them. Solution Wrapper B in Figure 4 shows what this Portal-DB2 integration would look like; the line labeled 3 corresponds to the application wrapper that encompasses the integration tasks required to properly fuse the two products together.

By wrapping the core installation procedures for each product separately from the procedures needed for integration, providers may recombine products and/or components as they see fit for redistribution as highly customized solutions for clients.

Lastly, by installing products within this framework, the concepts of platform and location fade. As far as the wrapper is concerned, it does what the framework tells it to do. Since the framework supports distribution across machines comprised of a heterogeneous set of operating systems in different geographic locations, wrappers written to work within that framework automatically gain that flexibility as well. This allows for a zero effort transition from a local installation to a remote one; in fact, it allows for a single machine to coordinate the installation of products across a variety of machines at the same time.

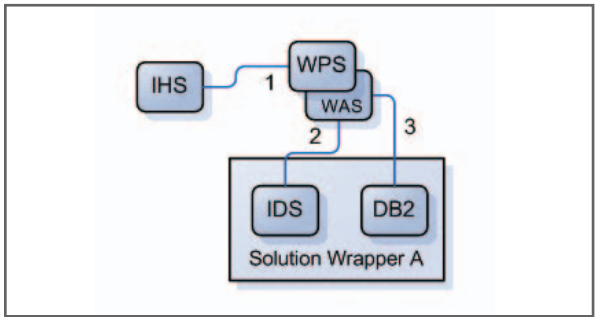


FIG 3: SIMPLE SOLUTION WRAPPER WITH MULTIPLE PRODUCTS

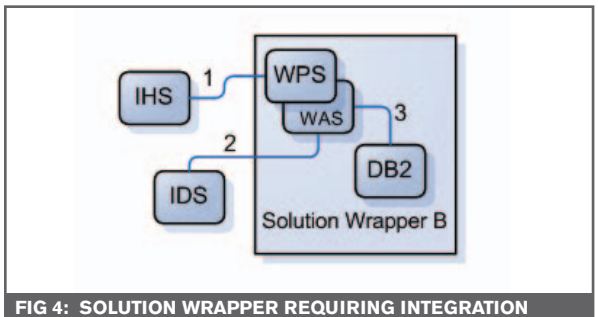


FIG 4: SOLUTION WRAPPER REQUIRING INTEGRATION PROCEDURES

Problem Solved?

But what do developers of installation scripts really have to worry about? Is it important to keep in mind that multiple products may integrate with the operating system, or an LDAP, or even a third-party program to provide security? Should time be spent to ensure that products from a single vendor cooperate and/or share some values to provide an integrated service? Is it important to be mindful of the developers and consultants who will want to install multiple versions of the same product on the same machine, thus requiring investigation into co-existence issues? Yes *and* no.

It is true that all of these issues are important, but I argue that they shouldn't be pushed to a separate group of people to write the actual installation scripts or application wrappers. Instead they should be owned by the actual development teams who have the deep product expertise to ensure the products work together in an integrated environment. Integration is not going away - it is becoming more prevalent.

Looking Forward

Integrated Runtime should only be looked at as a temporary fix because this solution doesn't truly eliminate the installation issues seen in the past – it only hides them. Don't get me wrong, the idea of wrappers and abstraction is good, but wrappers alone aren't providing a complete solution. This framework will have greater impact when the proprietary installation procedures are replaced and/or removed from the equation altogether.

The ideal solution would be for the development teams to collaborate on the interfaces that their products will use to communicate with one another; moreover, each needs to provide a separate and convenient, yet consistent, way for the installers to write the scripts and/or wrappers on top of them. The key is agreement. They all need to agree.

Having a solution that is overwhelmingly different from how things currently operate cannot be implemented quickly.

Most likely IBM will use a hybrid solution: wisely continuing to invest in Integrated Runtime while at the same time passing down specific suggestions and requirements to the product groups. As the issues are addressed closer and closer to the root of the problem, IBM's installation procedures will become more and more flexible and configurable. Stay tuned…down the road, I'm sure they will become as impressively robust as the products they are installing.

Reach Over 100,000

Enterprise Development Managers & Decision Makers with...

Offering leading software, services, and hardware vendors an opportunity to ***speak to over 100,000 purchasing decision makers*** *about their products, the enterprise IT marketplace, and emerging trends critical to developers, programmers, and IT management*

Don't Miss Your Opportunity to Be a Part of the Next Issue!

Get Listed as a Top 20* Solutions Provider

For Advertising Details Call 201 802-3021 Today!

*ONLY 20 ADVERTISERS WILL BE DISPLAYED. FIRST COME FIRST SERVE.

SYS-CON MEDIA

The World's Leading i-Technology Publisher

Using Macromedia Flash MX 2004

Develop Web Services Clients

BY SONNY HASTOMO

Sonny Hastomo is a product development consultant with the electronic delivery channel group for banking solutions at Sigma Cipta Caraka, Ltd (an IBM business partner) based in Indonesia. His current focus has been on integrating and interfacing legacy applications using the OS/400 and AIX operating systems. sonny.net@rocketmail.com

Interoperability is the key to using Web services architecture since many applications can collaborate to maximize business functions. With the capability of creating an application on a different architecture, including the application architecture, hardware infrastructure, and even the operating system, Web service architecture is the bridge to optimizing the IT Department.

In this article we will create a simple Web service using an EJB application and we will create a Flash MX project as a Web service client interface. We will use Websphere SDK v5.0.1 for the server-side scenario and Macromedia Flash MX 2004 for the client side. The complete source code for this article can be downloaded from www.sys-con.com/websphere/sourcec.cfm.

Prerequisites

You will need the Windows system that installed Macromedia Flash MX 2004, and you will also need an operating system (Windows or Linux) on which you have installed WebSphere SDK Web Service 5.0.1.

Server-Side Scenario

Creating a Simple Web Service from an EJB Application

First, create the J2EE Enterprise Application Project – for this article, we named it MATHServices.

Next, open J2EE Perspective from Window > Open perspective > other > J2EE. Then open J2EE Hierarchy and expand the EJB Modules; you will see the MATHServicesEJB Project as shown in Figure 1.

Create the Session Enterprise Bean at MATHServiceEJB Project (MathOperation)

Create the session enterprise bean called MathOperation and enter the following properties (see Figure 2):

- ***Bean name:*** MathOperation
- ***Source folder:*** ejbModule
- ***Default package:*** com.math.service

Add the logic function in MathOperationBean. Click at MathOperationBean class; then put the following code to that class:

```
public String
ApplicationInfo(String
strUserName) {
   return "Hello " + strUserName
+ " Welcome to Math Operation
Service";
 }
 public Integer Multiply(Integer
a, Integer b) {
   return new Integer(a.intValue()
* b.intValue());

 }
 public Integer Add(Integer
a,Integer b) {
   return new Integer(a.intValue()
+ b.intValue());
 }
 public Integer Substract(Integer
a, Integer b) {
   return new Integer(a.intValue()
- b.intValue());
 }
```

Now you are ready to move to the outline perspective and promote ApplicationInfo, Multiply, Add, and Subtract method to the remote interface (see Figure 3).

Finally deploy the code for the MathOperationEJB (see Figure 4).

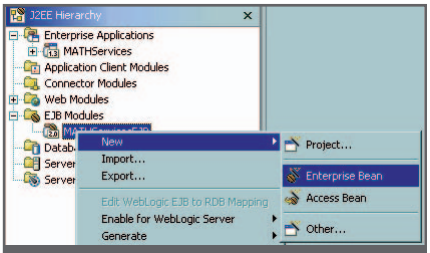


FIG 1: OPEN MATHSERVICEEJB PROJECT

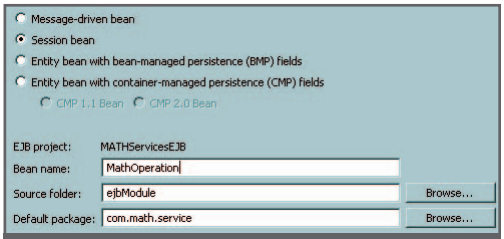


FIG 2: CREATE SESSION ENTERPRISE BEAN CALLED MATHOPERATION

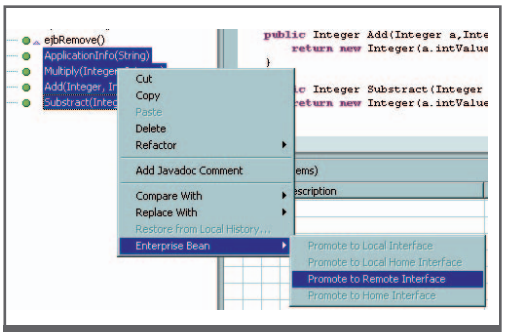


FIG 3: PROMOTE TO REMOTE INTERFACE

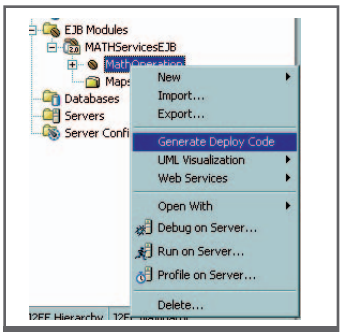


FIG 4: GENERATE DEPLOY CODE

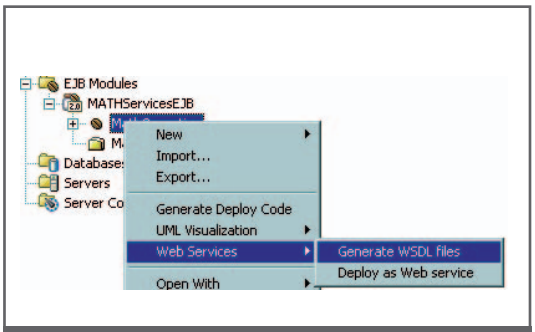


FIG 5: GENERATE WSDL FILES

Generate a Web Services WSDL

Expand MathOperationEJB and right-click MathOperation session bean and start Web Services > Generate WSDL File (see Figure 5).

Enter the following properties from the Web Service Deployment Settings screen:

- ***Web service deployment environment selection:*** Use Default
- ***Web project:*** MATHServicesWeb

Click Next. On the Web Service EJB Configuration screen, configure the enterprise bean as a Web service, click Next (see Figure 6).

On the next screen we will configure the JavaBean as a Web service (see Figure 7). Use the default parameter from Websphere (configure the JavaBean as a Web service). The following parameter is:

```
- Web Service Uri = http://tem-
puri .org/com.math.service.
MathOperation
- ISD File = Web Content/ WEB-
INF/isd/java/com/math/service/
MathOperation.isd
- WSDL Service document name =
Web Content/ wsdl/com/math/ser-
vice/MathOperationService.wsdl
- WSDL binding document name =
Web Content/ wsdl/com/math/ser-
vice/MathOperationBinding.wsdl
- WSDL EJB binding document name
=  Web Content/ wsdl/com/math/ser-
vice/MathOperationEJB.wsdl
- WSDL interface document name =
Web Content/ wsdl/com/math/ser-
vice/MathOperation.wsdl
- WSDL schema folder name =  Web
Content/ wsdl
```

The Web service definition file will later create a Web Content folder in your MATHServicesWeb Project. Click next to proceed to the Web Service Deployment Setting.

Specify the methods to deploy at the Web Service JavaBean Methods screen. Edit the encoding style for each method, if required. At this screen you can configure which method/function you want to publish as Web services. For our purposes, we will publish all the JavaBean methods that we have created. For both input and output encoding, choose SOAP encoding.

Click Finish to create the Web Service Deployment Description/WSDL file. Now run the Server Application: expand Server Configurations, right-click WebSphere v5.0 Server Configuration, and choose Run on Server.

Client-Side Scenario

Creating a Flash MX Project as a WebSphere Web Service Client Interface

Create the Flash MX Project – here we named it MathClientPJX.

On the General tab of the New Document box, choose Flash Document to create a document file to work with the input/output interfaces.

Add a label component on the canvas to design the interface for the Flash MX document. Give it the instance name: lblAppNotified [you will use this later to display the invoked method ApplicationInfo(String arg)].

Figure 8 shows the result of creating the Flash MX document interface; you will need to create three

-continued on page 49

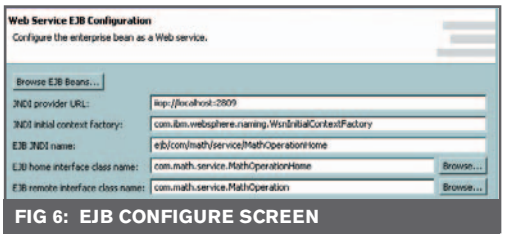


FIG 6: EJB CONFIGURE SCREEN

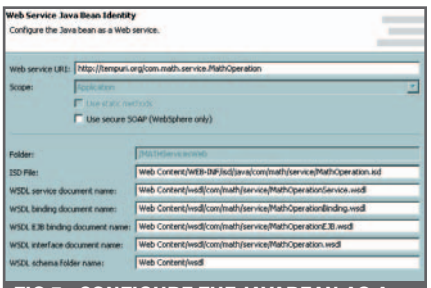


FIG 7: CONFIGURE THE JAVABEAN AS A WEB SERVICE

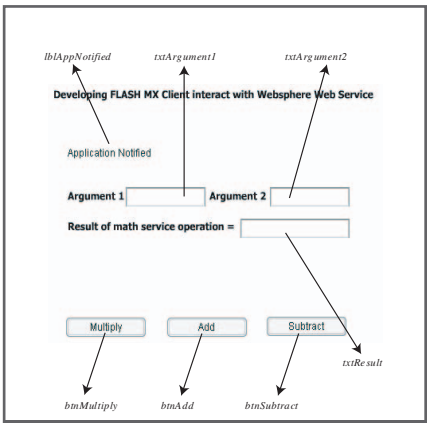


FIG 8: FLASH MX DOCUMENT INTERFACE

Instance separation provides facility of information

Profiles for WebSphere Application Server 6.0

BY KEVIN HAVERLOCK

Kevin Haverlock is an advisory software engineer for IBM's WebSphere Application Server product. He joined IBM in 1995 at Research Triangle Park, NC, where he worked as a developer for the Tivoli division. In 2000 he transferred to the WebSphere Application Server organization and is currently an architect and developer for the WebSphere Application Server Express product. kbh@us.ibm.com

The new IBM WebSphere Application Server (WAS) v6.0 software introduces the concept of Server Profiles. Profiles can be thought of as a specific server runtime environment operating within a separate instance of the JVM.

Each runtime environment has its own configuration files, logs, properties, and other attributes. Profiles can make each Java 2 Enterprise Edition (J2EE) application server runtime unique and separate from the server binaries and from other profiles. The separation of static binaries from configuration files provides a number of benefits for system administrators.

The WAS v6.0 profiles are similar to the wsinstance tool provided with WAS v5.x, but with some important differences. The wsinstance tool creates configuration files for separate JVM instances, but shares other information across instances of WAS v5.x. In contrast, profiles draw a sharp degree of separation, so much so that each profile is separately administered with its own configuration, logs, J2EE applications, and other attributes. The separation that profiles provide allows for a unique instance of WebSphere that has not been available in previous releases of WAS.

Easier Than Multiple Installs

More than one instance of the same IBM WAS product may be required on the same machine. This is often the case if the administrator desires to keep a degree of separation between applications and environments. This may be the case if you have a development group sharing the same WebSphere install instance. When using the WAS profiles, each profile runs as its own 1.4.2 JVM instance, thus providing a physical as well as a logical separation between instances of the product. This offers an advantage over multiple installs because it saves space. In addition, service updating is simpler since an administrator only needs to maintain a single set of product binaries. Profiles can also be created and deleted more easily than with whole product installs. With the separation of product binaries and profiles, it's possible for a system administrator to allow groups to have access to the profile information that defines their application server, while at the same time locking down the WAS runtime binaries.

Easier Application Updates

Another useful aspect of profiles is the staging of application updates. In this scenario, you have updated an application and wish to test it on the targeted WebSphere Application Server. The problem is that an older version of your application is already running and being used. You could uninstall the application and reinstall the new update, yet that would require you to take the application offline and then reinstall if your test fails. A solution would be to create a new profile and test it there. The test profile will be operating in its own instance of WAS and will be logically separated, thus giving you a level of security and separation should your tests fail. Of course, it only protects you while running in the context of WAS, and not if the two applications are sharing the same database or any other shared resource the two applications may access.

Figure 1 shows a typical WebSphere Application Server Network Deployment topology running with two profiles. The first is a default profile for a WAS Base instance; this profile contains information for one server. The second is a WAS Deployment Manager instance that contains two servers in the profile. Both are logically grouped within a node. A node normally corresponds to a physical computer system with a specific IP address. Associated with the application server is the embedded HTTP server, which directs requests to the application server. The embedded HTTP can be associated with a production HTTP server such as IBM's HTTP Server product based on Apache, or another third-party product.

The profile defines the characteristics of the WAS instance that is running in the context of the JVM. The profile contains all of the information that makes the server instance unique, such as port mappings, available services, data sources, JDBC providers, etc. The profile information

is contained in the <WAS_ROOT>/profiles directory. Table 1 illustrates the contents of the profile directory and their functions within the WAS. A default profile is created for you when you initially install WAS v6.0.

Creating New Profiles

For one reason or another, you have decided to create a new WAS Profile. Perhaps you have two development groups that share the same install instance but have different configuration environments. Or perhaps you wish to have two versions of the same application running, but with different environments. WAS v6 supports two ways of creating a new profile: the profile creation wizard and command line.

Profile Creation Wizard

The easiest way to create a new profile is to run the Profile creation wizard tool located in the <WAS_ROOT>/bin/ProfileCreator directory. Figure 2 shows the profile creation tool that will step you through the process of creating a profile. You will need to decide the name of the profile, the directory of where the profile information will be stored, the node and hostname of the server, and any changes to the default ports. The wizard will attempt to avoid WAS port conflicts by looking at the existing configured ports. It does not attempt to look at services outside of WebSphere that may already be running on those ports during the profile creation.

Creating Profiles with the Command Line

Another way to create a profile is via the wasprofile command line. The wasprofile creation tool is located in the <WAS_ROOT>/bin directory of the application server. For Linux, the file is wasprofile.sh and for windows it's wasprofile.bat. The wasprofile command is the primary tool used to manage profiles and provides the greatest amount of flexibility. Listing 1 creates a profile called MyNewProfile.

The -profilePath specifies the location in which the profile directory should reside. Normally, profiles are placed in the profile repository directory located in <WAS_ROOT>/profiles, but this can be overridden.

The parameter -templatePath specifies the location of profile templates that will be used in creating the new profile. The profile templates directory contains a number of Ant scripts and default configurations that act as a starting point for the tool in creating a new profile. Which template is used is important when considering what instance of WebSphere you wish to create. As an example, the WAS ND version of the product supports running in three modes: standalone base application server, a managed base application server, or as a deployment manager. The specific example uses the default profile to create a standard application server.

The profile tool only supports executing one instance at a time. If you try to start another instance of wasprofile while the first one is still completing, you will receive an error message.

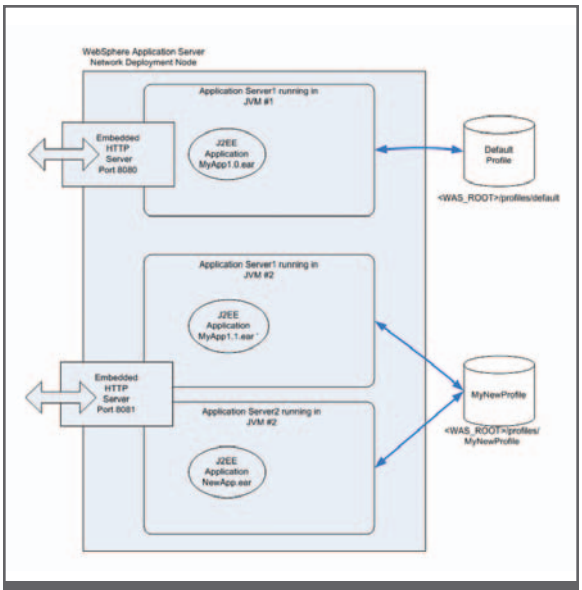


FIG 1: WEBSPHERE APPLICATION SERVER NETWORK DEPLOYMENT NODE

The -nodeName and -cellName specify the name of the node and the name of the WAS cell. A node is a grouping of servers, while a cell is a logical grouping of nodes.

The -hostName is the name of the machine on which that the profile resides.

When the profile is created, the directory structure described in Table

BIN	The bin directory contains executables that can be run within the context of the profile that was created. You can also use the bin directory created in the <WAS_ROOT>/bin directory, but you will be required to specify which profile the operation is being performed against.
CONFIG	The config directory contains all of the default configuration files that are associated with the WAS. Since it is similar in structure to WAS 5.x, the configuration files are in XML format.
ETC	Contains key store and other information. If you installed the samples, the key store files are located in the directory.
FIRSTSTEPS	Contains the J2EE application files that have been installed within the product.
SAMPLES	Contains scripts and source code for the samples. The directory only exists if you elected to install the samples.
INSTALLABLEAPPS	Contains applications that can be installed into WAS
INSTALLEDAPPS	Contains the J2EE applications that have been installed in the servers for this profile.
INSTALLEDCONNECTORS	This directory contains installed resource adaptors. The resource adaptors are system-level drivers that a Java application can use to connect to an Enterprise Information System (EIS).
LOGS	Contains the error logs.
PROPERTIES	The Properties directory contains the individual property information and is used for version control of the runtimes that are currently installed. The information is specific to the profile.
TEMP	This directory contains temporary files used by WebSphere and includes compiled JSP pages.
TRANLOG	Specifies the name of the directory where the server stores transaction service log files for recovery.
WSTEMP	This directory contains temporary files used by WebSphere administration, which include user preferences and workspaces. The directory contains subdirectories for each administrative user.

TABLE 1: THE PROFILE DIRECTORIES AND THEIR FUNCTIONS

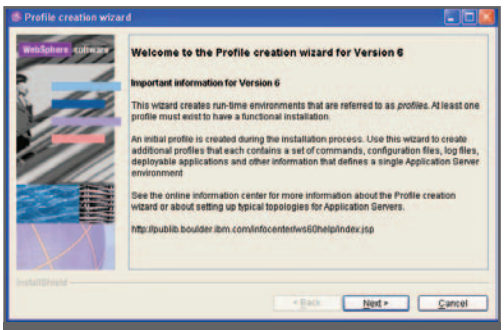


FIG 2: THE PROFILE CREATION WIZARD

1 will be created in the directory location that you specified. The default profile repository is <WAS_ROOT>/profiles/. Once the profile is created, you will need to start the application server. If you used the profile creation wizard, you will have the option of starting the FirstSteps application when the profile creation is complete. FirstSteps is an easy tool to start and test your profile. You can later rerun the FirstSteps tool by going into the firststeps directory of where your profile was created.

Another way of starting your application server using your new profile is via the command line. WebSphere provides two ways of starting the application server; it is with the same command but from two different locations. The first location is the <WAS_ROOT>/bin directory. Contained here is a helper script called startServer.cmd on Windows, and startServer.sh on Linux. When starting the application server from here, you will need to specify the profile to be started. For example, if your profile was called MyNewProfile, then the command would be:

LISTING 1

```
./wasprofile.sh \
-create –profileName MyNewProfile \
-profilePath /usr/IBM/WebSphere/
AppServer/profiles/MyNewProfile
-templatePath /usr/WebSphere/
AppServer/profileTemplates/default \
-nodeName leazar01node \
-cellName leazar01Cell \
-hostName leazar.raleigh.ibm.com
```

```
startServer.sh server1 -profile
MyNewProfile
```

server1 is the default server name given to Application Server when the profile was created. There are a number of other commands that are contained in this directory. If you have a specific profile that you want to change the state of, then you will need to specify the -profile command. The system administrator can use the commands in the <WAS_ROOT>/bin directory to control all of the profile instances. The commands in the <WAS_ROOT>/bin directory are intended for system administrators to execute.

The other location is within the bin directory of the profile you created. If you created the profile MyNewProfile, then the startServer.sh command would be located in <WAS_ROOT>/profiles/MyNewProfile/bin. The difference here is that all of the executables stored in this directory run in the context of the newly created profile. The -profile MyNewProfile is not required to start the application server.

The reason for the difference is that the profiles can be owned by a different group aside from the system administrator who is controlling the WAS. As the owner of MyNewProfile, I may have the need to start and stop the servers in my profile, but at the same time not have the authority to start and stop servers for another profile.

Managing Profiles

The wasprofile command also supports managing profiles as well as creating them. The description below contains some of the commands that the wasprofile provides.

Deleting a Profile

When you delete a profile, it removes the named profile and its contents from the directory. Any customization or additional scripts that you may have added to the profile directory may also be deleted as well. You should use this tool rather than deleting the profile directory manually since this will prevent corruption of the profile information that WAS maintains.

```
./wasprofile.sh –delete –profile-
Name profile_name | - profilePath
profile_path [-debug]
```

Listing Additional Profiles

To list the profiles that WAS knows about, use the following command:

```
./wasprofile.sh –listProfiles [-
debug]
```

To check the integrity of a profile registry:

```
./wasprofile.sh -validateRegistry
[-debug]
```

Get the path of an existing profile from the name:

```
./wasprofile.sh -getPath  -pro-
fileName profile_name  [-debug]
```

Conclusion

Profiles in WAS v6 offer flexibility to meet growing changes in environments. With profiles it is possible to create new application servers that have a degree of separation between other J2EE applications that might be running on the server. When you create a profile, you are creating a new instance of the WebSphere Application Server that shares the same runtime binaries, yet can have a unique configuration that is separate from another WebSphere instance. The separation has advantages for setting up and maintaining your WebSphere installation.

Resources

- Williamson, L., Chan, L., Cundiff, R., Lauzon, S., and Mitchell, C. (2005). *IBM WebSphere System Administration*. IBM Press.
- *WebSphere Application Server Information Center:* www-306.ibm.com/software/webservers/appserv/was/library/

Develop Web Services Clients

-continued from page 45

buttons, three textinputs, and one label component, and you must define every component with the instance name shown in Figure 8. Then save the Flash document (for our purposes, name it MathClient) and add the file to existing Flash project.

Now you need to create the MathAction class file to add the math operation behavior:

- Create new ActionScript file from File > New > ActionScript File.
- Save the ActionScript file as MathAction.as (remember, you need to use the same name as the class definition).
- Add the ActionScript File to Flash MX Project (see Figure 9).
- Enter the code shown in Listing 1 (Listings 1-4 can be downloaded from www.sys-con/websphere/sourcec.cfm).
- Open the action toolbar, click Frame 1, and enter the code shown in Listing 2.
- Click the Object button btn-Multiply and enter the code shown in Listing 3.

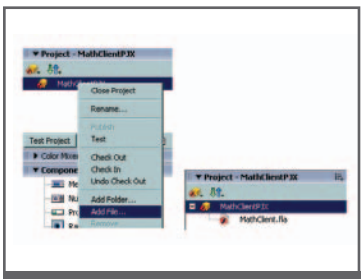


FIG 9: ADD MATHCLIENT FILE TO FLASH PROJECT

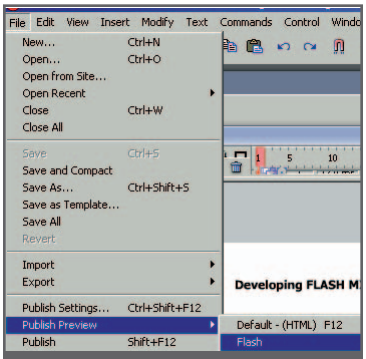


FIG 10: PUBLISH FLASH MX APPLICATION

btnMultiply will invoke the MathObject wrapper and call the Multiply method from Web service.

- Repeat this step for button btn-Add and btnSubtract, and enter the code shown in Listing 4.

Publishing WebSphere Web Services and Testing Flash MX Client Interactions with Web Services

Run Websphere v5.0 Test Environment. Click on menu File > Publish Preview > Flash to publish the Flash MX Application (see Figure 10). (For more information, visit www.macromedia.com.)

Summary

In this article, we built a Flash MX client for a simple math operation Web service that accepts two integer parameters and calculates the parameter to pass the result to the client. We also examined how to use an existing Web service, running on the application server provided within the WSDK, from within Macromedia Flash MX 2004 Professional. We then developed and tested a Flash MX 2004 client invoking the Web service and displaying the results in the Flash GUI application.

WebSphere Journal Coming Next Month...

The Support of Composite Applications in WebSphere Portal
Part 2

Energy and Utility Companies Heat Up Their Network Operations
With a newer, smarter approach

Please Read This Entire Article As Our Options Have Changed!

BY: BRIAN GARR

Being the good techie that I am, I did the majority of my holiday shopping on the Internet last month. That means when things go wrong as they rarely, but sometimes do, you have to turn to the telephone for customer service. That was fun. Using your index finger, you poke a series of digits and alternate between looking at the phone so you can see the numbers and listening for the appropriate responses until you either give up, or actually find the data you are looking for after much listening and punching. And, by the way, "please listen to this entire message because our options have changed." I simply don't believe that every contact center in the U.S. has recently changed their options!

So how about if we change the way we do customer service? How about if we make it easy for customers to get to the data they want, when they want it, over a plain old telephone? Are we really there yet?

Yes, in fact, we are. Skeptics beware, speech recognition works. There are robust systems out there today where you can simply ask your question using natural language, and get to the answers you are looking for. And as the technology has matured, standards have evolved and the technology has moved more into the mainstream of the IT center.

"Skeptics beware, speech recognition works… there are robust systems out there"

There are literally thousands of speech applications out there today based on the VoiceXML standard. Application development tools based on the Eclipse framework ease the learning curve for VoiceXML programmers. Reusable Dialog Components leverage the unique skills of the Voice User Interface expert across the development environment, and fit into the WebSphere programming model. MRCP obviates the need for proprietary APIs between IVRs and speech engines, preserving your investment in speech applications, even if you choose to change gateway vendors.

So what does this add up to, after all these years of the prognostication that next year will be "the year of speech"? Well, let's *not* talk about 2005 being the year of speech, but perhaps we can talk about 2005 as being the year that we make a big dent in "please listen to this entire message because our options have changed." Let's make 2005 about improving customer satisfaction by extending our enterprise to the telephone so customers can call the contact center and say "what is my balance?," or "there is an error in my statement," and get the help they need right away. The twin pressures of "reduced operating costs" and "improved customer service" don't have to be an either/or proposition. Conversational access is all about easy-to-use self-service. Speech recognition and speech synthesis (text-to-speech) are ready for prime time.

My New Year's resolution is to rid the world of "push one for ….". An ambitious resolution, I agree, and it may take a few years, but I am committed. The technology, the tools, and the skills are all there now for this shift to happen.

Brian Garr is the segment lead for the call center and voice portal segment for IBM pervasive computing. He is an evangelist and speaker worldwide on machine translation, text to speech, and speech recognition. He received the Smithsonian Institute's "Heros of Technology" designation in 1998 for his work in machine translation. bgarr@us.ibm.com

YOUR TESTING SUCKS.

1. Writing test code isn't testing – it's coding.
2. Your developers aren't making testable software.
3. It's impossible to get the business side involved in quality.
4. Your existing testing tools have become shelfware.
5. Your customers have become the primary testers.

LISA from iTKO.

Because everyone should own quality.

iTKO Try a free eval of LISA 2.5 at **www.itko.com**

We understand.

You've probably invested considerable time and money trying to automate your testing. But it just didn't work.

You were sold a tool that let you point and click to test on Day 1, and on Day 2 you were maintaining a test script - inserting bugs into your tests at the same rate as you inserted bugs into the code during development.

The tools were too hard for the QA teams to use, they required intense training and developer knowledge, and the seats were more expensive than a presidential fundraiser.

It's OK. We've seen this scenario at every customer we talk to.

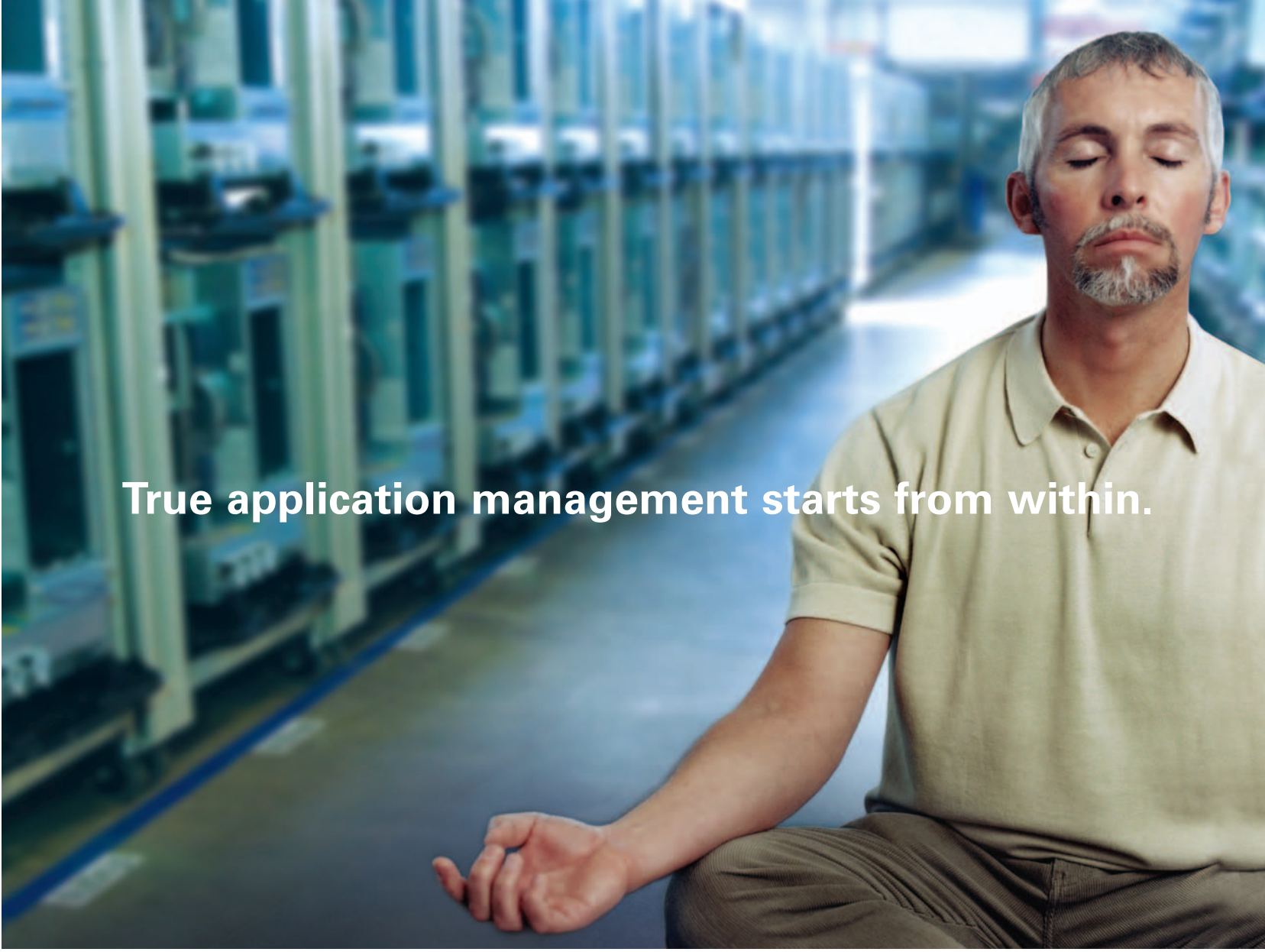
LISA is different.

LISA is a comprehensive, no-code automated testing solution ideal for distributed applications and teams.

Using "inline testing" technology, LISA allows even non-developers to instrument and test web services, J2EE apps, databases and websites.

LISA automates test design, staging and monitoring for unit, regression, load and performance tests. And since LISA is pure Java, she's equally stylish on Linux, OS X, Unix or Win clients.

But don't take this ad's word for it. Try LISA for free and talk to the experts at iTKO. We'll show you how everyone can own quality.



Motive brings an enlightened approach to application management, building management intelligence into the application itself. Only Motive transcends the fragmented, disjointed reality of traditional application management, to deliver the visibility, insight and automation that support and development teams need to keep applications running smoothly.

This is no mere vision for the future. It's here now in application management products from Motive. Learn more about Motive's breakthrough approach in a new white paper about built-in management at www.motive.com/within1.

motive

www.motive.com