SYS-CON

MEMORY • BANKING • PORTALS • STRATEGY

WebSphere® JOURNAL

www.SYS-CON.com/WebSphere

The World's Leading Independent WebSphere Resource

APRIL VOLUME 3 ISSUE 4

DISPLAY UNTIL JUNE 30, 2004

$8.99US $9.99CAN

04>

0 09281 03422 3

Migration: From Here to There to WebSphere

BY JUERGEN **EFEISH** & MAX **KING** PAGE 12

Movin' On Up

time to migrate to the next generation of WebSphere

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

Ask the Experts

Prolifics, a premier-level IBM Business Partner and 1 of 3 WebSphere Service Providers retained by IBM, specializes in WebSphere migrations.

Having conducted numerous migrations, both from earlier versions of WebSphere and from other J2EE application servers, such as BEA WebLogic, Oracle AS or Sun iPlanet, Prolifics has migration tools and a methodology that avoids unnecessary risk, time, effort and cost.

From J2EE App Servers to WebSphere

Prolifics conducts Migrations from other J2EE Application Server environments such as BEA WebLogic, Oracle AS, Sun iPlanet, etc.

From WebSphere v3.5/ v4 to v5

Prolifics migrates from earlier versions of WebSphere to IBM's latest release.

Taking You Step by Step

Prolifics offers MIGRATION ASSESSMENTS, PROOF-OF-CONCEPTS, and WEBPSHERE MENTORING to help you validate, plan, and execute a successful transition.

Prolifics

Winner of 2003 IBM BUSINESS PARTNER LEADERSHIP AWARD

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

Prolifics

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

diagnosys

Are your Web applications open 24 hours? Join Sun's expert John VanSant and H&W for a Web seminar series April 6 and June 22 to learn how to make your J2EE applications run at top speed. Register today at: www.hwcs.com/wsj4.asp

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

You need DiagnoSys.

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

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

© 2003-2004 H&W Computer Systems, Inc.
DiagnoSys is a trademark of H&W Computer Systems, Inc.

H&W
www.hwcs.com | 1.800.338.6692

Bridging the Enterprise Computing Gap For 25 Years

Sarbanes-Oxley: The New Rising Star

BY JACK MARTIN

Ineffectual corporate management has given a great gift to programmers, system administrators, and CIOs – endless corporate accounting scandals. Our federal government has not missed this scandalous behavior, as they have passed an extraordinarily strong, far-reaching law to contend with financial fraud.

Officially it's called the Public Company Accounting Reform and Investor Protection Act, but it's more commonly known as the Sarbanes-Oxley Act. Designed to keep corporate managers honest, the Sarbanes-Oxley Act rewards dishonesty with a 10- or 20-year prison sentence for CEOs and CFOs. It also has a provision for taking any and all ill-gotten gains from a dishonest executive. There are approximately 14,000 publicly traded companies in the U.S.

Recently, the Securities and Exchange Commission (SEC) charged Jeffrey Skilling, the former president and CEO of Enron, with fraud. The SEC is seeking to seize all of his ill-gotten gains and permanently bar him from acting as a director or officer of any publicly held company. On top of that, he is facing a maximum of 325 years in prison and hundreds of millions of dollars in fines. Considering that Andrew Fastow, who reported to Skilling at Enron, got off with 10 years in prison and forfeited $23 million in cash and assets, this is serious stuff!

Executives are now directly responsible for establishing and maintaining an adequate internal control structure and procedures for financial reporting. The reports must contain a written assessment of the effectiveness of the internal control structure and procedures of the issuer with regard to financial reporting, as of the end of the most recent fiscal year of the issuer.

To put this in perspective, the average billion-dollar company has about 50 disparate financial systems running at any given time, some of which have been running for more than 30 years. Remember Y2K, with all of those old systems designed in the 50 years after World War II? The designers didn't anticipate ever needing to factor in the century change. Starting June 15, 2004, you need to ensure that everything coming out of the old mainframes, client/servers, and new application servers is correct and adds up perfectly.

Adding to the complexity of the corporate information-technology topology is the fact that most of them are running two or three enterprise resource planning (ERP) systems. These take years to install and configure – and no one can guarantee that any ERP system is 100% on the money 100% of the time.

The icing on the cake is that about 50% of the time this information is output to an Excel spreadsheet, opening the door for endless misinterpretations and mistakes.

Sarbanes-Oxley compliance is very different from Year 2000 readiness. With the Y2K fire drill all you needed to do was get the computer to roll over on January 1, 2000, and not shut off or miscalculate. Y2K was a one-time event – and there was no Y2.1K. Sarbanes-Oxley compliance is an every-day, every-hour issue that must be rolled up into a tight, neat package every three months to support the quarterly financial statements.

For a company to comply with Sarbanes-Oxley, the accounting, financial management, and legal departments all ultimately funnel their data through information technology, which opens issues as to the completeness and accuracy of every code fragment and algorithm in every project.

White-shoe law firms, the large accounting firms, and a myriad of management consultants have all begun Sarbanes-Oxley practices.

–continued on page 49

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

WebSphere® JOURNAL

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

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

WRITERS IN THIS ISSUE
Kulivr Singh Bhogal, Buell Duncan, Juergen Efeish, Karla Feuer, Lloyd Hagemo, Louis J. Iacona, Ravi Kalidindi, Max King, Kwang Sik Kang, Bhadri Madapusi, Jack Martin, Jeff Schaffzin, Steve Stover, Andrew Sweet

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

PRESIDENT AND CEO FUAT KIRCAALI
PRESIDENT, SYS-CON EVENTS GRISHA DAVIDA
GROUP PUBLISHER JEREMY GEELAN
SENIOR VP, SALES & MARKETING CARMEN GONZALEZ
VP, INFORMATION SYSTEMS ROBERT DIAMOND
VP, SALES & MARKETING MILES SILVERMAN
PRODUCTION CONSULTANT JIM MORGAN
FINANCIAL ANALYST JOAN LAROSE
ACCOUNTS RECEIVABLE CHARLOTTE LOPEZ
ACCOUNTS PAYABLE BETTY WHITE
ADVERTISING DIRECTOR ROBYN FORMA
DIRECTOR OF SALES AND MARKETING MEGAN MUSSA
ASSOCIATE SALES MANAGERS KRISTIN KUHNLE
BETH JONES
CONFERENCE MANAGER LIN GOETZ
LEAD DESIGNER LOUIS F. CUFFARI
ART DIRECTOR ALEX BOTERO
ASSOCIATE ART DIRECTORS RICHARD SILVERBERG
TAMI BEATTY
WEB DESIGNERS STEPHEN KILMURRAY
CHRISTOPHER CROCE
CONTENT EDITOR LIN GOETZ
CIRCULATION SERVICE COORDINATORS SHELIA DICKERSON
EDNA EARLE RUSSELL
LINDA LIPTON

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

WORLDWIDE NEWSSTAND DISTRIBUTION
CURTIS CIRCULATION COMPANY, NEW MILFORD, NJ

FOR LIST RENTAL INFORMATION:
KEVIN COLLOPY: 845 731-2684, KEVIN.COLLOPY@EDITHROMAN.COM
FRANK CIPOLLA: 845 731-3832, FRANK.CIPOLLA@EPOSTDIRECT.COM
© COPYRIGHT 2004 BY SYS-CON PUBLICATIONS, INC. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPY OR ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM, WITHOUT WRITTEN PERMISSION. FOR PROMOTIONAL REPRINTS, CONTACT REPRINT COORDINATOR, CARRIE@SYS-CON.COM. SYS-CON PUBLICATIONS, INC. RESERVES THE RIGHT TO REVISE, REPUBLISH AND AUTHORIZE THE READERS TO USE THE ARTICLES SUBMITTED FOR PUBLICATION.

ALL BRAND AND PRODUCT NAMES USED ON THESE PAGES ARE TRADE NAMES, SERVICE MARKS OR TRADEMARKS OF THEIR RESPECTIVE COMPANIES. SYS-CON PUBLICATIONS, INC., IS NOT AFFILIATED WITH THE COMPANIES OR PRODUCTS COVERED IN WEBSPHERE JOURNAL.

WEBSPHERE® IS A REGISTERED TRADEMARK OF IBM CORP. AND IS USED BY SYS-CON MEDIA UNDER LICENSE. WEBSPHERE® JOURNAL IS PUBLISHED INDEPENDENTLY OF IBM CORP., WHICH IS NOT RESPONSIBLE IN ANY WAY FOR ITS CONTENT.

©2003 Wily Technology, Inc. The Wily logo is a trademark of Wily Technology, Inc. Java is a trademark of Sun Microsystems in the U.S. and other countries.

Two years without a vacation.
The application's up. It's down.
It's up. It's down.

I'm to blame. Steve's to blame.
Someone's always to blame.

Not any more.

Get Wily.™

Enterprise Java
Application Management
1 888 GET WILY
www.wilytech.com

wily technology

IBM's logging and tracing facility for WAS

Keeping in the Know with JRas

BY KULVIR SINGH **BHOGAL**

& KWANG SIK **KANG**

JRas is a logging and tracing facility built into IBM WebSphere Application Server (WAS) that relies on JRas as its internal logging framework. Programmers can also leverage this powerful logging system infrastructure to keep an eye on their enterprise applications powered by WAS.

Developers can use the JRas Java APIs to generate both log and trace messages. Used properly, these two mechanisms can provide priceless information regarding the application's process of execution.

In this article, the reader will learn to use JRas through the context of a sample application. The provided source code will allow developers to immediately begin taking advantage of the JRas facility and start considering it for their own enterprise applications.

Why the Fuss about Tracing and Logging?

In a Utopian world, your applications would always work and there would be no need to monitor them, but the reality is that you don't live in Utopia. Your enterprise applications demand attention. You might leverage JRas to provide simple status messages to keep WebSphere administrators abreast of the execution of an enterprise application. At the same time, JRas diagnostic trace (trace) statements might be embedded within your code, which can be extremely helpful to diagnose and remedy problems should they arise with your application.

Differentiating Between Messages and Traces

In JRas terminology, messages consist of information that is brief, clear, and useful to an end user. A simple example might be a status message saying an application was started successfully. Message entries are informational records intended for end users, system administrators, and support personnel.

On the other hand, trace entries contain more detailed information that might include the timestamp, execution thread ID, execution method, and a description of an error. Typically, development teams and service engineers use this detailed information to dissect and troubleshoot an application. Enabling the trace facility is a costly operation in terms of taxing the system resources and should only be performed to troubleshoot the system.

Terminology

Before beginning our logging endeavors, let's review some terminology critical to understanding JRas.

- ***Event classes:*** JRas defines message and trace event classes.
- ***Event types:*** JRas ships with predefined event types for messages and tracing. For example, some message types include information, warning, and error messages. Some trace types include entry, exit, and trace.
- ***Loggers:*** These represent the workhorse of JRas. JRas defines message and trace loggers. As their names imply, message loggers create message records, while trace loggers are responsible for creating trace records.
- ***Message handlers:*** Loggers use registered handlers to output their events. For example, a file handler might be used to write an event to a file.
- ***Formatters:*** Formatters can be used with JRas handlers to format data destined to an output device. In this article, the default formatting will be used. To learn more about formatting, view the information provided in the WebSphere Application Server Information Center Reference Library.
- ***Message masks:*** Masks are configured for a logger to process only a subset of messages. For example, a mask can be configured for a logger to process only error messages and ignore informational/warning messages.

A Lab Rat

To demonstrate the usage of JRas, a simple stateless session Enterprise JavaBean (EJB) will be used. Download the provided EAR artifact and import it into your WebSphere Studio workspace or deploy it to

WAS. The logging uage is demonstrated in a business method of the JRasEJB. This method, as shown here, capitalizes an incoming String and prepends the following text: "Right back at you:"

```
public String echoMe(String argu-
ment)
{
  String whatToReturn = "Right
  back at you: " + argument;
  whatToReturn = whatToReturn.
  toUpperCase();
  return (whatToReturn);

}
```

The full implementation of the JRasEJBBean without the logging incorporated is shown in Listing 1.

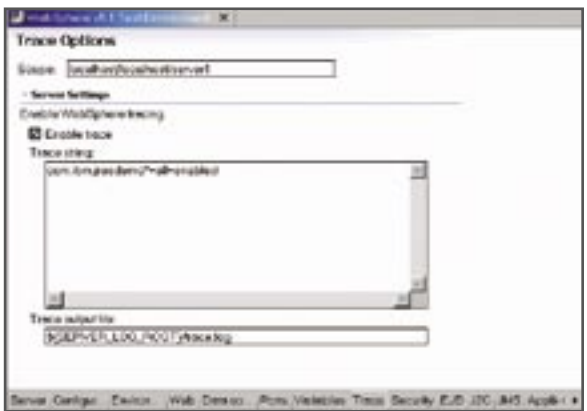


FIG 1: ENABLING TRACING IN WEBSPHERE STUDIO

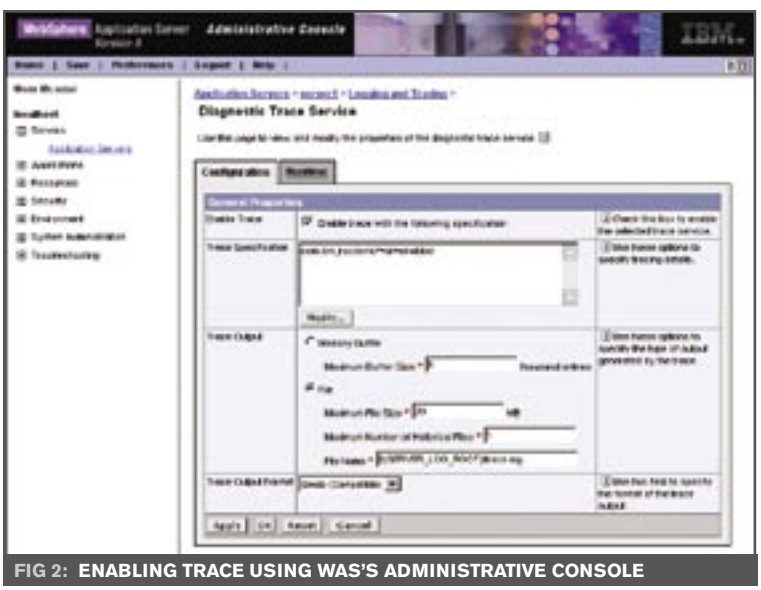


FIG 2: ENABLING TRACE USING WAS'S ADMINISTRATIVE CONSOLE

Let's Do Some Logging!

The process of incorporating JRas logging into the application is rather straightforward.

1. Retrieve a reference to the JRas manager.
2. From the returned manager, retrieve message and/or trace loggers.
3. Call methods on the message and trace loggers to create message and trace entries.

This exact process will be implemented into the EJB. Review Listing 2, as it shows the final implementation.

Our steps of retrieving a reference to the JRas manager as well as retrieving message and trace loggers were accomplished with the code shown in Listing 3.

The syntax is straightforward; note that that following arguments are provided to create the logger and tracer:

1. Name of our organization
2. Application name
3. Component identifier
4. Class name

The trace and message loggers are static final member variables allowing for a single initialization phase at instantiation.

Now modify the echoMe method to leverage the logger and tracer using the code shown in Listing 4.

The trace file will contain entries of method entrances and exits. This is facilitated through the use of entry and exit methods of the RASTraceLogger object, along with the predefined event type of RASITraceEvent.TYPE_ENTRY_EXIT. There are several other available RASITraceEvent enumerated types. Refer to the WebSphere Application Server Information Center for the full breakdown of available types. Note that trace events will appear only in the server's trace.log file, discussed later in this article.

Examine the source code and notice that the RASMessageLogger's textMessage method is used periodically to post messages to the console. Different RASIMessageEvent types, namely RASIMessageEvent.TYPE_INFO, RASIMessageEvent.TYPE_WARNING, and RASIMessageEvent.TYPE_ERROR, are used in the provided implementation. These different types are displayed differently on the console, as will be shown later.

Enabling Logging in Application Developer and WebSphere Application Server

To reap the benefits of the logging implementation, WebSphere Studio's trace facility must be enabled. Accomplish this by choosing the Trace tab in the Server Configuration view of the server housing the enterprise application. Ensure that "Enable Trace" is selected. By default, the String text box will have the value:

```
*=all=disabled
```

Modify this entry with the appropriate package hierarchy. For this

Kulvir Singh Bhogal works as an IBM Software Services for WebSphere consultant, devising and implementing WebSphere-centric solutions at customer sites across the nation. He has over 50 patents pending in a myriad of technology areas.
kbhogal@us.ibm.com

Kwang Sik Kang, consultant for IBM's Software Services for WebSphere, has extensive experience in providing J2EE and messaging-based enterprise solutions.
kkwang@us.ibm.com

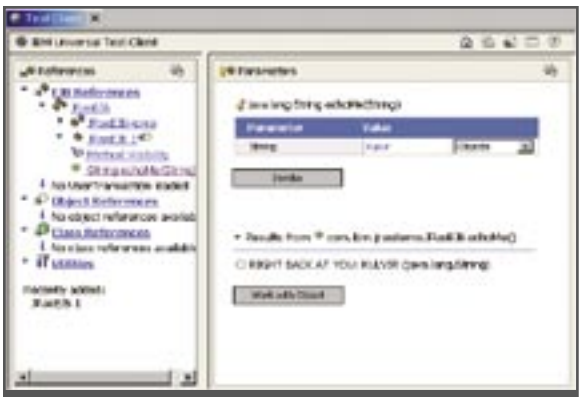


FIG 3: TESTING THE ECHOME EJB METHOD

exercise, add the following entry:

```
com.ibm.jrasdemo*=all=enabled
```

The modification is shown in Figure 1.

The tracing facility in WAS can be enabled through the Administrative Console. Select the appropriate application server instance housing the enterprise application and enable the tracing facility from the Configuration section. Choose "Logging and Tracing Option" and "Diagnostic Trace" (see Figure 2).

Test the business method of the sample EJB, echoMe, using WebSphere Studio's Universal Test Client. The successful execution is shown in Figure 3.

Understanding What Happened in Our Console

Running the EJB will output to the console messages we incorporated into the code (see Figure 4).

In Figure 4, note that the event types are signified by the "A", "W", and "E" notations displayed to the left of the class name. These stand for Audit, Warning, and Error respectively. In the JRas realm, the TYPE_INFO maps to a WAS platform native type of "Audit". Similarly, the TYPE_WARNING maps to the native type of "Warning", and TYPE_ERROR maps to the native type "Error". Take note that the trace messages don't appear in our console and appear only in the trace file, which we will now analyze.

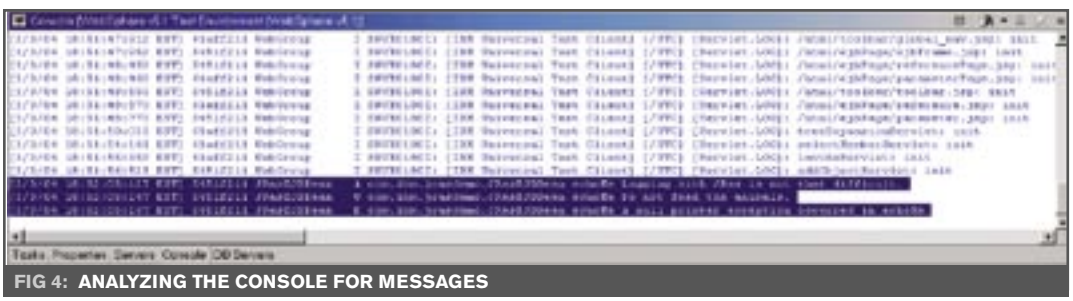


FIG 4: ANALYZING THE CONSOLE FOR MESSAGES

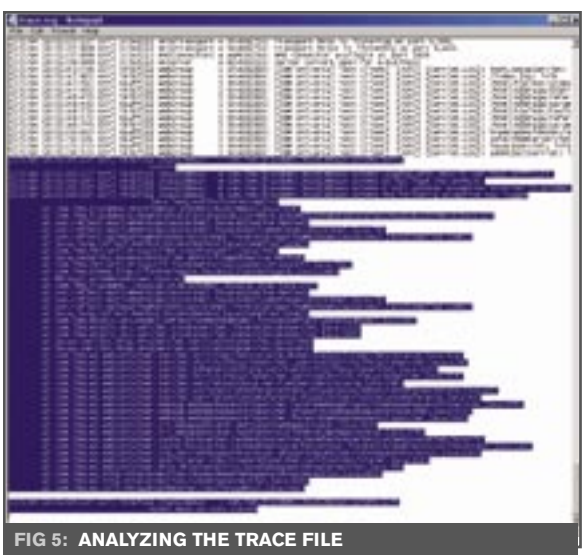


FIG 5: ANALYZING THE TRACE FILE

Understanding What Happened in Our Trace File

Open the trace.log file and view the detailed information provided regarding the application's execution (see Figure 5).

Let's point out some interesting information about our trace file.

- Method entry and exit points are signified by ">" and "<" respectively, noted to the left of the fully qualified class name. Note that the incoming (i.e., Kulvir) and outgoing (i.e., RIGHT BACK AT YOU KULVIR) values are also displayed.
- Note that both message and trace information can be found in the trace file.
- The exception, hidden from the console, is vividly described in the form of a stack trace. End users may provide custom error codes seen in the console that map to more detailed information found in the trace file to diagnose application problems.

Performance Considerations

Enabling the use of JRas within the application code is a good technique for diagnosing an enterprise application. However, this benefit does not come without cost. Enabling the trace facility requires extensive disk I/O processing, potentially affecting the application's performance.

Conclusion

In this article, we introduced the JRas logging facility. Other popular logging systems such as Apache Log4J and Java 1.4 Java Logging API have a large industry following. However, JRas offers a tight integration with WAS's system management, in particular with the Administrative Console. As shown in this article, JRas integrates with both WebSphere Studio and WAS. Enterprises can leverage JRas to monitor applications in an extremely powerful fashion.

References

- *Using the JRas Message Logging and Trace Facility:* http://www-306.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter/was/pdf/atswpj00.pdf
- *WebSphere Application Server Information Center Reference Library:* www-306.ibm.com/software/webservers/appserv/infocenter.html

< Know More. No Less. >

Knowledge.
Experience.
Confidence.
From Our Team to Yours

TRAINING

Innovative,
custom-tailored Courses
to maximize your ROI

- Object Oriented Analysis & Design
- Requirements, Modeling and MDA™
- Java™, J2EE
- WebSphere™ & WebLogic™
- .NET™
- XML and Web Services

- Senior, real-world architects & developers
- A+ educators with advanced degrees
- Fortune 1000 customers
- 30,000+ technical professionals trained

MENTORING

Building confidence,
delivering results

CONSULTING

Technical mastery.
Commitment to excellence.
At work for you.

- Enterprise Architecture
- Application development
- Software Migration
- Application Integration
- Technology Assessment
- Team & Process Assessment

Know More.
www.inferdata.com for a 10% discount!

InferData

LISTING 1: THE JRASEJBBEAN WITHOUT LOGGING INCORPORATED

```
package com.ibm.jrasdemo;
import com.ibm.ras.RASIMessageEvent;
import com.ibm.ras.RASITraceEvent;
import com.ibm.ras.RASMessageLogger;
import com.ibm.ras.RASTraceLogger;
import com.ibm.websphere.ras.Manager;
/**
 * Bean implementation class for Enterprise Bean: JRasEJB
 */
public class JRasEJBBean
   implements javax.ejb.SessionBean
{
   private javax
          .ejb
          .SessionContext mySessionCtx;
   /**
    * getSessionContext
    */
   public javax
          .ejb
          .SessionContext getSessionContext()
   {
          return mySessionCtx;
   }
   /**
    * setSessionContext
    */
   public void setSessionContext(
          javax.ejb.SessionContext ctx)
   {
          mySessionCtx = ctx;
   }
   /**
    * ejbCreate
    */
   public void ejbCreate()
          throws javax.ejb.CreateException
   {
   }
   /**
    * ejbActivate
    */
   public void ejbActivate()
   {
   }
   /**
    * ejbPassivate
    */
   public void ejbPassivate()
   {
   }
   /**
    * ejbRemove
    */
   public void ejbRemove()
   {
   }
   public String echoMe(String argument)
   {
          String whatToReturn =
                    "Right back at you: "
                              + argument;
          whatToReturn =
                    whatToReturn.toUpperCase();
          return (whatToReturn);
   }
}
```

LISTING 2: JRASEJBBEAN WITH LOGGING INCORPORATED

```
package com.ibm.jrasdemo;
import com.ibm.ras.RASIMessageEvent;
import com.ibm.ras.RASITraceEvent;
import com.ibm.ras.RASMessageLogger;
import com.ibm.ras.RASTraceLogger;
import com.ibm.websphere.ras.Manager;
/**
 * Bean implementation class for Enterprise Bean: JRasEJB
 */
public class JRasEJBBean
   implements javax.ejb.SessionBean
{
   static final RASTraceLogger tracer;
   static final RASMessageLogger logger;
   static {
          Manager manager =
                    Manager.getManager();
          tracer =
                    manager
                              .createRASTraceLogger(
                              "YourCompanyNameWouldGoHere",
                              "YourSampleApplicationName",
                              "YourComponentIdentifier",
                              JRasEJBBean
                                        .class
                                        .getName());
          manager.addLoggerToGroup(
                    tracer,
                    "ALoggingGroup");
          logger =
                    manager
                              .createRASMessageLogger(
                              "YourCompanyNameWouldGoHere",
                              "YourSampleApplicationName",
                              "YourComponentIdentifier",
                              JRasEJBBean
                                        .class
                                        .getName());
   }
   private javax
          .ejb
          .SessionContext mySessionCtx;
   /**
    * getSessionContext
    */
   public javax
          .ejb
          .SessionContext getSessionContext()
   {
          return mySessionCtx;
   }
   /**
    * setSessionContext
    */
   public void setSessionContext(
          javax.ejb.SessionContext ctx)
   {
          mySessionCtx = ctx;
   }
   /**
    * ejbCreate
    */
   public void ejbCreate()
          throws javax.ejb.CreateException
   {
   }
   /**
    * ejbActivate
    */
   public void ejbActivate()
   {
   }
   /**
    * ejbPassivate
    */
   public void ejbPassivate()
   {
   }
```

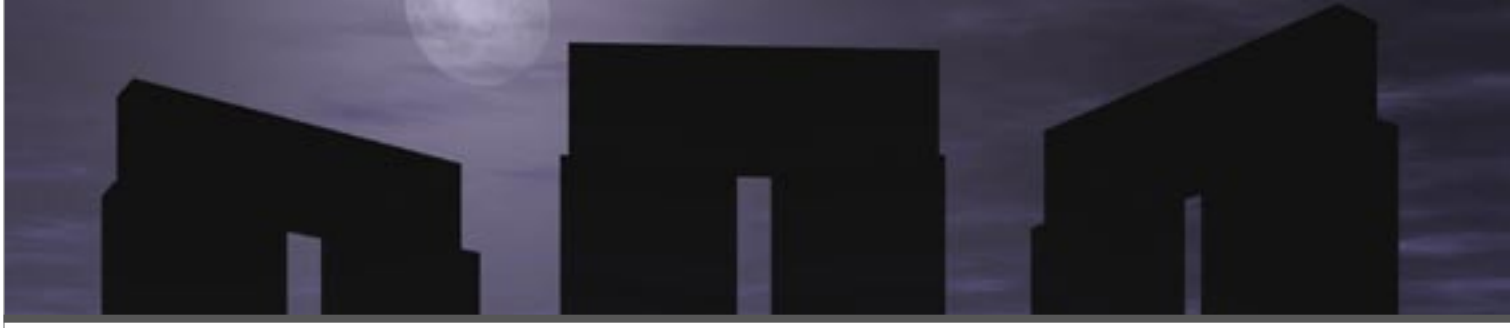
```
   /**
    * ejbRemove
    */
   public void ejbRemove()
   {
   }
   public String echoMe(String argument)
   {
          tracer.entry(
                    RASITraceEvent
                              .TYPE_ENTRY_EXIT,
                    this,
                    "echoMe",
                    argument);
          logger.textMessage(
                    RASIMessageEvent.TYPE_INFO,
                    this,
                    "echoMe",
                    "Logging with JRas is not that diffi-
cult.");
          String whatToReturn =
                    "Right back at you: "
                              + argument;
          logger.textMessage(
                    RASIMessageEvent
                              .TYPE_WARNING,
                    this,
                    "echoMe",
                    "Do not feed the animals.");
          whatToReturn =
                    whatToReturn.toUpperCase();
          String nullString = null;
          try
          {
                    // force an exception
                    int causeAnError =
                              nullString.length();
          }
          catch (Exception e)
          {
                    logger.textMessage(
                              RASIMessageEvent
                                        .TYPE_ERROR,
                              this,
                              "echoMe",
                              "A null pointer exception
occurred in echoMe.");
                    tracer.exception(
                              RASITraceEvent
                                        .TYPE_ERROR_EXC,
                              this,
                              "echoMe",
                              e);
          }
          tracer.exit(
                    RASITraceEvent
                              .TYPE_ENTRY_EXIT,
                    this,
                    "echoMe",
                    whatToReturn);
          return (whatToReturn);
   }
}
```

LISTING 3: RETRIEVING A REFERENCE TO THE JRAS MANAGER/RETRIEVING MESSAGE AND TRACE LOGGERS

```
static final RASTraceLogger tracer;
static final RASMessageLogger logger;
static {
   Manager manager =
          Manager.getManager();
tracer =
   manager
   .createRASTraceLogger(
   "YourCompanyNameWouldGoHere",
   "YourSampleApplicationName",
   "YourComponentIdentifier",
   JRasEJBBean.class.getName());
   manager.addLoggerToGroup(
          tracer, "ALoggingGroup");
logger =
   manager
   .createRASMessageLogger(
   "YourCompanyNameWouldGoHere",
   "YourSampleApplicationName",
   "YourComponentIdentifier",
   JRasEJBBean.class.getName());
}
```

LISTING 4: MODIFYING THE ECHOME METHOD

```
public String echoMe(String argument)
   {
          tracer.entry(
                    RASITraceEvent
                              .TYPE_ENTRY_EXIT,
                    this,
                    "echoMe",
                    argument);
          logger.textMessage(
                    RASIMessageEvent.TYPE_INFO,
                    this,
                    "echoMe",
                    "Logging with JRas is not that diffi-
cult.");
          String whatToReturn =
                    "Right back at you: "
                              + argument;
          logger.textMessage(
                    RASIMessageEvent
                              .TYPE_WARNING,
                    this,
                    "echoMe",
                    "Do not feed the animals.");
          whatToReturn =
                    whatToReturn.toUpperCase();
          String nullString = null;
          try
          {
                    // force an exception
                    int causeAnError =
                              nullString.length();
          }
          catch (Exception e)
          {
                    logger.textMessage(
                              RASIMessageEvent
                                        .TYPE_ERROR,
                              this,
                              "echoMe",
                              "A null pointer exception
occurred in echoMe.");
                    tracer.exception(
                              RASITraceEvent
                                        .TYPE_ERROR_EXC,
                              this,
                              "echoMe",
                              e);
          }
          tracer.exit(
                    RASITraceEvent
                              .TYPE_ENTRY_EXIT,
                    this,
                    "echoMe",
                    whatToReturn);
          return (whatToReturn);
   }
```



Migration: From Here to There to WebSphere

Migrating to WebSphere v5 from earlier versions and from other J2EE application servers

BY JUERGEN **EFEISH**

& MAX **KING**

Why migrate to WebSphere v5? Whether you are currently using WebSphere v3.5 or v4, or are using a different J2EE application server altogether, there are many reasons that justify the move.

First there is the corporate choice – when choosing WebSphere you are choosing IBM. This means that you benefit from their reputation for superior support and their rather vast portfolio of software options. You also benefit from the level of investment they are putting into WebSphere, which equals more than the annual revenues of some of their competitors.

Next there is the technology choice. Certainly, by moving to the latest and greatest, you can take advantage of all its advancements and new features, which should match your growing business and technical needs. Additionally, WebSphere probably averages out to a lower cost of ownership than most of its competitors. Compared to similarly priced options such as WebLogic, it offers lower software pricing, and more dramatically, it offers lower maintenance costs. Compared to the freeware or shareware market, WebSphere has a more expensive purchase price, but with all its complementary tooling it actually incurs lower development and deployment costs and provides easier extensions to portals and pervasive devices, as well as offering more reliable support.

Finally, there are the business implications. End-of-support dates or end-of-life dates are significant driving factors for any company running critical business systems – support for WebSphere v3.5 ended in November 2003, and support for WebSphere v4 will end in September 2004. In fact, even if you are on a different J2EE application server such as iPlanet, WebLogic, or Oracle AS, an end-of-support date may be the prime time to make the move to WebSphere instead of performing a version upgrade, due to the similar levels of effort. You may think that your application is running fine, so why change? Well, you don't want to be held back, prevented from upgrading your hardware or buying new software products because they aren't supported by earlier versions of the application server.

Migration Strategy

Each migration has its own nuances, so we are not trying to detail a "walk-through" migration guide. After all there is no "one-size-fits-all" when it comes to migration. Instead, we are trying to give you some general information to guide you and help you get a feel for the effort.

A STARTER STRATEGY – JUST TRY IT!

First, go for it – take a few code samples and get them running in WebSphere v5. Second, see if your application will compile, build, and deploy to WebSphere out of the box. It won't – but you're on the right track. Third, add some method to the madness: devise a migration methodology and consult the best practices for migrating.

We have been refining our own methodology on migration for several of our customers. In this article we hope to share some of the expert advice that we have gathered along the way.

A HIGH-LEVEL STRATEGY

After many migrations we have come up with a list of steps that have proven to be very useful, no matter which application server you are migrating from. This list also considers that when we do a migration we usually deal with applications that have been in use for a while and that have been patched, fixed, and updated. Many developers have had their hands on the code, documenting it or not, so over time the application's code documentation has become outdated. This means that when we migrate an application we also have the chance to take care of all of this.

1. ***Verify baseline:*** Confirm the application behavior on the original system (prevent the addressing of migration problems that never existed in the first place). A benefit, this also progressively builds the future documentation of the system.
2. ***Stage source code:*** Take a snapshot of the current production code – version control your code base.
3. ***Import source code:*** WebSphere Studio Application Developer v5 provides ready templates for the J2EE

1.2 and J2EE 1.3 project structure. We find it beneficial to use WebSphere Studio Application Developer v5 as early as possible in the migration steps.

4. ***Correct compilation errors:*** Based on the imported source code WebSphere Studio Application Developer v5 creates an analysis report on the fly (Task List) listing the Java compilation errors, problems in the Web layer of the application, and missing J2EE 1.3 or J2EE 1.2 requirements. (This includes JSPs and HTML pages.)
5. ***Complete deployment descriptors:*** WebSphere Studio Application Developer v5 plays an active role in this part and provides the developer with deployment descriptor editors and wizards to ensure J2EE-compliant applications.
6. ***Test and debug:*** Prepare the WebSphere Studio Application Developer v5 internal WebSphere Application Server v5 Test environment, including data source setup. Conduct a full application test to determine eventual runtime errors as well as database connectivity problems. These are steps that must be completed before deploying an application EAR file into a QA environment. Make full use of WebSphere Studio Application Developer test environment and debugging capabilities.
7. ***Deployment strategy:*** Use WebSphere Studio ANT support to create a deployment EAR file for testing in a WebSphere QA architecture.

Migration Tooling

We recommended WebSphere Studio Application Developer as your primary migration tool, but here is a list of other WebSphere migration tools. Note that most of the functionality in these tools is already embedded in WebSphere Studio Application Developer.

WEBSPHERE V4 UPGRADE TOOLS

- ***WASPreUpgrade:*** To save the configuration from a prior version of WebSphere Application Server
- ***WASPostUpgrade:*** To convert the configuration saved by WASPreUpgrade into a WebSphere Application Server v4 configuration and application migration

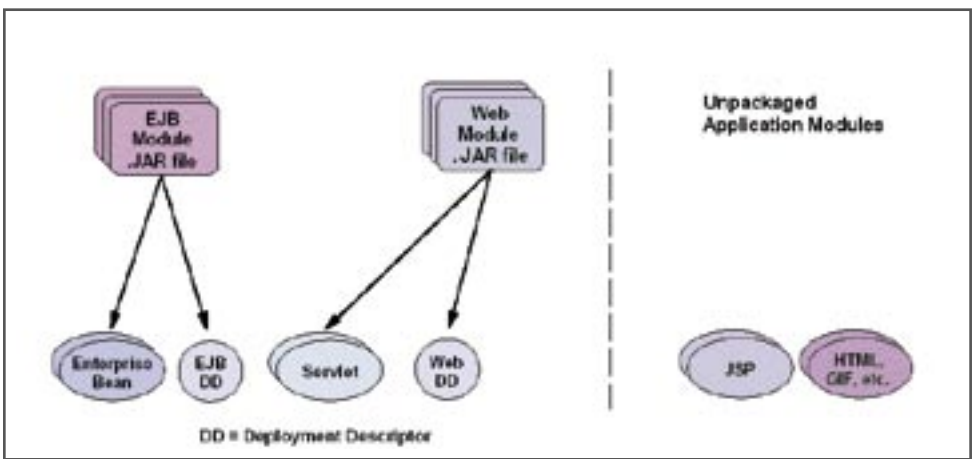


FIG 1: WEBSPHERE V3.5 APPLICATION PACKAGING

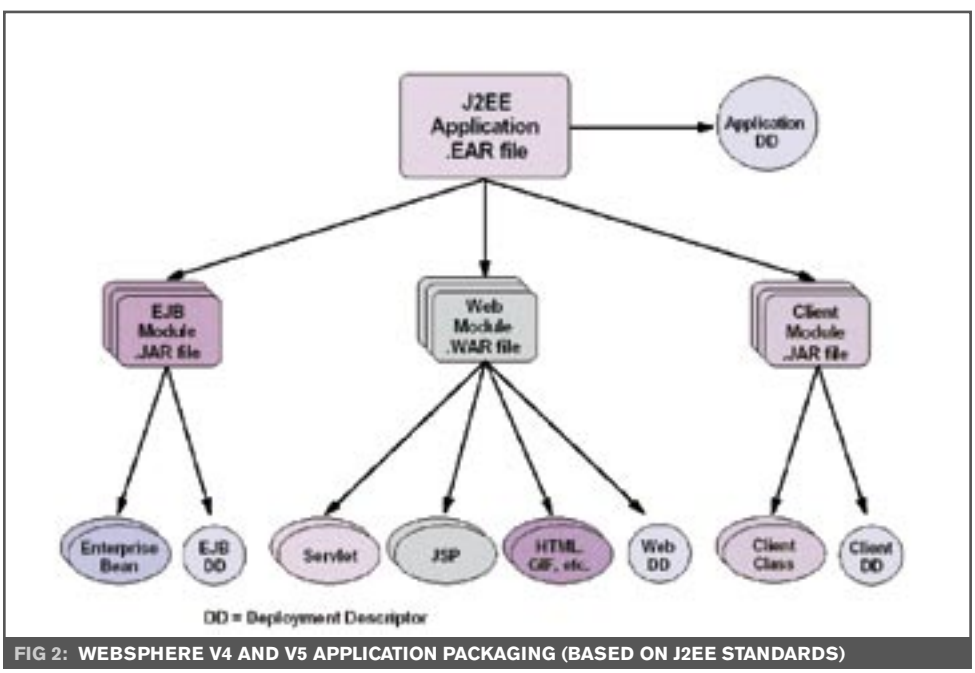


FIG 2: WEBSPHERE V4 AND V5 APPLICATION PACKAGING (BASED ON J2EE STANDARDS)

WEBSPHERE V5 UPGRADE TOOLS

- ***ClientUpgrade:*** For upgrade of J2EE client JARs (primarily a v4-to-v5 tool)
- ***WASPreUpgrade:*** To save the configuration from a prior version of WebSphere Application Server
- ***WASPostUpgrade:*** To convert the configuration saved by WASPreUpgrade into a WebSphere Application Server v5 configuration and application migration
- ***MigrateWC:*** For JSP and servlet source code conversion
- ***Ejbdeploy:*** For EJB 1.0-to-1.1 conversion
- ***Earconvert:*** To convert a J2EE 1.2 EAR file to a J2EE 1.3 EAR file
- ***mb2mdb:*** WebSphere Enterprise Edition 4.0 JMS listener application for using message-driven beans
- ***CACT:*** Class API checker tool; checks for supported API levels in applications

Expert Advice on Migration from WebSphere v3.5 or v4 to WebSphere v5

SELECTING TARGET VERSIONS

If you are on WebSphere v3.5, and are deciding which WebSphere version to migrate to, we recommend the latest (currently v5.1). Table 1 shows new features in versions 4 and 5. You should migrate to version 4 only if third-party software used in your application is not supported by version 5 – otherwise you just end up with more work, time, and money spent. Table 2, which identifies the key factors involved in different migrations, demonstrates that migrating from version 3.5 includes much the same factors, whether you're migrating to version 4 or version 5.

If you are on WebSphere v4, the migration to version 5 is even easier because version 4 applications are using the J2EE packaging structure. Remember, WebSphere v5 is backwards-compatible, so you can already run your version 4 application in a WebSphere v5 environment (using WebSphere v4 data sources).

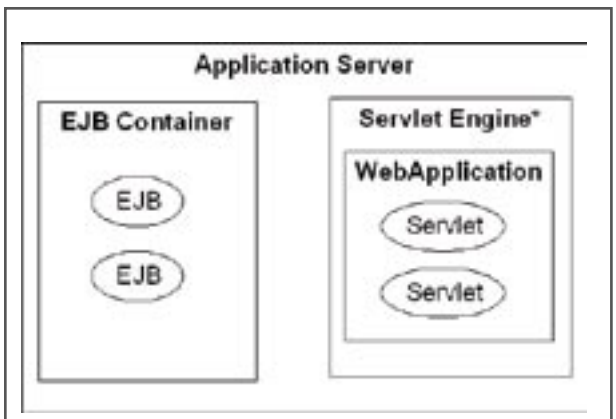


FIG 3: WEBSPHERE APPLICATION SERVER V3.5 ARCHITECTURE (SIMPLIFIED VIEW)

J2EE COMPLIANCE AND PACKAGING

One of the most important differences between WebSphere v3.5 and later versions of WebSphere is J2EE compliance. Figures 1 and 2 show the difference between WebSphere v3.5 application packaging and the J2EE-compliant WebSphere v4 and WebSphere v5 application packaging. Based on these figures you can see how we have to package the applications to comply with the J2EE standard.

We need to adapt the EAR file for the deployment environment (see Figures 3 and 4 for the architecture comparison). Note that in Figure 3 the servlet engine is a Web container, but since WebSphere v3.5 did not really support the concept of a WAR file, we refer to it as a servlet engine.

Expert Advice on Migration from Another J2EE Application Server to WebSphere

GENERAL ADVICE

In an application server migration you need to move not only the application and its code to the target environment, but also the developers, users, third-party software components, and systems. As part of the migration effort it is important to allocate time for testing the application's integration with other software. Additionally, we recommend that you invest in some training and mentoring of the new environment for your development team so that they are best equipped to maintain the application.

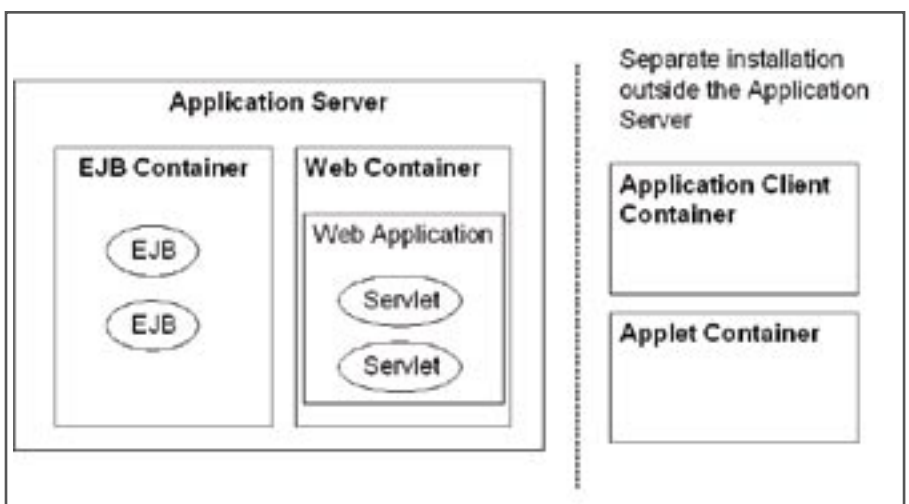


FIG 4: WEBSPHERE APPLICATION SERVER V4/V5 ARCHITECTURE (SIMPLIFIED VIEW)

If your organization is an ISV that sells a solution in the marketplace you may need your application to run in multiple application server environments for greater market reach, yet maintain a common code base and elaborate build processes. Prolifics has done this for several customers.

Often in an application server migration organiza-

tions choose to perform a proof of concept in which they conduct a sample migration. The migration is conducted on an identified representative vertical slice of the existing application that exercises the application's core features and functionality. The experience gained in packaging, building, and deploying this is also useful for teams outside of development.

A migration is a project. The same principles that apply to any IT project apply to a migration, so consider iterative cycles (including testing) and risk mitigation. Watch out for scope and, more commonly, scope creep. For example, if you migrate the application server, migrate the DBMS, migrate your messaging platform, touch up the code, and upgrade the operating system all in one go, then plan on a very cautious integration effort. We recommend proceeding with no more than one product change per development iteration.

Finally, consider involving a migration expert to conduct an application review with a seasoned eye for potential code issues, and have him/her advise the migration team, or even execute the work.

J2EE COMPLIANCE

The first step during the application migration is to determine what J2EE versions you are using today and what the target version will be in your WebSphere target platform (see Table 3). Then compare and contrast the different API versions (e.g., EJB, servlet, JSP, etc.) to plan out the areas to focus on during the migration.

In addition, you should analyze the application to determine which code violates or loosely interprets the J2EE standards. This code may not seamlessly migrate over to the new environment. We detail a common example of J2EE noncompliance in the accompanying sidebar.

PROPRIETARY AND HOME-GROWN COMPONENTS

As part of the analysis process, it is also quite valuable to identify proprietary components specific to the application server, or home-grown middleware modules. These will also require dedicated attention during the migration strategy. Table 4 details an example of proprietary components included in WebLogic.

Next Steps

Whether you go the do-it-yourself route or use expert outside assistance, we recommend that you follow these next steps for your migration.

1. ***Assessment:*** It is always important to assess and plan for the overall migration effort, as well as understand the skills development necessary for adapting to the new environment. During this phase you will analyze existing applications and code, and create a comprehensive plan for efficiently making the move. The assessment can also be a valuable tool to confirm overall costs in an effort to study the total cost of ownership and return on investment involved in a move.
2. ***Proof of concept:*** This is more applicable during an application server migration, and goes a long way toward confirming costs and mitigating any concerns you have regarding the risk.
3. ***Migration:***
   –*Consider outsourcing.* A migration effort is really

V4 NEW FEATURES	V5 NEW FEATURES
• J2EE 1.2 compliant	**• J2EE 1.3 compliant**
• WebSphere editions (Single Server Edition, Advanced Edition, Enterprise Edition)	• Administration model using JMX
• Java 2 Connectors (JCA)	• WLM enhancements **
• Web services	• WebServices enhancements
• Web server plug-in	• Security enhancements
• Embedded HTTP server	• Messaging support (embedded JMS server, support for MDBs)
• Administration tools	**• Administration tools - JACL support for WAS scripting (wsadmin replaces wscp)**
• Performance enhancements *	
• Other tools	

TABLE 1: NEW AND IMPROVED FEATURES FOR WEBSPHERE V4 AND V5. THE HIGHLIGHTED FEATURES ARE AREAS OF PARTICULAR IMPORTANCE.

V3.5 TO V4	V3.5 TO V5	V4 TO V5
√ J2EE standard	√ J2EE standard	√ J2EE standard
√ Administration tools	√ Administration tools	√ Administration tools
√ Web server plug-in	√ Web server plug-in	
√ Embedded HTTP server	√ Embedded HTTP server	

TABLE. 2: THE AREAS TO CONSIDER WHEN MIGRATING TO A NEW VERSION OF WEBSPHERE

TECHNOLOGY	WEBSPHERE V4	WEBSPHERE V5
Java 2 Platform, Enterprise Edition (J2EE)	1.2	1.3
Java 2 Platform, Standard Edition (J2SE)	1.3.1	1.3.1
WebSphere v5.1 uses JDK 1.4.1		
Enterprise JavaBeans (EJB)	1.1	2.0
JavaServer Pages (JSP)	1.1	1.2
Java Servlets	2.2	2.3
Java Authentication and Authorization Service (JAAS)	–	1.0
Java Activation Framework (JAF)	1.0	1.0.1
JavaMail	1.1	1.2
Java API for XML Parsing (JAXP)	–	1.1
J2EE Connector Architecture (JCA)	–	1.0
JDBC Standard Extension (JDBC)	2.0	2.0
Java Message Service (JMS)	1.0	1.0.2
Java Naming and Directory Interface (JNDI)	1.2	part J2SE 1.3
Java Transaction API (JTA)	1.0	1.0.1
RMI-IIOP	1.0	part J2SE 1.3

TABLE 3: SUPPORTED API STANDARDS

TYPE NARROWING

Type narrowing with PortableRemoteObject is often not implemented to specification in some application servers. The EJB 1.1 specification states that all client-side representations of an EJB's Home and Remote interfaces must be narrowed. This includes beans referenced by JNDI, those created by the home interface, references returned by finder methods, and references returned from one bean to another.

Examples of Proprietary WebLogic Components:

- Use of proprietary WebLogic RMI
- Use of weblogic.workspace.common.WorkspaceDef
- Use of Custom authentication weblogic.security.realm.Class
- WebLogic Load at Start-Up: weblogic.common.T3StartupDef
- WebLogic Schedulable Tasks: weblogic.time.common.Schedulable
- WebLogic Triggerable Tasks: weblogic.time.common.Triggerable
- WebLogic InitialContext
- WebLogic DB ConnectionPooling: odbcPool
- WebLogic JDBC DriverManager
- Use of Java Threads in the JVM
- Use of Custom Classloaders
- Use of JSP Tag Libraries
- WebLogic Internationalization support

TABLE 4: **PROPRIETARY COMPONENTS: WEBLOGIC EXAMPLE**

a one-time-only effort. Rather than utilizing your staff on this effort and building up skills they will never require again, deploy them to strategic projects where it is essential to use developers who know your business. By leveraging expert consultants with previous expertise, you gain from their tips, tricks, and tools that expedite the process, reducing overall costs.

–*Do it yourself.* There are several do-it-yourself resources and books that can make this a smoother process. Consider finding a mentor with experience in this area so that you benefit from his or her accumulated knowledge.

4. ***Skills development:*** When moving from one environment to another you will need to arm your development team with the skills to maintain the system. Dedicate training time for the new environment and the new tooling.

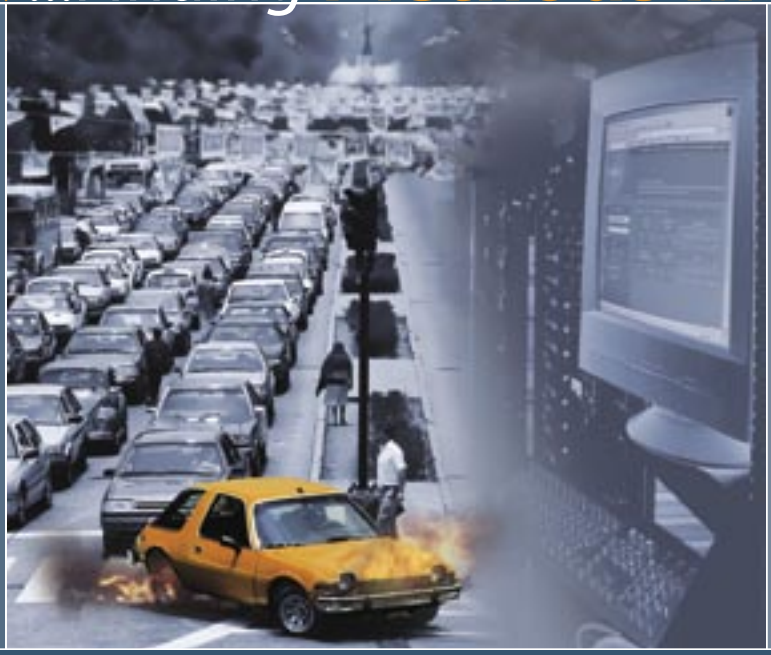
Juergen Efeish, based in Prolifics' New York office, is a senior member of Prolifics' specialized WebSphere Consulting Division – a team that IBM calls on to solve the toughest of their customers' challenges and deliver training, mentoring, and development services. He has extensive experience delivering WebSphere solutions for clients worldwide, including the Federal Reserve Bank, Verizon, National Healthcare Group – Singapore, AT&T, Ernst &Young, JP Morgan Chase, BMW, and Avaya. jefeish@prolifics.com

Max King, based in Prolifics' London office, has extensive experience delivering WebSphere solutions for clients worldwide, including JP Morgan Chase, NYSE, Wal-Mart, Lufthansa, BNP Paribas, MetLife, Honda, and Hertz. As a senior member of the Prolifics WebSphere Consulting Division, he specializes in J2EE architecture and best practices, application development and deployment, production "crit-sit" support, production readiness assessments, application migration, and messaging and integration. mking@prolifics.com

What's Obstructing Your Application Flow?

With Cyanea/One™, your company will gain valuable insight into its J2EE applications. With a few mouse clicks, you can quickly drill down into your applications, all the way down to the method level. Isolate, diagnose and proactively resolve bottlenecks and resource consumption issues. Accomplish all these without touching your code or even requiring knowledge of your application.

...*Finding* **Methods in the Madness.**



Find methods at:
www.cyanea.com/wsdj/nomoremadness.html
1-877-CYANEA8 | sales@cyanea.com

Cyanea™ and Cyanea/One™ are trademarks of Cyanea Systems Corp.
© 2003 Cyanea Systems Corp. All Rights Reserved.

Failure to manage memory can significantly hinder performance

Hunting Java Memory Leaks

BY LLOYD HAGEMO

& RAVI KALIDINDI

In a production environment, memory leaks can force organizations to add more memory and hardware resources. They can even cause an application to crash unexpectedly. In theory, Java memory leaks should not emerge as a development or production issue because the garbage collector is responsible for memory management.

Built-in Java memory management enables developers to sidestep tedious memory management considerations and is a fundamental reason why Java is a relatively straightforward development environment. Practically, however, the garbage collector function is unable to accurately address all memory management considerations, due to programming oversights and Java language program loopholes that allow memory leaks.

The garbage collector's job is to periodically collect unreferenced objects. Java memory leaks typically occur because the garbage collector reserves, or allocates, memory for each object associated with a program – whether or not the program uses the object. The garbage collector, as a result, reserves memory for each object, but does not have the intelligence to reallocate memory reserved for an idle object to another command. Consequently, the garbage collector falsely assumes that memory is being used by objects. The objects that are not being used but still hold references are not collected, thus leading to memory leaks.

Failure to adequately manage memory can significantly hinder the performance of your Java and J2EE applications. This column will outline the root causes of memory leaks and detail best practices for identifying and eliminating memory leaks in development and production to ensure robust Java and J2EE application performance.

Four Leading Causes of Memory Leaks

Memory management is a complex process, and the source of a memory leak is not always evident. Outlined below are four leading causes of memory leaks.

STATIC COLLECTION CLASSES

Static collection classes, such as HashMap and Vector, are susceptible to memory leaks because they are designed to hold other referenced objects. Static key words (as shown in Listing 1) are likely to cause memory leaks because static variables remain in memory as long as the application runs, regardless of its object creation and destruction.

LISTENERS

There are several listener types available in J2EE that provide notification when an event occurs. Until a J2EE application receives event notification, any storage held by that application will not be garbage-collected by the Java Virtual Machine (JVM). Web application listener events are new in the Servlet 2.3 Specification. These listeners include servlet creation and invalidation as well as application startup and shutdown. The two listener classes, javax.Servlet.servletContextListener and javax.servlet.http.HttpSessionListener, when not implemented correctly, can lead to long-running applications that hold storage. To avoid this problem, keep the storage associated with servlet sessions small to minimize the effect on the heap. The Graphics Interchange Format (GIF), for example, should not be held in a session object that is kept in session using a listener.

PHYSICAL CONNECTIONS

Physical connections, such as database and network connections, are not going to be garbage-collected automatically unless they are disconnected explicitly. Java database connections are typically initialized using the DataSource.getConnection() method, which is used to handle physical connections. These physical connections must be freed when they are no longer required by the application using the close() method. Because the scope of such connections is independent of the JVM, the garbage collector does not

have the ability to affect a physical connection. When your application server uses the connection pool, the DataSource.getConnection() method allows you to release connections back to the pool so other transactions can use the physical connection to access the database.

Connection pools bring advantages to application development by reducing the overhead of an application by physically connecting to and disconnecting from a database. The physical connections are maintained by the application server and managed from a pool that is farmed out to each application that retrieves data from a database. Using a connection pool limits the number of physical database connections. Therefore, it is very important for an application to release its connection once it has finished sharing information with a database.

JAVA NATIVE INTERFACE REFERENCES

Similar to the physical connections mentioned above, the Java Native Interface (JNI) allows Java code to run with other languages, such as C and C++. The references created in the C or C++ application environments should be unreferenced explicitly. Because the C or C++ storage is not allocated from the JVM Heap, it will not be garbage-collected by the JVM. Because the JVM does not have the ability to deallocate JNI references, JNI memory must be managed directly by the C or C++ application.

A Best-Practice Approach to Hunting Memory Leaks

It is not always easy to detect memory leaks by scanning your code. Not all of the programmers involved with a J2EE project may have the time or experience to code memory management appropriately. Although many project managers face this issue, there is no standard solution to this problem because nearly every Java environment is unique. There are, however, guidelines that development teams should follow to optimize memory performance. An overview of best practices for ensuring a high level of memory performance throughout each phase of the "build, run, manage" application infrastructure life cycle follows.

BETTER CODING

Listing 2 illustrates how programmers can overlook memory leaks. The helper object in Listing 2 is a memory leak until another class removes it from the LongReferent class. Listing 2 shows that the garbage collector is not always responsible for collecting memory. The memory leak in this program can be avoided by removing the helper object from LongReferent explicitly, by the same class or by some other class. Objects stored in a HashMap are removed using the clear() or remove(object key) HashMap class method calls. Once the reference is removed from the HashMap, it is a candidate for garbage collection. Always double-check that the references are released, especially when you write static collection classes, listeners, database access code, and JNI code.

One approach to avoiding memory leaks is to write both "create method" and "remove method" commands in each class, then check for remove-method usage. In the unit-testing phase, you can search for create-method and remove-method usage in the code. The memory code is functioning properly if there is an equal number of calls for the create method and remove method commands. If the number of calls for these methods don't match, you should fix the code by adding or removing methods wherever they are missing.

BETTER APPLICATION PROGRAM INTERFACE

To achieve an added level of control over memory and garbage collection from the program, J2EE introduced the java.lang.ref package. For example, the WeakHashMap class java.lang.ref package uses weak keys to reference objects within the garbage collection. An object is available for the garbage collector when the WeakHashMap key is invalid. The garbage collector can make a key invalid if the memory is low, allowing the object it references to be freed. The garbage collector moves the key through three stages: finalizable, finalized, and reclaimed. It is not unlikely that the application could try to retrieve an object only to determine the object has been reclaimed by a garbage-collection cycle. This process works effectively for objects that can be re-created and easily collected whenever memory is running low.

Listing 3 shows the usage of the WeakHashMap class. Listing 3 is similar to Listing 2, except that it uses the WeakHashMap instead of the HashMap class. The difference between the two classes is that HashMap contains strongly referenced keys, which cannot be removed until the program unreferences them explicitly. The WeakHashMap class contains weakly referenced keys that can be removed

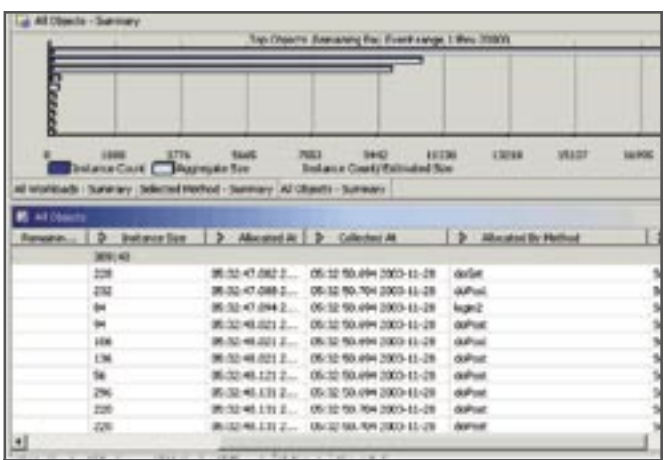


FIG 1: MEMORY ALLOCATION AND COLLECTION MAP

by the garbage collector without unreferencing, or removing, keys in the program when memory is running low. The garbage collector algorithm determines the removal of weakly referenced keys. This varies by JVM.

The garbage collector generally removes unused, weakly referenced objects first so that it can allocate the unused memory for new objects. The program also checks for null keys and refreshes them with keys and values when it uses WeakHashMap. As part of this process, the garbage collector works in the background to remove the keys automatically. These classes are important to consider, especially when you implement memory-hog functionality – such as caching modules – that enables cached objects to be easily re-created. This approach can be used to replace the HashMap calls in the first example.

BETTER DEVELOPMENT AND TESTING TOOLS

Even when following best practices, it is virtually impossible to avoid memory leaks in medium or large projects in which many programmers are writing thousands of lines of code. Fortunately, there are effective performance tools available to pinpoint memory leaks automatically. Some products are built into development solutions, and others are stand-alone tools. Some of the newest development tools include profilers, which enable developers to monitor performance issues from the same development integrated development environment. Profiler solutions from Candle Corp. and other vendors can pinpoint memory leaks automatically as part of a broader development tool. An effective profiler tool must enable developers to:

1. Identify which method allocated the memory
2. Clarify if the memory was released by the garbage collector and how long it was allocated
3. Determine which objects are holding the reference to the memory; Listing 1 shows that the HashMap is holding the memory reference

The bar chart in Figure 1 shows the top 10 allocated object references. The table below tells the programmer what program allocated the memory and if it had been garbage-collected. An object reference tab that shows the method holding the reference to the memory is also available. Many of today's tools can make it very easy to pinpoint memory leaks and, therefore, avoid production outages.

Subscribe Now!
$69.99
1 YEAR/
12 ISSUES
www.SYS-CON.com/JDJ
or call
1-888-303-5282

The World's Leading i-Technology Magazine

Just Got Better!

Now with expanded business coverage, *JDJ* is the world's premier independent, vendor-neutral print resource for the ever-expanding international community of IT professionals.

The Most Popular i-Technology Magazine in the World !

SYS-CON MEDIA
The World's Leading i-Technology Publisher

© 2004 SYS-CON MEDIA. ALL RIGHT RESERVED. OFFER SUBJECT TO CHANGE WITHOUT NOTICE

THERE ARE SHORTCUTS TO SUCCESS.

AND OUR CERTIFIED WEBSPHERE INSTRUCTORS WILL SHOW YOU THEM.

When you take a WebSphere class with WinMill Software, you're learning the tools of the trade from the most talented and respected experts in the country. Our vendor-certified instructors are experienced professionals who are well-versed in the latest techniques and spend 75% of their time in the field working on real-world projects. Through an intensive hands-on lesson plan and expert knowledge transfer, you'll learn to harness the true power of WebSphere.

Call 1-888-711-MILL (6455) or email us at inquiry@winmill.com for our latest price specials and offers.
Go to www.winmill.com to view our Course Schedule.

FREE RESOURCE: Enjoy access to WinMill's On-Line Education Forum (OLEF), designed to simplify and speed the application of your newly acquired skills. It's like taking your instructor home with you. Ask us about it!

420 Lexington Avenue, Suite 444, New York, NY 10170

We are currently offering the following courses:

Developing Web Applications with WebSphere Studio 5.0

IBM WebSphere Portal Administration Version 4

IBM WebSphere Portal Application Development Version 4

Introduction to Java using IBM WebSphere Studio Application Developer

Servlet and JSP Development with WebSphere Studio Version 5

Using EJBs to Develop Enterprise Applications with WebSphere Studio 5.0

WebSphere Application Server V5 Administration

Courses are held at our state of the art facility in New York City or on site per request.

Vendor-certified training also available for Macromedia, Check Point and Mercury Interactive products.

WINMILL SOFTWARE®
We make things happen!

BETTER PRODUCTION TOOLS

Although profilers excel at identifying memory leaks and other performance problems, developers and testers must be able to use the tools effectively. Memory problems that are overlooked in development will likely cascade in the production phase. Profiles provide development teams with a significant amount of performance data, which, in turn, may create more development cycles in the project due to the additional work associated with finely tuning memory management. Added production cycles may not be feasible for projects that have tight deadlines.

Fortunately, monitoring tools in production enable users to track J2EE application performance in real-world environments. Production profiling solutions from Candle, IBM, and other vendors monitor resource utilization for memory consumption. These tools provide visibility into the J2EE application environment that enables users to proactively improve performance and avoid production downtime.

Conclusion

Memory leaks often destabilize an application in production by consuming large quantities of memory or causing the JVM to crash. Moreover, whenever the system is low on memory the garbage collector tries to collect the memory more frequently, thus diminishing application performance. During development, programmers must pay special attention to memory leak problems associated with static collection classes, listeners, physical connections, and JNIs. The latest profiling tools identify memory problems long before the testing and production phases.

Production monitoring tools enable you to track application health and view otherwise-hidden performance metrics. Organizations are better able to tune memory management, which can improve the overall production stability and uptime. Organizations that address memory leaks by using a combination of development best practices and profiling and management tools can optimize memory management, which can increase the performance of their J2EE applications.

Lloyd Hagemo is a senior director for the Application Infrastructure Management (AIM) group at Candle Corp. He is responsible for IBM WebSphere tools development. He has led the successful development of more than 20 products for the WebSphere environment, including operating system utilities, network performance and tuning products, WebSphere MQ configuration and management tools, and application integration solutions.
lloyd hagemo@candle.com

Ravi Kalidindi is a senior software engineer for Candle's AIM group. He has worked closely with Java and J2EE since its inception on a wide variety of J2EE projects. He has published several articles that focus on Java and J2EE best practices.
ravi_kalidindi@candle.com

LISTING 1: CONTAINS STATIC HASHMAP TO HOLD HELPER MEMORY ALLOCATION

```
public class LongReferent {

   private static HashMap referent = new
HashMap();

   public void add(String index, Object
reference){
          referent.put(index, refer-
ence);
   }

   public Object get(String index){
          return referent.get(index);
   }
   public void remove(String index){
          if(get(index) != null){
                 referent.
remove(index);
          }
   }
}
```

LISTING 2: CAUSES A MEMORY LEAK BY ALLOCATING AND STORING MEMORY IN THE STATIC HASHMAP CLASS???

```
public class Caller {

   public static void main(String[]
args) {
          // 1. create an object
          Helper helper = new Helper();

/* 2. add object reference to a collec-
tion which stays in memory  for long
time */
          LongReferent referent = new
LongReferent();
          referent.add("helper", help-
er);

/* 3. set the object reference to null
so that it is eligible for Garbage
Collection */
          helper = null;

/* 4. helper is not going to be garbage
collected since the reference is added
in the LongReferent class */
   }
}
```

LISTING 3: WEAKHASHMAP USED TO FREE MEMORY IN COLLECTIONS

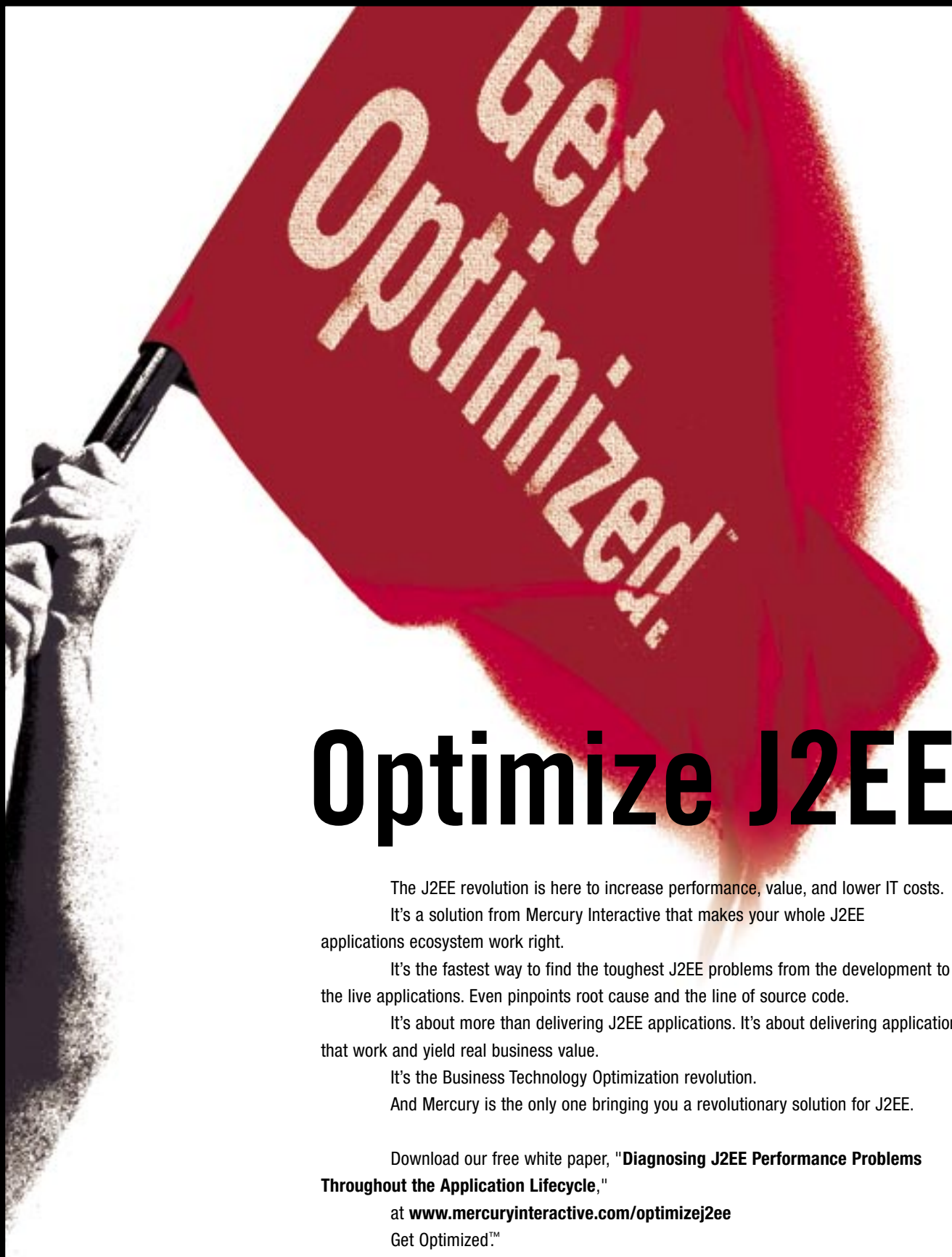
```
public class LongReferent {

   private static WeakHashMap referent =
new WeakHashMap();

   public void add(String index, Object
reference){
          referent.put(index, refer-
ence);
   }

   public Object get(String index){
          return referent.get(index);
   }

   public void remove(String index){
          if(get(index) != null){
                 referent.
remove(index);
          }
   }
}
```



Optimize J2EE.

The J2EE revolution is here to increase performance, value, and lower IT costs.

It's a solution from Mercury Interactive that makes your whole J2EE applications ecosystem work right.

It's the fastest way to find the toughest J2EE problems from the development to the live applications. Even pinpoints root cause and the line of source code.

It's about more than delivering J2EE applications. It's about delivering applications that work and yield real business value.

It's the Business Technology Optimization revolution.

And Mercury is the only one bringing you a revolutionary solution for J2EE.

Download our free white paper, "**Diagnosing J2EE Performance Problems Throughout the Application Lifecycle**,"

at **www.mercuryinteractive.com/optimizej2ee**

Get Optimized.™

MERCURY INTERACTIVE

© 2003 Mercury Interactive Corporation. Mercury Interactive, Get Optimized, and the Mercury Interactive logo are trademarks or registered trademarks of Mercury Interactive Corporation in the United States and/or select foreign countries.

IBM Middleware Helps Banks Deal with Challenges

Five new middleware solutions address the banking industry's needs

BY KARLA **FEUER**

Banks are facing increasing pressure to reinvent and expand their businesses. They need to respond to increasing regulatory and security requirements, expectations for new levels of customer service, the broader globalization of markets, and price pressure on their services.

IBM has also had to expand its banking products and services to meet business challenges faced by the banking industry. The result is five new middleware solutions that address the specific priorities and problems faced by banks today.

The solutions are part of IBM Software Group's strategy to deliver industry middleware solutions – a strategy based on customer buying behavior that indicates they prefer to buy solutions designed for their specific industry. Each industry middleware solution contains functions from IBM's WebSphere, Lotus, Tivoli, DB2, and Rational middleware brands combined with industry-specific middleware, applications from independent software vendors (ISVs), and industry-expert services.

For banking, IBM and its partners relied on years of experience working directly with large banking customers. "Our banking customers told us what challenges they need to address," says Wes Galbo, Banking Market Segment manager for IBM. "They also told us they need each of the four components of our solutions – the middleware, the industry-specific middleware, the industry applications, and the services."

Wes Galbo
Banking Market Segment manager for IBM

Among the biggest challenges banks face is the need to transform their branches to provide better customer service, cut costs, and generate more revenue. This is difficult now because their operating systems are approaching end-of-life and their technology infrastructure is inflexible, unmanageable, and expensive to support. One result of this is that bank employees can't get a single view of a customer's activities, which would be useful in both sales and service efforts.

IBM's Middleware Solution for Bank Branch Transformation is intended to help retail banking customers reduce costs, enhance sales, and improve customer service by improving employee collaboration and synchronizing customer information with the bank's other channels – the call center, ATMs, and the Internet. Branch transformation solutions involve migrating to newer, more appropriate technology and consolidating servers.

Doing so retools the physical space for maximum effectiveness and maximizes the payback from technology investments, says Galbo. The synchronized data helps banks "identify opportunities better in face-to-face and self-service interactions, and creates a unique experience for each customer. This also enhances brand value."

The branch transformation solution includes services that provide a branch strategy and roadmap, design and implementation, business process optimization, infrastructure resource management, asset management, help desk, network consulting, business resiliency, and continuity services (see Figure 1). The solution includes core middleware from WebSphere, DB2, Lotus, Tivoli, and Rational, in addition to new industry-specific middleware that supports the management of large distributed two- and three-tier branch infrastructures (see Figure 2). Some of the partners leveraging the branch transformation solution are S1 Corporation, Siebel, Financial Fusion, Eontec, and Callidus Software.

"Many banks are looking at the branch as the logical place to start building a comprehensive, integrated multichannel infrastructure. Using a common infrastructure for all customer channels provides for a quick return on investment." Galbo says (see Figure 3). They also need to empower all their channels to improve service and increase revenue. This need is one of the four other banking industry challenges IBM's solutions address.

Again, banks' attempts to empower their channels are hampered by the fact that different sales channels refer to different bank data. As a result, banks are unable to assign sales and service resources to customers based on profitability. In addition, they cannot drive effective marketing campaigns across multiple channels.

IBM's Middleware Solution for Banking Customer Insight helps line-of-business, sales, and marketing executives automate the creation and management of

multichannel, CRM, and targeted marketing strategies to motivate the right customers to embrace the most effective services and channel mix. The solution also delivers targeted, actionable information to channel employees at the point of customer contact. Overall, the solution helps banks develop a customer-centric business strategy, design systems based on the strategy, and implement the necessary processes and technology programs.

"These steps help maximize a bank's customer retention and satisfaction, and reduce the overall cost of managing customer relationships. The result is enhanced customer profitability and loyalty," Galbo says. "The improved relationship increases sales, maximizes return on investment, and ultimately can provide a competitive advantage."

Another challenge facing banks is the need to simplify the processes by which they make wholesale payments. Banks are the only entities legally permitted to make these payments, and customers expect them to do so in a timely, accurate, and trustworthy manner. In addition, customers are increasingly viewing payment services as a commodity purchase. The result is that while approximately 35 percent of a large, worldwide bank's revenue may come from payment services, they represent approximately 40 percent of their costs.

The reasons for the high costs are that large banks typically have, on average, 30 different payment systems. This duplication is in customer-facing applications, processes, functions, and data warehouses. In addition, there are multiple settlements, statements, inquiries, histories, and adjustments – and it is difficult to automate and monitor risk management activities related to wholesale payments.

IBM's Middleware Solution for Wholesale Payments is designed to help banks reduce costs associated with maintaining multiple, independent payments systems. It is intended to allow banks to react quickly and effectively to regulatory and market changes, new customer demands, and the intricacies of liquidity management, Galbo says.

The solution enables banks to cut costs by consolidating common processes, reusing business logic software code, and monitoring mission-critical payments. It also helps them differentiate their services to different types of customers.

This end-to-end solution can be carried out incrementally while leveraging legacy systems, bringing tangible returns on investments at each step, Galbo says.

Banks also need to simplify their risk and compliance systems in order to comply with increasing and evolving requirements in regulations such as Basel II, Sarbanes-Oxley, and the USA PATRIOT Act.

As with other processes, however, the data for risk and compliance activities is spread throughout banks. Furthermore, most current data management processes associated with risk and compliance are not capable of responding to regulations and requirements that seem to change almost daily. As a result, banks have little ability to monitor and manage internal compliance processes.

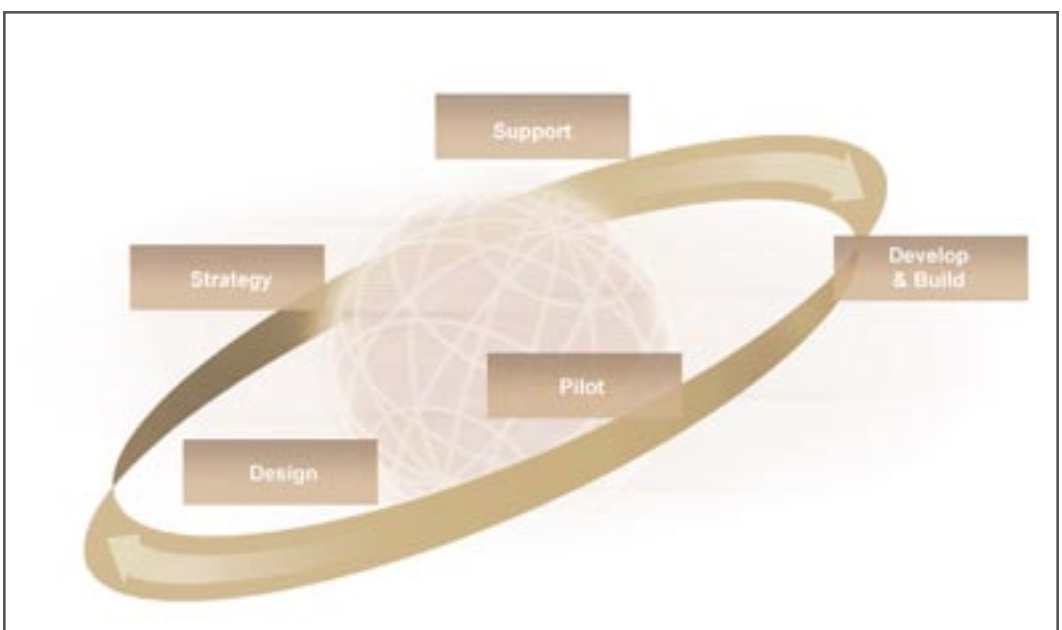


FIG 1: **BRANCH TRANSFORMATION SERVICES: IBM CAN ASSIST BANKS IN ALL PHASES**

IBM's Middleware Solution for Banking Risk and Compliance is designed to help banks effectively manage risk and compliance activities throughout the enterprise. It does so by providing common data and services to assist in compliance. This can also help banks exceed compliance requirements to achieve new business efficiencies and thus competitive advantage, Galbo says.

The solution works by collecting credit, market, and operational risk data from across a bank's organization and providing the foundation for supervisory review. This ensures data quality and consistency and provides an enterprise view of risk and compliance activities with up-to-the-minute and ad hoc access to information and reports. The solution also automates interaction with vested parties to resolve issues.

"Banks are thus able to more easily identify abnormalities and take corrective action to minimize risks," Galbo says. "In addition, by reducing manual processes and managing human interventions, they minimize operational risk."

Another critical challenge facing banks today is the need to transform their core systems, which form the critical basis of their operations. Core systems are the primary repository of customer, account, and product data, and are responsible for key functions such as processing and posting transactions, performing deposit accounting, maintaining loan accounts, keeping securities positions, and clearing payments.

These systems also provide most bank employees with incomplete views of each customer. In addition, they are difficult to upgrade to support new customer-centric processes, making it challenging to build new products and services in response to new opportunities. As a result, it is difficult for banks to align their products with customer needs, which can set them apart from competitors.

IBM Middleware Solution for Banking Core Systems Transformation helps banks update, replace, and reengineer the core business processes and computer systems that support both staff and customers from the back office of retail banks. It also helps banks link core applications throughout their operations within the context of business processes.

All of IBM's five industry middleware solutions for banking are built on open standards, allowing freedom of choice and interoperability while preserving investment, Galbo says.

Customers don't have to buy the full package all at once. "We've made our solutions comprehensive, but because they are modular, customers can buy pieces or the entire solution," he says. In addition, IBM's stable of banking-specific partners allows banks to assemble best-of-breed solutions, Galbo says, adding that IBM is recruiting an ecosystem of ISVs on an ongoing basis. "It's not in the traditional product sense, but by industry solution," Galbo says.

Other IBM advantages, Galbo says, are its leadership in defining and developing open standards (IBM's infrastructure is open, unlike Microsoft's) and IBM has unbiased relationships with business application software vendors, unlike SAP and Oracle (IBM is not in the applications business).

IBM's banking solutions also help customers become fully integrated, on-demand enterprises, Galbo says. "IBM's on-demand strategy is not about technology for the sake of technology; it's about enabling new ways of doing business and helping an organization reach new levels of innovation while continuing to deliver the increases in productivity that are necessary to improve the bottom line," Galbo adds. "That's important in an industry facing the challenges banking faces."

"Businesses adapting to cope with ever-increasing pressures from competition and other factors can benefit from being fully integrated across people, processes, and information, including suppliers and distributors, customers, and employees," Galbo says. "They can then respond with flexibility and speed to any customer demand, market opportunity, or external threat."

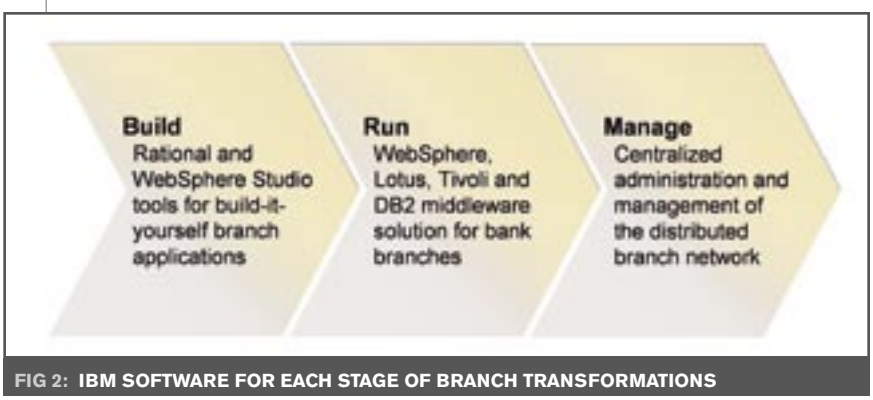


FIG 2: **IBM SOFTWARE FOR EACH STAGE OF BRANCH TRANSFORMATIONS**

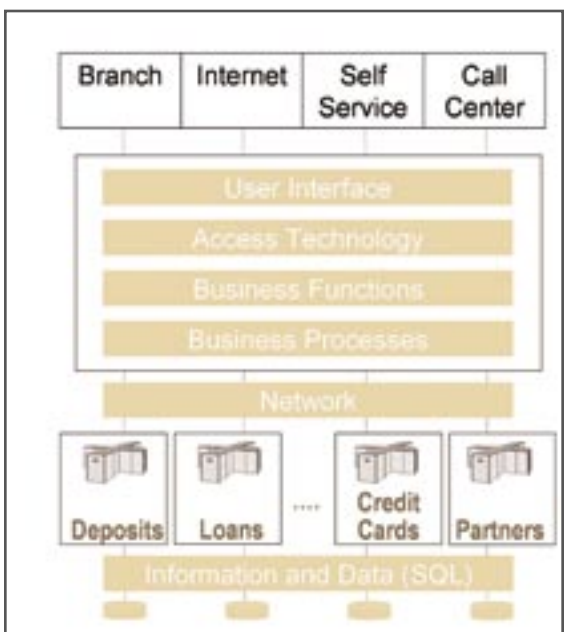


FIG 3: **ROI FOR MULTICHANNEL TRANSFORMATION**

Karla Feuer has been a corporate communications consultant since November 1994, specializing in strategies and implementation of employee and executive communications, public relations, newsletters, Web site content, and general writing and editing. She has handled numerous assignments for IBM, including eight years with the IBM Software Group, producing newsletters, executive communications, and Web writing. swsnews@us.ibm.com

Think .NET development is more productive than J2EE?

Think **again**.

Think **better**.

ThinkCAP™

ClearNova's ThinkCAP is a comprehensive application platform that simplifies and accelerates the development and maintenance of J2EE-based business applications by 50 to 70%.

ThinkCAP's visual & intuitive designers bring high productivity to business developers (those with VB or PowerBuilder-like skills), content owners, and administrators while allowing J2EE architects & programmers to leverage its component infrastructure and build business logic using the tools and approaches they prefer. ThinkCAP utilizes existing infrastructure, web services, legacy systems, and business applications.

ThinkCAP saves organizations time and money—and lowers project risks. Applications are written faster and require less maintenance. Project teams utilize in-house skills and require less training. Existing infrastructure and application servers are leveraged.

With ThinkCAP you can build quality applications faster.

Learn more about ThinkCAP at www.clearnova.com/thinkcap

CLEARNOVA

APPLICATION DEVELOPMENT • SIMPLIFIED • ACCELERATED

- Highly Visual Development Environment
- MVC Framework with Page Flow & Actions
- Advanced Data Aware Controls:
  Forms, DataViews, Queries, Navigations
  Workflows, Graphs, Treeviews, Grids, Tabs
- Smart Data Binding™ to data, objects, XML, sessions, or requests
- Browser & server-side validation
- Visual unit testing with RapidTest™
- Service Flow Designer aggregates Web Services, EJBs, XML, and POJOs
- Content Management engine & tools
- Supports .NET clients
- Integrated, seamless security
- Use any app server or 3rd party tool

Comparing WebSphere Commerce 5.5 and WebSphere Portal content publishing 5.0 Personalization

Getting Personal

BY BHADRI **MADAPUSI**

Whether you are a business user, an IT architect, or a developer using WebSphere Portal content publishing general personalization framework and WebSphere Commerce, you will need to know the difference between the personalized marketing campaigns created using these two sets of tools. Furthermore you can enable WebSphere Commerce server to make use of the WebSphere Portal content publishing personalization framework to provide marketing campaign capability. In order to enable WebSphere Commerce server to use WebSphere Portal content publishing personalization framework for organizing and delivering marketing campaigns, you must first know the difference between the two tools.

IBM WebSphere Portal content publishing provides a general personalization framework that you can use to organize and implement sets of personalization behavior such as Web page personalization and the delivery of personalized marketing campaigns. IBM WebSphere Commerce marketing tools make use of the Blaze personalization engine and its internal personalization engine as a means to organize and deliver personalized marketing campaigns.

This article compares the WebSphere Portal content publishing general personalization framework and WebSphere Commerce marketing tools on the following criteria: concept, capability, and usability. The article concludes by summarizing the trade-offs in using one tool over the other.

Both approaches consider a personalized marketing campaign to be a mapping between a set of contents and a set of customer segments (see Figure 1). The two differ in the way this mapping is organized (at creation time) and carried out (at runtime).

WebSphere Commerce Marketing Tools

WebSphere Commerce marketing tools use two fundamental components to deliver personalized marketing functions: initiative and campaign. An *initiative* defines the mapping between content and segments. A *campaign* is a logical container for groups of related initiatives that have a common business objective; a campaign exists within the scope of a commerce store. Figure 2 illustrates the relationship between campaign and initiative.

Each initiative maps the content to be displayed to the shopper, the segment (shopper segmentation group), and some additional condition (see Figure 3), which includes days on which the initiative must be active and the shopper's purchase behavior, among others. Initiatives are essentially used to achieve the business objective of showing particular content to certain users. Figure 3 illustrates the relationship between an initiative and its entities.

You can display the following content types to shoppers using the WebSphere Commerce solution: a set of product SKUs, a set of categories, or a set of collateral images to achieve business objectives such as suggestive selling (upselling or cross-selling products to a customer based on their purchase behavior) and awareness advertising (displaying advertisements to increase customers' awareness of a product or to inform customers about upcoming events or discounts). Figure 4 illustrates these content types.

An *e-Marketing Spot* is a JavaBean used on a JavaServer Page to display the content returned by an initiative. Marketing managers select the initiatives they want to show and map them to an e-Marketing Spot. This mapping is called a *schedule*. Each initiative has an active lifetime, which is specified during the mapping. Figure 5 illustrates the relationship between an initiative and an e-Marketing Spot.

Figure 6 is a conceptual model of

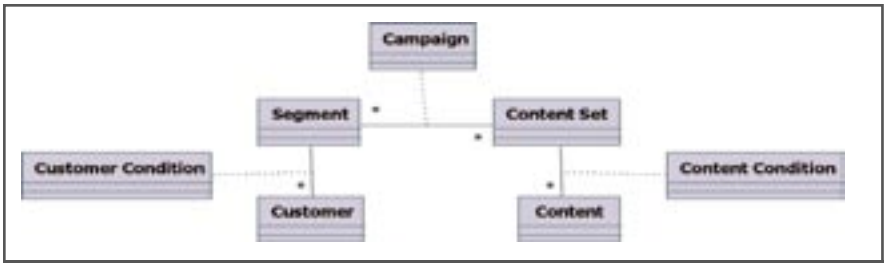


FIG 1: BASIC PERSONALIZED MARKETING CAMPAIGN CONCEPT IN WEBSPHERE COMMERCE AND WEBSPHERE PORTAL CONTENT PUBLISHING

a WebSphere Commerce personalized marketing campaign.

WebSphere Portal Content Publishing Personalization

WebSphere Portal content publishing personalization provides rules-based filtering, i.e., content definition and segment definition are maintained as rules. There are three fundamental types of rules: action, profiler, and binding.

The content definition rules, known as *action rules*, are used to select content for display. The segmentation rules, known as *profiler rules,* are used to categorize a Web site visitor or the characteristics or circumstances of his or her visit, such as the time of day in which it occurs. A profiler rule contains one or more profiles, each of which defines an individual segmentation rule. The profiler rule acts as a container for related profiles. A *binding rule* combines profilers and actions into sophisticated conditional clauses that cause actions to be performed based on evaluation by the profilers. Binding rules essentially are used to achieve the business objective of showing particular content to certain users. Figure 7 illustrates the relationship between actions, profilers, and bindings.

WebSphere Portal content publishing uses strongly typed JavaBeans called content spots to display the content returned by a binding. You can select the content to be shown by a content spot by mapping an action rule or binding rule to the content spot. Each such mapping has an active lifetime that must be specified during its creation. If a content spot is mapped to more than one binding or action during the same time period, then contents will be selected at random based on the weights assigned to individual mapping. Figure 8 illustrates the relationship between a content spot, a binding, and an action.

A campaign in WebSphere Portal content publishing is also a container that groups mappings (binding–content spot mappings or action–content spot mappings) that have a common business objective. Unlike a WebSphere Commerce campaign, a personalization framework campaign has the following attributes: active lifetime, priority, and splits.

Multiple campaigns can be active during the same time period. In that case a user must specify a priority for each campaign so the framework can determine which campaign to use. In case a user specifies multiple campaigns with the same priority, a random campaign is chosen based on the weights assigned to individual campaigns. Figure 9 illustrates the conceptual model of a WebSphere Portal content publishing personalization framework.

Comparison of Concepts Between the Two Models

Table 1 lists the conceptual differences between the two models. A number of capability and usability differences between the two models arise from these conceptual differences. I will discuss these differences in later sections.

Some of the concepts of the WebSphere Portal content publishing personalization framework can be abstracted and mapped to concepts in the WebSphere Commerce Marketing tool. Table 2 describes these mappings.

Differences in Capability

The conceptual difference between the models affects the capability of each model. The following sections discuss the capability differences during personalized marketing campaign creation (setup) and runtime (execution time).

CAMPAIGN CREATION

Being a general framework, WebSphere Portal content publishing has a free-form query interface to create actions (content rules) and profiles (segmentation rules). This interface allows the creation of a rich set of complex actions and profiles. Conversely, WebSphere Commerce uses a wizard to create content and segments. This interface restricts the creation of content and segment rules to a limited set.

The profile in WebSphere Portal content publishing can be used to monitor the runtime behavior of

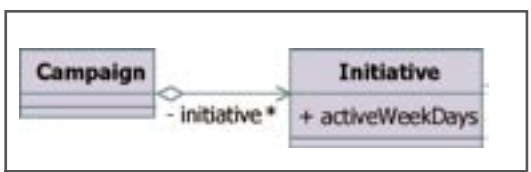


FIG 2: RELATIONSHIP BETWEEN CAMPAIGN AND INITIATIVE

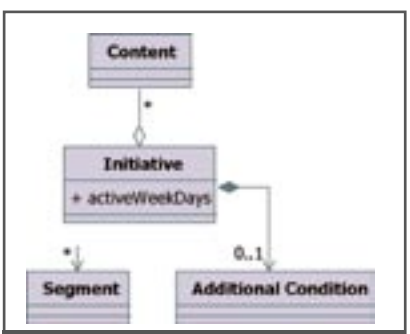


FIG 3: RELATIONSHIP BETWEEN AN INITIATIVE AND ITS ENTITIES

the user, including pages visited or actions performed, and show appropriate content based on these runtime behaviors. WebSphere Commerce allows runtime monitoring of user actions using additional conditions but this is restricted to items in the customer's shopping cart and purchase history.

In WebSphere Portal content publishing, actions can be directly mapped to content spots, so all of a store's customers can view the content related to an action when an associated content spot is executed; the content is no longer restricted by user profile. In WebSphere Commerce, content cannot be implicitly mapped to an e-Marketing Spot; only an initiative can be mapped to an e-Marketing Spot. In WebSphere Commerce, if you want to show content to all customers, you must explicitly state that during creation of the initiative.

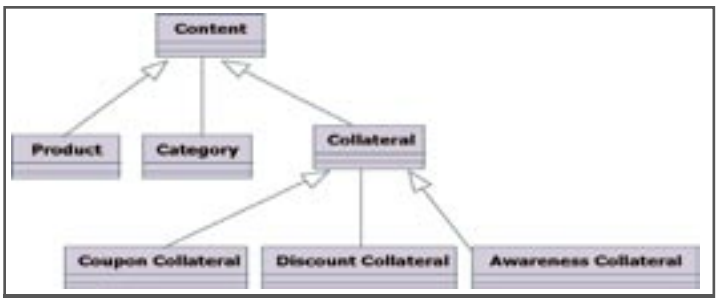


FIG 4: CONTENT TYPES IN A WEBSPHERE COMMERCE CAMPAIGN

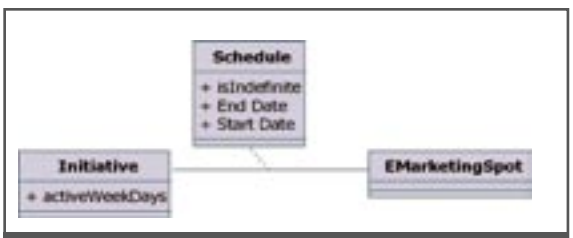


FIG 5: RELATIONSHIP BETWEEN INITIATIVES AND AN E-MARKETING SPOT

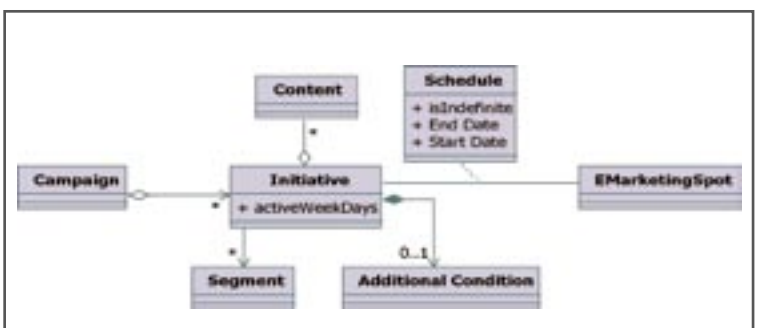


FIG 6: WEBSPHERE COMMERCE MARKETING CAMPAIGN CONCEPTUAL MODEL

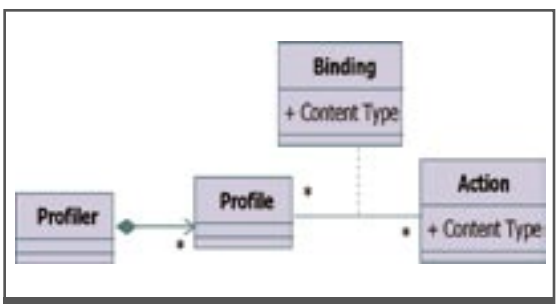


FIG 7: THE RELATIONSHIP BETWEEN ACTIONS, PROFILERS, AND BINDINGS

WebSphere Portal content publishing not only allows creation of rules to display content to customers (based on customer profile) but also allows the creation of rules to update underlying data in the data store. WebSphere Commerce allows only the creation of rules to display content to customers.

WebSphere Portal content publishing allows you to create default content to be shown (called Normal View); this default is used by the runtime environment unless superseded by an active campaign. WebSphere Commerce does not have a mechanism to show any default content.

WebSphere Portal content publishing also offers a preview mode. It is possible to test the operation of rules and campaigns, and to see how the rules will respond to customers with various attributes, or at different times or dates. This preview functionality is not available in WebSphere Commerce. Table 3 summarizes the differences discussed in this section.

CAPABILITY DIFFERENCES AT RUNTIME (CAMPAIGN EXECUTION TIME)

Content spots in WebSphere Portal content publishing are typed, meaning that at runtime a content spot can retrieve content of only one type. The e-Marketing Spots in WebSphere Commerce are typeless. At runtime a spot can retrieve content of multiple types.

In WebSphere Portal content publishing, only the campaign with the highest priority is evaluated. If this campaign does not have a mapping (binding or action/content spot mapping) for the content spot used in the store page, the runtime will move on to the campaign with next-highest priority until it finds a campaign with a mapping for the content spot that can return at least one content item. If none of the campaigns for the content spot can return content, the runtime defaults to Normal View. In WebSphere Commerce, all of the active campaigns are evaluated. The e-Marketing Spot combines the content retrieved from all of the active campaigns for display.

Campaign splits and mapping splits play a significant role in WebSphere Portal content publishing. When multiple active campaigns have the same priority, the splits are used to calculate a percentage chance that one mapping will be used instead

Versata's Rules

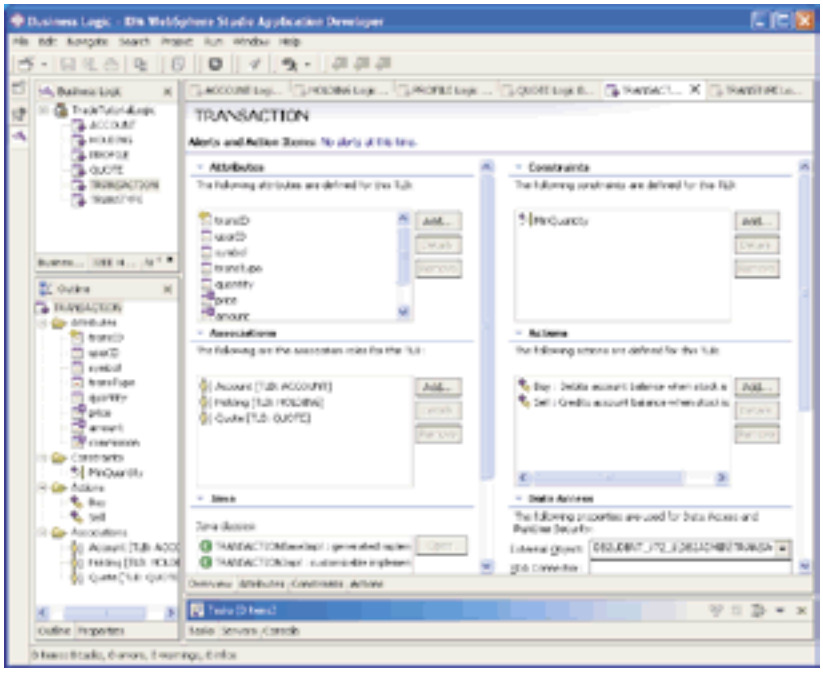
for

Simplifying WebSphere Studio Development:

RULE THREE

DEVELOPERS LIKE NEW SOFTWARE, ESPECIALLY **FREE** SOFTWARE.

J2EE can be a bear without Versata business logic software. As a plug-in for IBM's WebSphere Studio, Versata helps you develop and manage server-side business logic components, driving data for JSPs, HTML, Java, and Web services - making life a whole lot easier. Rather than program hundreds of lines of code, you simply specify your business logic as business rules: Versata does the rest.



versata™

To download a **free** copy of the Versata Business Logic Designer for WebSphere Studio, visit **www.versata.com/businesslogicdesigner**

of the others. The decision made by this split percentage is valid for the entire customer session. WebSphere Commerce does not use the split concept, as it executes all active campaign initiatives.

A marketing manager may want to suspend active campaigns due to business reasons. WebSphere Commerce offers the capability to suspend active campaigns at runtime. Because WebSphere Portal content publishing does not offer this capability, in order to suspend a campaign it must be republished with a new timeframe.

WebSphere Portal content publishing uses the BRBean rules engine to evaluate the content rules and segment rules. WebSphere Commerce uses the Blaze rules engine and its internal engine to evaluate the content rules and segment rules. Table 4 summarizes the runtime differences discussed in this section.

WEBSPHERE PORTAL CONTENT PUBLISHING PERSONALIZATION FRAMEWORK	WEBSPHERE COMMERCE MARKETING TOOL MODEL
A campaign is a container that groups a set of mappings (binding or action-content spot mappings). Campaigns also define other semantics such as active lifetime, priority, and splits.	A campaign is a container that groups a set of initiatives (content-segment mappings) and has no additional semantics.
Active lifetime of the mapping is controlled at the campaign level, and at the mapping level. The duration defined at the mapping level must not exceed the duration defined at the campaign level.	Active lifetime of an initiative (content-segment mapping) is controlled at the initiative-e-Marketing Spot mapping (schedule) level.
The content spot is typed, that is, one spot can return only one type of content.	The e-Marketing Spot is typeless, that is, the same spot can be used to retrieve content of different types.
Bindings (content-segment mapping) and actions can be mapped to content spots.	Only initiatives (content-segment mapping) can map to e-Marketing Spots.
Campaigns take effect only when they are published to the production server.	Campaigns take effect immediately upon creation.

TABLE 1: CONCEPTUAL DIFFERENCES BETWEEN THE WEBSPHERE COMMERCE CAMPAIGN MODEL AND THE WEBSPHERE PORTAL CONTENT PUBLISHING PERSONALIZATION FRAMEWORK

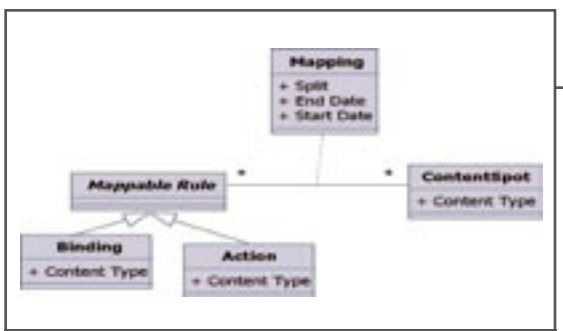


FIG 8: RELATIONSHIP BETWEEN A CONTENT SPOT, A BINDING, AND AN ACTION

FIG 9: WEBSPHERE PORTAL CONTENT PUBLISHING PERSONALIZATION FRAMEWORK CONCEPTUAL MODEL

DIFFERENCES IN USABILITY

The WebSphere Portal content publishing personalization framework is meant to be a generic model, and can be plugged into any Web application. The WebSphere Commerce marketing solution is meant to work within the scope of WebSphere Commerce. This basic difference gives rise to the following differences in usability between the two models.

The pluggable framework provided by WebSphere Portal content publishing expects certain resources to be provided by the application into which the framework is plugged. These resources are wrappers to the base product data store and are used in the creation of actions and profiles. WebSphere Commerce campaigns, on the other hand, are self contained; the resources used in the rules are provided out-of-box by WebSphere Commerce.

One of the major differences between WebSphere Portal content publishing and WebSphere Commerce is the user interface. WebSphere Portal content publishing uses a structured hypertext query builder interface to build actions (content rules), profiles (segment rules), and bindings. WebSphere Commerce instead uses a wizard-based business user interface to build initiatives. In WebSphere Commerce, content rules, segment rules, and content-segment mappings are constructed automatically, and the end user is sheltered from building the query logic directly.

As WebSphere Portal content publishing uses a free-form query builder, creating new rules requires only the resources that are required to construct rules. To create new rules in WebSphere Commerce requires changes to the user interface and probably changes to the Blaze rules engine as well (depending on the type of change). Table 5 summarizes the differences in usability discussed in this section.

WEBSPHERE PORTAL CONTENT PUBLISHING PERSONALIZATION FRAMEWORK	WEBSPHERE COMMERCE MARKETING TOOL MODEL
Mapping (Binding-content spot mapping)	Schedule (Initiative-e-Marketing Spot mapping)
Binding (Action-profile mapping)	Initiative (Content-segment mapping)
Action	Content
Profile of a profiler	Segment
Content spot	e-Marketing Spot
Campaign	A set of schedules
No mapping	Campaign

TABLE 2: MAPPING OF CONCEPTS

WEBSPHERE PORTAL CONTENT PUBLISHING PERSONALIZATION FRAMEWORK	WEBSPHERE COMMERCE CAMPAIGN TOOL MODEL
Action and profile rules can be rich and complex	Content and segment rules are simple and limited
Can monitor shopper runtime behavior using profiles	Can monitor limited shopper runtime behavior using additional condition
Action rules can be directly mapped to content spots	Content cannot be mapped directly to e-Marketing Spots. Only initiatives can be mapped to e-Marketing Spots
Update action rules can be created	No mechanism to create update rules
Normal View acts as a default campaign	No default campaign
Allows preview of rules and campaign	No preview functionality

TABLE 3: SUMMARY OF CREATION-TIME DIFFERENCES

Conclusion

There are trade-offs in using one model over the other. To conclude the article, I list these tradeoffs below.

The WebSphere Portal content publishing personalization framework has a powerful content and segment rule creation mechanism compared to that of the WebSphere Commerce Campaign model.

WebSphere Portal content publishing operates in an author-publish environment, and has a preview mode to preview all of the campaign rules before publishing to the production server. The WebSphere Commerce campaign solution does not operate in an author-publish environment. An initiative becomes active as soon as it is created, and does not have a preview mechanism.

The Normal View feature in WebSphere Portal content publishing acts as a default campaign, whereas there is no default initiative in WebSphere Commerce. However, default content can be built into the JavaServer Pages.

One of the major differences between WebSphere Portal content publishing and WebSphere Commerce is the runtime behavior. At runtime with WebSphere Portal content publishing only the campaigns with the highest priority that can return content are evaluated, and if these campaigns do not have mappings (binding or action-content spot mapping) for the content spot

WEBSPHERE PORTAL CONTENT PUBLISHING PERSONALIZATION FRAMEWORK	WEBSPHERE COMMERCE CAMPAIGN TOOL MODEL
Content spots are typed and can retrieve contents of only one type	e-Marketing Spots are typeless and can retrieve contents of multiple types
Only a campaign with the highest priority and that can return contents is executed	All the active initiatives are executed
Splits at the campaign and mapping level play a significant role	Splits are not needed
Cannot suspend active campaigns at runtime.	Can suspend active campaigns at runtime
Uses BRBean rule engine to evaluate rules	Uses Blaze rule engine to evaluate rules

TABLE 4: SUMMARY OF RUNTIME DIFFERENCES

WEBSPHERE PORTAL CONTENT PUBLISHING PERSONALIZATION FRAMEWORK	WEBSPHERE COMMERCE CAMPAIGN TOOL MODEL
A generic solution that can be plugged into any Web application	Part of WebSphere Commerce, and can be used only within the scope of WebSphere Commerce
Base product must provide resources to construct rules	All the resources are provided out of the box
Free-form query builder interface to construct rules	Wizard-based business user interface for constructing rules
Addition of rules requires creation of new resources	Addition of new rules requires changes to user interface and Blaze rule projects

TABLE 5: USER INTERFACE DIFFERENCES

specified in the store page, then the runtime will default to the Normal View. WebSphere Commerce always evaluates all of the active initiatives and returns the union of content that is targeted at the customer in a random order.

WebSphere Portal content publishing does not have a mechanism to suspend a campaign at runtime, whereas WebSphere Commerce does have a mechanism to suspend an initiative at runtime.

The WebSphere Portal content publishing personalization framework is a pluggable solution, so it is highly customizable and flexible compared to the WebSphere Commerce campaign model. However, with the WebSphere Portal content publishing campaign model you cannot create rules out of the box. WebSphere Portal content publishing expects the application it is plugged into to provide resources to create content and segment rules. For WebSphere Commerce, all the resources required to create basic content and segment rules are provided out of the box.

WebSphere Portal content publishing provides a free-form query user interface to construct content rules (action), segment rules (profile), and bindings. This user interface may be suitable only for technical users, and business users may find it difficult to use. WebSphere Commerce provides a wizard-based business interface to create initiatives. The low-level rules are created automatically. But the downside of this is that customization requires extensive modifications to the user interface.

For more in-depth information on how campaigns are organized and executed, refer to the WebSphere Portal content publishing online help documents and the WebSphere Commerce help documents online help.

Bhadri Madapusi is a software developer working in the IBM Toronto Lab's Electronic Commerce Division. He earned his MSc in information systems from BITS, India, and his MSc in computer science from Queens University, Canada. Bhadri's interests include software modeling, design patterns, and data management. bomadapusi75@yahoo.com

Notice: All the opinions expressed in this document are views of the author and not of IBM Canada Ltd., or any other individual or organization.

Trademarks: IBM and WebSphere are registered trademarks of International Business Machines Corporation in the United States, other countries, or both. Other company, product, and service marks may be trademarks or service marks of others.

PREVIEWING MAY 11, 2004, AT NETWORLD+INTEROP, LAS VEGAS

INFORMATION STORAGE+SECURITY JOURNAL!

FOR MORE INFORMATION VISIT

WWW.ISSJOURNAL.COM

SYS-CON MEDIA

From the World's Leading i-Technology Publisher

HATS Off to Legacy

Make a great impact in a short amount of time using available resources

BY LOUIS J. **IACONA**

Get ready for another acronym – HATS (Host Access Transformation Server). Let's begin with a notional conversation – an "elevator pitch" of sorts – between two IT managers, Jim and Dennis.

JIM: Hey Dennis, I hear that we can now create journal entries from our intranet. How is that possible? Did we contract that same outfit to build us another application?
DENNIS: Not really; we're using WebSphere HATS.
JIM: Never heard of it. What exactly is that?
DENNIS: Oh, it's very slick. It's an extendable framework that can transform legacy application screens into Web browser content – in real time. You can now click your way through journal forms in a browser instead of special-keying your way around the 3270 emulator.

Over 60% of the world's software systems are considered legacy: so-called "green-screen" applications – with line-oriented user interfaces, generally hosted on mainframe-size computers, and implemented in languages like COBOL and RPG. They're likely accessed through special terminals or terminal emulator types such as 3270, 5250, and VT.

Legacy applications typically represent significant investment and tend to support critical business operations. IT organizations are experiencing unprecedented budget cuts, and it would be reasonable to surmise that legacy applications won't be going away any time soon. This is much to the chagrin of many end users and business-unit managers.

While legacy applications are typically "well-oiled machines" that perform their functions efficiently and with predictable response times, their overall presentation and usability often leave a lot to be desired. A line terminal interface can be nonintuitive and difficult to navigate. Furthermore, each legacy application can be limited to a narrow band of functionality – as they were often designed with functional compartmentalization in mind.

This can present a situation in which no single legacy application supports the job function of any given staff member. At best, this leaves users bouncing from one legacy application to another. In the worst case, related information is simply not accessible by the users who would benefit the most, and crude workarounds are sought.

Fortunately, what most organizations actually have is a hybrid environment – one in which legacy systems coexist with a modern enterprise platform (i.e., desktops, browsers, J2EE, .NET, etc.). These organizations can now be offered an opportunity to leverage their legacy systems with their investment in a 21st century enterprise by creating composite applications.

Composite applications draw upon an architectural pattern from which applications are built by combining some new logic with existing applications. WebSphere HATS applications are an example of a specialized composite pattern, one that transforms and combines legacy applications by negotiating with the same interface used by terminal emulators.

HATS is difficult to describe simply and completely, but here's an attempt: HATS provides a set of components to rapidly build, deploy, and launch customized applications that dynamically transform line-oriented interfaces into content that can be presented by a Web browser. Casting aside the details for the moment, this has some very interesting implications. An organization's legacy applications might be made accessible through:

- An internal intranet and/or portal
- A public Web site, extranet, or VPN
- A PDA device such as a wireless Palm Pilot or a Pocket PC

Providing access to legacy content in this manner could not have been seriously considered a short time ago without a major redevelopment effort.

Backdrop

HATS is included in IBM's Host Integration Solution (HIS) offering. It is one of a set of WebSphere Application Server (WAS)–centric components that extend 3270,

5250, and VT interfaces to a standard Web browser (or to custom component). HATS applications perform this transformation according to a predefined rule set and programmed logic. In a grand three-tier context, HATS applications are the user interface, although they can (and often do) extend a business-rule tier or introduce logic of their own.

The term HATS is used in both a development and a deployment context. During development, "HATS" refers to a plug-in within the WebSphere Application Development Studio. Here, "HATS" typically refers to a project that is being developed within the studio. At some point, a completed project is deployed as a HATS application within a WAS instance.

The following HATS terms are used throughout this column.

- ***HATS:*** The complete set of development and runtime components that make up the HATS framework – essentially, the product at large.
- ***HATS Studio:*** The WebSphere Studio plug-in that allows a developer to view a HATS-specific perspective and create HATS projects.
- ***HATS project:*** The complete set of HATS resources and other code that gives definition and form to a HATS application.
- ***HATS application:*** An application (an EAR) generated from a HATS project that is then deployed and used – not unlike any other WebSphere-deployed application.
- ***Hosted application:*** An existing legacy application – as one relates to HATS, a 3270, 5250, or VT emulated application. A HATS application is associated with at least one hosted application. As of HATS Release 5 (the latest), a HATS application can be configured to "connect" to multiple hosted applications, a feature that previously could be leveraged only through the HIS Host Publisher product.

At first look, there is much to appreciate about the HATS framework. During development, the HATS Studio (just another WebSphere Studio perspective) provides a set of wizards to rapidly specify a rule set for native-screen-to-browser-content transformation. At runtime the benefits of any WAS deployed application apply to HATS applications, including:

- A zero client-side footprint – it's HTML/DHTML within a browser
- Standard underpinnings (servlets, JSPs, tag library extensions, etc.)
- Extendibility through J2EE-compliant application services
- Scalability, including support for load balancing and failover

IN AND OUT OF FOCUS:

This article is not a substitute for a technical tutorial; there are plenty of online resources to get you started and provide you with reference material (see the Resources section). The main goal here is to provide a basic awareness of the product: what it does and how it does it, the HATS value proposition, and pointers to more detailed information. Also reference the IBM Web site and/or a representative for more transient information such as pricing, the license model, and product/platform compatibility issues.

PREREQUISITES

A surface knowledge of the following areas would be useful, as they are not explained in detail here:

- Basic J2EE components
- The core WebSphere product suite
- The common incarnation of legacy systems

Beyond a thorough understanding the HATS product, the broader skill set required to execute or lead a HATS engagement through completion is dependent on the level of required customization and the sophistication of the deployment scenario. Extending the list above, some of the skill set areas that might be needed in a more complex HATS engagement include:

- GUI design and advanced graphics
- 3-tier Web application design
- OOD/OOP and advanced J2EE features
- Advanced WAS deployment
- Pervasive content delivery

Regardless of the complexity, a successful engagement is most contingent on building the right application – covering the required use cases. With that in mind, someone on the team must have a solid grounding in a unified approach/process to application development.

The Developer Experience – Building a HATS Application

A developer starts the HATS application development process by creating a new project under the HATS perspective. The following information needs to be gathered and provided to the HATS Studio before it can create a complete and working project.

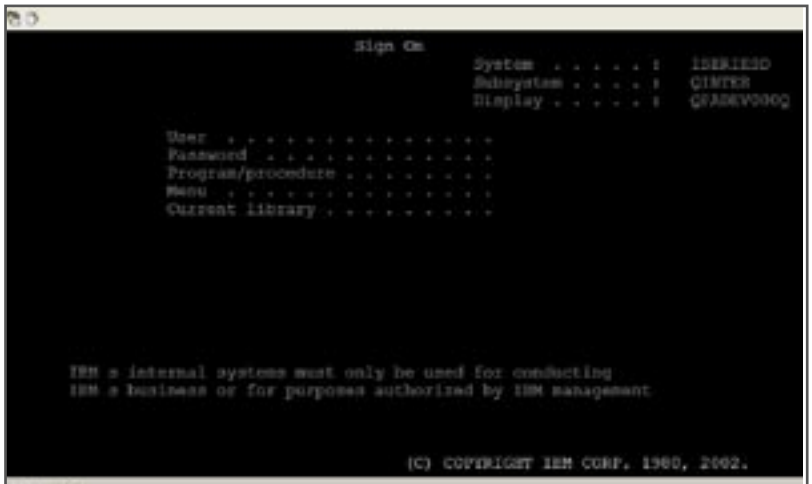


FIG 1: 5250 EMULATED SIGN-ON SCREEN

- Hosted application hostname or IP address
- Hosted application TCP/IP service port (it's typically 23 – the Telnet service port)
- The selection of a default page template for all transformation. The HATS Studio provides several examples that can be used and/or customized. (Templates are further described later in this article.)
- Type of terminal emulation (3270, 3270E, 5250, or VT)
- If SSL is to be employed for security, know the location of the certificate file.

The HATS Studio will take a few moments to generate the HATS project.

If no screen customization is to be applied, you're done. Assuming that an authentication sign-on sequence is known for the hosted application, testing can commence within WebSphere Studio (or through a deployed application URL, once you've deployed). Let me state clearly that this is never how real-world engagements begin or end. There is rather limited value for a default, screen-by-screen emulator-to-HTML transformation. Also, the standard page templates are quite generic and are not very meaningful to a given Web application (see Figures 1 and 2). Now let's dig deeper.

FIG 2: **DEFAULT HATS APPLICATION TRANSFORMATION**

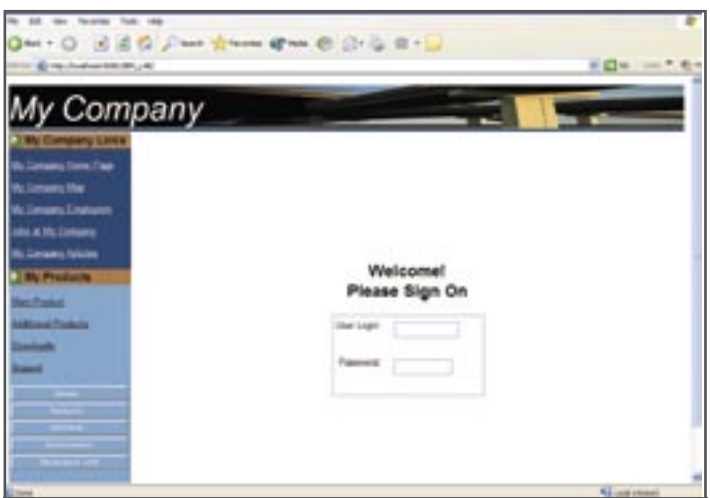


FIG 3: **SOMEWHAT CUSTOMIZED TRANSFORMATION**

HATS RESOURCES – WHAT A HATS PROJECT IS MADE OF

Once a HATS project is created, the underlying application can be systematically customized through HATS resources and custom Java code. HATS resources are identified and described below. *Note:* As is typical with WebSphere Studio perspectives, HATS resources can be managed either through easy-to-use designer views or by modifying their native persistent format. Most HATS resources are maintained as XML, JSP, or other markup language.

PROJECT CONFIGURATION

A HATS project is created and configured with many global settings such as those identified above and those that apply to the runtime transformation on the whole. Examples of such settings include:

- Existing application details – where it is hosted, what terminal emulation type is used, etc.
- The default page template.
- Global text replacement rules
- Priority ordering of screen customization
- Basic presentation attributes of HTML "widgets"

A project's configuration is stored as an XML document.

TEMPLATE

A page template defines the window dressing that is presented with each transformed screen, including:

- The basic header and footer, which might include the company name, logo, and copyright notices, etc.
- Standard navigation to other points of interest
- Style sheets that affect global presentation and behavior: colors, text fonts, mouseovers, etc.

A HATS project is always associated with a default template, but each transformation can be associated with a unique template. A template is typically stored as a JSP file, but tends to contain mainly HTML.

SCREEN CUSTOMIZATION

A HATS resource defines two things: the recognition criteria to be used to identify a given screen (or screen type), and a collection of actions to be taken when the recognition criteria is met. Examples of actions performed by a given customization include

- Applying a transformation
- Extracting or manipulating the value of a global variable
- Executing business logic in a Java class
- "Playing" a macro
- Presenting a specified Web address (URL)

The lion's share of a HATS application's runtime processing revolves around applying actions assigned to screen customizations. Each screen customization can be

applied to all or to only specific screens. Screen customizations are maintained as XML documents.

TRANSFORMATION

The action most commonly applied to a screen customization is the application of a transformation to a hosted screen. Simply put, a customization associates screen-recognition criteria with a set of actions to be taken, while a transformation defines how a screen is to be converted to Web content. Transformations are maintained as JSP content.

A close examination of a given transformation will uncover the use of a HATS-specific tag library – mainly to specify how selected areas on hosted screens are to be translated into HTML. The following code snippet is an example of a call to one such tag library, which is used to transform the user password text box in Figure 3.

```
<HATS:Component type="com.ibm.hats.component.
InputFieldExtract" widget="com.ibm.hats.widget.
TextInputWidget" row="7" col="50" erow="7" ecol="69"
label="" componentSettings="" widgetSettings="" />
```

There is always at least one transformation in a given HATS application, a default transformation that is applied when no screen customization has been triggered. This transformation performs a basic "copy" of screen content and its relative screen positioning. Text is carried over as is, and user input structures are transformed into HTML form elements.

GLOBAL VARIABLE

A global variable can be defined and used throughout a HATS application. It can be assigned a value that appears at a specific location on a hosted screen, or it can be set with a literal or programmatic value. The value of a global variable can be set and accessed from virtually every other HATS resource. Global variables are primarily used as a parking spot for values that are truly static and global, and for passing information obtained from a given transformation to another.

MACRO

A macro is used to automate some interaction between the user and the hosted application. For example, a macro can be used to automatically respond to an application prompt, thereby making that interaction transparent to the HATS application user (since that screen won't be transformed to HTML). In practice, macros are mainly used to skip through and avoid redundant screens or interactions that add no value to the use case being implemented. Macros represent one of the more powerful HATS resources.

A macro is made active by assigning it as one of the actions to be taken during a screen customization. A macro is maintained in an XML document, which defines the prompts and responses to be automated.

SCREEN CAPTURE

A screen capture is an ASCII text (XML) representation of a screen from a hosted application. Developers can create HATS application resources (such as screen customizations and transformations) for captured screens without being connected to the hosted application's network, allowing some implementation to be conducted offline.

HATS at Runtime

Here's what a HATS application does: the HATS runtime engine (class library code) executing on behalf of each deployed HATS application performs the following processing each time a user invokes the URL that requests a given hosted application screen.

First, it iterates through a list of screen customizations looking for a match in the specified screen recognition criteria.

If a match is found, the application performs custom processing as defined in an associated rule set and returns the transformed screen to the requesting user, i.e., it applies all of the actions that were assigned to the matched screen customization in the order specified, such as:

1. Play a macro
2. Extract the value of a global variable and pass it to some business logic
3. Apply a transformation

Otherwise, the application goes on to the next screen customization. If the screen customization list is exhausted without a matching criteria, default processing (the default transformation) is applied, and the screen is returned to the requesting user.

SHOW ME

The Figures 1–3 present a visual example of a HATS application at work. This HATS application was created using a sample (for testing purposes, a 5250 emulated hosted application that IBM maintains was used). Figure 1 shows how this application's sign-on screen looks through a typical emulator. Figure 2 shows a default transformation, i.e., without defining a screen customization for this sign-on screen. Finally, Figure 3 shows the sign-on screen presented after a customized Transformation was applied as part of a screen customization (which was triggered when this screen's recognition criteria was met). Notice that only the critical portions of the sign-on screen have been carried over – the text boxes for the user's login and password value. All text has been added as custom HTML content. Notice the rather generic heading and navigation areas that have been defined as part of the selected template for our project.

In the interest of saving space, this demonstration was intentionally terse. We've merely scratched the surface, but by now you should have a good idea of how HATS can be applied to accomplish the basic goal of "Webifying" a

legacy application. See the Resources section for pointers to more detailed information.

PRIOR TO IMPLEMENTATION

The good news for prospective customers is that a HATS application can be deployed relatively quickly. A working, demonstrable application can be deployed on day one – and much customization can be accomplished in a matter of days/weeks, but how can the true scope of a HATS effort be estimated? The following factors have been shown to be significant for estimating the effort and time required to design, build, and deploy a HATS application:

1. Total number of screens composing the target hosted application and the general level of complexity (e.g., depth and breadth of screen hierarchy, screen flow, form content, etc.) of the application
2. Number of required screen customizations
3. Amount of custom business or data validation logic that needs to be implemented
4. Number of varying templates needed to be designed and built
5. Degree of transformation required across the application – the number of HTML views that:
   - Use a standard transformation
   - Require simple transformation
   - Require complex transformation
6. Level of the staff's experience with HATS and related WebSphere foundation
7. Availability of hosted application subject matter experts
8. Availability of hosted application's test plan
9. Complexity of deployment scenario:
   - Portal or nonportal?
   - Single or multiple releases of WAS deployed?
   - Single or multiple instances of WAS deployed (redundancy and failover needed)?
   - User access requires VPN/firewall configuration?

Conclusion

HATS provides a very effective workbench for rapidly creating WebSphere-deployed J2EE applications that transform a line-terminal interface into an HTML interface in real time. That says quite a lot, and this column has covered only selected capabilities. As a technologist, I am very impressed with the simplicity and power of the HATS approach. From my perspective, its technical high points are as follows:

- The HATS Studio allows for very rapid development. In its simplest form, a HATS application (for demonstration purposes) can be deployed in a single day.
- HATS applications generate thin-client browser content (HTML/DHTML) portable to a range of end-user devices.
- HATS applications are built on J2EE underpinnings – making them as extendable and interoperable as they need to be.
- HATS applications are deployed under WAS – making them as scalable and redundant as they need to be.
- The HATS architecture maintains a solid separation of concerns; existing legacy remains as is, and is "unaware" of the new presentation layer provided by HATS applications.

BUSINESS VALUE PROPOSITION

For organizations currently relying on legacy systems to support their business, HATS offers a clear value proposition. By virtue of a browser or similar device, specific legacy application features can be provided to users who would truly benefit. An organization will realize benefits unique to itself, but here are some general benefits that can cut across all businesses:

- Short-term ROI
- Real reductions in operating costs
- Increased productivity and employee satisfaction
- Boosted and/or newly realized revenue streams
- Ability to create differentiation from competition by offering valuable business content to partners, suppliers, and customers.

For technologists, it should be fairly easy to demonstrate HATS as the proverbial "low hanging fruit" that it is – making a great impact in a short amount of time with available resources. In today's environment, that's a tough proposition for decision makers to ignore without some examination. HATS presents a very compelling story to line-of-business and IT managers alike.

Resources

- *WebSphere Host Integration Solution:* www.ibm.com/software/webservers/hostintegration/features.html
- *WebSphere Host Access Transformation Services:* www-3.ibm.com/software/webservers/hats
- *IBM WebSphere Host Access Transformation Server InfoCenter:* www.ibm.com/software/webservers/hats/library/infocenter
- *WebSphere Host Publisher:* www.ibm.com/software/webservers/hostpublisher
- *HATS-related Redbooks and Redpapers:* www.redbooks.ibm.com
- *IBM Global Services Training Course Catalog (SW912 – IBM WebSphere Host Access Transformation Server Introduction):* www.ibm.com/services/learning/us/catalog

Louis J. Iacona has been designing and developing software professionally since 1982. He has been employed by organizations such as Bell Laboratories and Bellcore (now known as Telcordia), and has helped start multiple new venture companies. He currently leads the design and implementation of applications deployed under the WebSphere HIS and Portal products for a New York City–based integrator. louis.iacona@verizon.net.

LOOK FOR YOUR FREE...

Coming this **SPRING!**

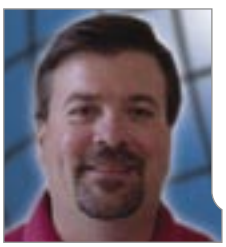
© 2004 SYS-CON MEDIA. ALL RIGHT RESERVED. ALL BRAND AND PRODUCT NAMES USED ARE TRADE NAMES, SERVICE MARKS, OR TRADEMARKS OF THEIR RESPECTIVE COMPANIES.

Bringing standards to a realm of disparity

JSR 168 – An Introduction to the Portlet Specification

BY KULVIR SINGH **BHOGAL**

& ANDREW **SWEET**



The Java Specification Request for the Portlet Specification (a.k.a. JSR 168), articulated by the Java Community Process in October 2003, aims to provide a standard for portlets that the portal arena has lacked. Portlets that are written to the JSR 168 spec will be deployable to any JSR 168–compliant portal.

The spec in essence defines a contract between a portlet and the portlet container that powers it. Areas covered by the APIs defined in the specification include topics such as aggregation, personalization, presentation, and security. As these concepts are core to the portal realm, they needed to be addressed by a spec in order to enable interoperability between portlets and portals.

A portal is a Web application that typically provides end users with personalization, single sign-on, and content aggregation from different sources. It is the responsibility of a portlet container to run portlets, providing them with a runtime environment and managing their life cycles. The container also persists portlet user preferences, enabling a customized end-user experience. IBM WebSphere Portal Server acts as a portlet container; it has an infrastructure that supports JSR 168. To the avail of developers entrusted with the task of developing portlets, you will also see tooling support via the tandem offerings of IBM WebSphere Studio Application Developer and the WebSphere Portal Toolkit version 5.0.2.

JSR 168 Lays Down the Law

Why pay attention to the specification? JSR 168 backs up its bark with some bite. Major players in the industry, including IBM, Sun Microsystems, Vignette, BEA, Plumtree, and Oracle took an active role in establishing the specification; all of these companies either already support or plan to support the spec with their respective products. Recognizing the popularity of the portal concept, the Apache Software Foundation is currently incubating a reference implementation of the Java Portlet Specification in a project called Jakarta Pluto.

Fundamentally, portlet architecture bears a lot of similarity to Java servlet architecture. However, portal developers face unique challenges such as user interface rendering restrictions and the challenge of displaying on the same page multiple portlets that may have to communicate with one another.

The standard API of JSR 168 has some significant differences from the IBM Portlet API introduced in WebSphere Portal version 4.1. These differences include the concepts of initialization parameters, portlet preferences (both user-independent and user-dependent), session state, and navigational state. You can learn more about the differences by reading "Best practices for developing portlets using JSR 168 and WebSphere Portal V5.0.2" and "Comparing the JSR 168 Java Portlet Specification with the IBM Portlet API" (see the Resources section).

One fundamental paradigm shift to note: portlets in JSR 168 do not extend servlets and their major interfaces. However, under the covers, JSR 168 still taps into much of the functionality offered through the servlet specification. Because the overall concepts of JSR 168 parallel many of those of the IBM Portlet API, developers familiar with the IBM Portlet API should experience a smooth migration.

WebSphere's Compliance

IBM took a major role in the defining of JSR 168, and WebSphere Portal v5.0.2 proudly supports the required specifications. In addition, it addresses some "optional" parts of the JSR, including expiration-based caching.

Because the JSR spec is in its first incarnation, it doesn't offer the feature-rich API customers might want to see in their portal. Accordingly, IBM offers nonstandard extensions. For instance, the only interportlet communication provided by JSR 168 is sharing of data through sessions. On the other hand, IBM WebSphere Portal offers technologies such as

click-to-action (cooperative portlets) in the WebSphere-specific toolbox. As a developer, you should keep in mind when using the extensions that the portability of your portlet is compromised by leveraging such technologies. This of course becomes an evaluation process. Does the functionality offered by nonstandard extensions provide a value-add for which you are willing to sacrifice portlet portability?

The Tools to Get Things Done

The JSR 168 Technology Preview is an extension to the WebSphere Portal Toolkit that is installed when you install Portal Toolkit v5.0.2. The extension provides a wizard as well as example code to get you jump-started with JSR 168-compliant portlet programming (see Figure 1). In its current incarnation, you can develop specification-compliant portlets within WebSphere Studio. These portlets can then be debugged remotely using WebSphere Portal Attach (a remote debug configuration).

Support for JSR 168 in WebSphere Portal Server

IBM enables its JSR 168 customers with a fixpack for WebSphere Portal v5.0.2. WebSphere Portal v5.0.2 incorporates a portlet runtime environment based on Apache Pluto (which, if you recall, is the reference implementation by the Apache Software Foundation supporting JSR 168).

Customers might ask themselves, "Does the portlet specification mean that the portal I run it in is a non-issue?" The short answer is: *no way*. IBM WebSphere Portal v5.0.2 is a J2EE application that runs on IBM WebSphere Application Server version 5. Portlets running in WebSphere Portal reap the performance benefits of the portal server's symbiotic relationship with WebSphere Application Server (WAS). The reputation of WAS precedes itself as a high-performance and extremely scalable transaction engine for powering even the most complex, dynamic e-business applications.

Also, as stated earlier, the portlet specification is still rather young and not that rich in its offerings. Enterprise organizations will typically want to leverage the extended offerings of WebSphere Portal.

Conclusion

Much of the popularity of portlets has stemmed from their major value-add of being reusable components. These components provide access to Web-based content, applications, and other content. Unfortunately, in the past, portlet reusability carried the caveat of "for use in a particular container of a particular portal server."

In a world of proprietary portlet APIs, JSR 168, the Java Portlet Specification, promises interoperability between portlets and portals. By following the specification, customers will no longer be bound to proprietary vendor APIs; they can build portlets as pluggable modules. The concept of portlet portability that JSR 168 brings to the table will most likely add to third-party vendors' motivation to develop portlets, as they can reach multiple vendors without having to code for each of their portal environments, thus further catalyzing the portal phenomenon.

Also on the Horizon

Another key specification approaching final status is WSRP (Web Services for Remote Portals), which aims to affect the portal arena significantly. We will introduce you to this specification in an upcoming article in ***WebSphere Journal.*** To whet your appetite just a bit, the WSRP specification, shepherded by the Organization for the Advancement of Structured Information Standards (OASIS), describes the use of Web services standards to integrate remote content and applications into portals.

"Why pay attention to the specification? JSR 168 backs up its bark with some bite"

Acknowledgment

The authors would like to thank Scott Karabin of IBM Software

FIG 1: THE JSR 168 PORTLET DEVELOPMENT WIZARD

Services for WebSphere for his review of this article.

Resources

- *JSR 168 Portlet Specification:* http://jcp.org/en/jsr/detail?id=168
- *IBM WebSphere Portal for Multiplatforms Version 5.0.2 Information Center:* http://publib.boulder.ibm.com/pvc/wp/502/index.html
- *Best practices: Developing portlets using JSR 168 and WebSphere Portal V5.0.2:* www-106.ibm.com/developerworks/websphere/library/techarticles/0403_hepper/0403_hepper.html
- *Comparing the JSR 168 Java Portlet Specification with the IBM Portlet API:* www-106.ibm.com/developerworks/websphere/library/techarticles/0312_hepper/hepper.html
- *JSR 168 Portlet Specification for WebSphere Portal Technology Preview:* www-106.ibm.com/developerworks/websphere/downloads/jsr168.html

Kulvir Singh Bhogal works as an IBM Software Services for WebSphere consultant, devising and implementing WebSphere-centric solutions at customer sites across the nation. He has over 50 patents pending in a myriad of technology areas. kbhogal@us.ibm.com

Andrew Sweet serves as a senior solution architect for IBM's WebSphere Services and is a frequent speaker and mentor around enterprise best practices and standards-based technologies such as J2EE and Web services. Andy also has served on the boards of several industry standards groups and technology councils, including OASIS WS-Security and UDDI, Web Services on WebSphere (WoW), IBM's WebSphere Advisory Board, and Washington University's Center for the Application of Information Technology program. asweet@us.ibm.com

FREE*CD! ($198.00 VALUE!)

Secrets of the Web services Masters

Every *WSJ* Article on One CD!

— The Complete Works —

CD is edited by ***WSJ*** Editor-in-Chief Sean Rhody and organized into 40 chapters containing more than 400 exclusive ***WSJ*** articles!

All in an easy-to-navigate HTML format! **BONUS: Full source code included!**

ORDER AT WWW.SYS-CON.COM/FREECD

**PLUS $9.95 SHIPPING AND PROCESSING (U.S. ONLY)*

©COPYRIGHT 2004 SYS-CON MEDIA. WHILE SUPPLIES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE. ALL BRAND AND PRODUCT NAMES ARE TRADE NAMES, SERVICE MARKS OR TRADEMARKS OF THEIR RESPECTIVE COMPANIES.

SYS-CON MEDIA

Only from the World's Leading i-Technology Publisher

Winners and Losers in the outsourcing game

Economic Treason: The New Future?

BY JEFF **SCHAFFZIN**

A trend begun on factory floors in an effort to replace low- and medium-skilled blue-collar workers so companies could save money, outsourcing has recently become a heated election-year issue. While it is difficult to hide plant closings and relocate their operations, it takes only a few clicks of a mouse on a virtual ledger to lay off hundreds, if not thousands, and hire 10 times as many more overseas. Who really benefits from this? Despite the reports from executives who swear by outsourcing, the answer is not as obvious as you might think.

In the beginning, no one – from the controllers and their financial analysts to the "nameless, faceless drones" on the manufacturing floor – really knew this would happen. Then one day someone realized it would be cheaper to take manufacturing jobs and send them elsewhere. Besides, the people here could always learn something else – the next big thing would always be around the corner. The result was devastating.

Families were forced to sell their homes, and countless lives were ruined. Many people still have not fully recovered from this attempt to "offshore" their work – work that can be done here just as efficiently as "there," wherever "there" is.

Fast-forward to the relatively recent past. Many people in various parts of the U.S. took part in one of the largest economic expansions in history, known as the "dot-com boom." Companies employed many people, each with huge amounts of heavily diluted stock options that may or may not have had a chance for success.

To feed this frenzy, companies claimed they needed even more people and thus created special visa programs to hire a constant stream of competent talent from overseas to make sure their widget or software application was better, faster, or cheaper than their competitor's. When the bubble finally burst in the early part of the 21st century, many of these companies went bankrupt, merged with other companies, or limped along into the present day.

Those that managed to hang on did grow, but had issues with showing it to the world. Some of these companies decided to lay off entire divisions and claimed huge savings, hoping that their stockholders wouldn't read their 10-Ks. Many of these companies showed that they were profitable, but quietly enlarged their operations overseas.

At least one U.S. CEO has gone on record claiming that the U.S. workforce is not smart enough, which justifies the need to go overseas for labor. When we hear something like this, we need to be very skeptical. Many of the people who work in the IT field are highly educated and need to be aware of what is going on around them. The question remains: Who are the winners and who are the losers of this outsourcing game?

- ***The American employee (loser):*** American employees lose in both the short and long term because they do not get an opportunity to defend themselves. If things do not change, they will continue to lose their jobs, not just to someone overseas, but to a CEO who is too shortsighted to see the big picture.
- ***The CEO (winner/loser):*** The CEOs are winners in the short term but losers in the long term. While CEOs may be winners for now because they gain personal wealth, they mortgage their future – eventually they will have nothing left to sell after the company (or companies) to which they have outsourced owns everything.
- ***Board members (winners/losers):*** Same as CEOs; board members are little more than a set of pawns in this game, because they don't have much say in what is going on. However, they should stand up and do what is right for their companies – and more important, the country – not just what is right for their wallets.
- ***Outsourced employees (unknown/winners):*** While many writ-

ers have stated that, thanks to American companies, many of the countries that are home to offshore employees are developing a middle class, it is unclear to me whether this is something we should feel responsible for. If other countries do this along with us, I am fully in support, but we should not be going it alone (or mostly alone). Further, this is not to say that I am a protectionist or an isolationist – we do need equality between the U.S. and other countries – but will these countries become like the animals in George Orwell's Animal Farm – where some are more equal than others? In the long term, they are truly the winners because at the current pace, companies will give them all of the technology, leaving us with nothing to work with – which is a very scary proposition.

So where does that leave us? According to a story I heard on the news, 30 states are working on legislation to restrict or punish companies that outsource. However, the question is, to what degree? Will it be a slap on the wrist or will it be something punitive? For the benefit of the whole U.S., it clearly needs to be the latter.

In general, most companies have looked only at their short-term profits as an indicator of whether or not to outsource. While this may be okay for an individual company, when everyone does it, it creates the crisis we have today. We have transformed our economy from one that is manufacturing based (where we create value) into one that is service based (where we simply move money from one pocket to another). The foreign countries that do the manufacturing for us will reap the benefits in the end.

There is no reason why companies can't do manufacturing here. People can be trained; assembly lines and factories can be built. Further, incentives can be put in place to get people interested in software development and IT positions in the medium and long term. It is very important that companies know that this is a very serious issue – not just an election-time phenomenon. Outsourcing truly is criminal!

Jeff Schaffzin is an IT industry veteran who has worked for a number of leading software companies in such segments as relational databases, product life-cycle management, business intelligence, and application servers. Jeff has served as a courseware developer/trainer, applications engineer, and in product marketing/management positions. He currently works as a writer and a marketing consultant in the enterprise software industry.
j.schaffzin@comcast.net

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

SYS-CON MEDIA

WebSphereDevelopersJournal.com

WebSphere DEVELOPER'S JOURNAL

Introductory Charter Subscription

SUBSCRIBE NOW AND SAVE $31.00 OFF THE ANNUAL NEWSSTAND RATE

ONLY $149 FOR 1 YEAR (12 ISSUES) REGULAR RATE $180

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

Do You Have Access to the Internet?

The World's Leading Independent WebSphere Developer Resource

Then Subscribe Online and Save $31!

It's that easy.

Put the power of partnering to work for you

Five Steps to Successful Strategic Partnering

BY BUELL **DUNCAN**

The ability to develop and nurture strategic partnerships can make the difference between success and failure in the high-tech industry. Just ask independent software vendors (ISVs), a segment of the industry that earns more than 40% of its revenue through successful partnering. ISVs have created an entire "ecosystem" of partners that opens up new opportunities and revenue streams.

The pace of innovation today is too fast for any single IT company to be all things to all customers. Last year alone the U.S. patent office awarded more than 16,000 patents to the top 10 global high-tech companies. Even a brief look at the industry's history reveals a graveyard of once-successful companies that failed to adapt fast enough to industry changes. Despite its long record of success, IBM suffered a near-death experience in the early '90s. New leadership and a new strategy were instrumental in engineering IBM's turnaround – and so was the power of its alliances with more than 90,000 business partners.

Partnering offers a company the power to win with a world-class team. In baseball, winning teams know the value of fielding the best players at every position. A star player may win individual games, but it takes the power of the whole team to win a World Series. The same is true for an IT company that needs to compete every day. Working with partners offers the stamina and flexibility of an all-star team. You can offer a broader range of IT solutions, maximize the value of your customer's investment, and help ensure market success.

How can you develop relationships with partners to increase your competitiveness and win in the marketplace? What have companies like yours done to create strategic partnerships that work? Following are five factors that lead to successful alliances.

First, make the strategic decision to partner at the highest executive level, and secure buy-in from all levels of employees, especially the folks who interact directly with customers. Lori Schafer, founder and CEO of Marketmax, decided partnering was the path to growth for the Wakefield, MA–based company, which sells software to large retailers, including Home Depot and Kohl's. She regularly attended trade shows and followed up with monthly personal phone calls to develop the partnerships that helped Marketmax accelerate revenue growth, generate industry awareness, and differentiate itself from competitors.

Second, determine if a potential partnership is the right match. Share your business strategy, understand each other's core competencies, and check synergy in goals, technology, and target markets. Donald Doane, CEO of Newark, NJ–based OpenDemand Systems, which provides rapid performance optimization software, has a litmus test for partnering. He finds that his company's successful partnerships create benefits for customers that neither partner can deliver alone. Together, the partners reach new markets and expand their revenue bases. At IBM, for example, we understood that building our own software applications was not our core competency. Small, entrepreneurial software companies do a much better job. So we choose to work with ISVs that know how to create best-of-breed applications, and we provide middleware for the e-business infrastructure.

Third, remember that an alliance is a formal business agreement, not just a handshake over lunch. Get a contract in writing to avoid misunderstandings, build in rigorous commitments on both sides, and have clear measurements for those commitments. Unfortunately, some would-be partners may view an alliance as a way to get sales leads rather than as a joint effort to drive new opportunities. OpenDemand's Doane says partners need to share both the risks and the rewards. Have specific requirements for resources, such as training and technical and marketing support. In addition, have clear revenue targets and regular meetings to monitor progress.

Buell Duncan is general manager of IBM Developer Relations, Software Group. He is responsible for IBM's worldwide relationship with independent software vendors and corporate developers, a strategic growth initiative in which IBM partners with rather than competes with developers of business application software. Prior to this appointment, Buell was general manager, IBM eServer iSeries, a business with 250,000 active customers worldwide.

Fourth, be clear that you may still compete with a partner in some areas, while collaborating in others. IBM, for example, competes directly with Microsoft in some areas, but we also cooperate on new standards, and make certain that IBM products run on the Windows platform, as well as other platforms. You may compete with someone today who you might partner with tomorrow. Your business strategy will determine when you partner and when you compete.

Fifth, follow through. The most common reason alliances fail is neglect. Both partners need to put in the time and resources to make the relationship work. Regular meetings help track results. In addition, to keep an alliance healthy and profitable, make an individual executive responsible for results, and put an infrastructure in place to support joint business development and deal with issues as they arise. Having a formal process in place to renew or exit a partnership is valuable. Partnerships need to evolve with market conditions, and be flexible enough to be transformed when necessary. But if both partners decide it makes business sense to exit, then a well-executed plan can save time, capital, and human resources. "If you're not seeing revenue within the first six months, then you're not getting traction, and it may be time for both parties to say good-bye," recommends Sandy DeFelice of Marketmax.

These five steps can help put the power of partnering to work for you. With the right partners, you can offer customers more choices, best-in-class technology, and world-class service, which are the best plays to execute when you want to compete and win in the high-tech industry.

–continued from page 4

In Silicon Valley and other technology hot spots, venture-funded Sarbanes-Oxley software companies are beginning to appear with increasing regularity.

A good example is Nth Orbit, which is offering a product called Certus that provides a systematic approach to compliance. Their lead investor is Sequoia Capital. Sequoia was an early investor in and worked with Cisco Systems, Yahoo!, Redback Networks, Google, Network Appliance, Cypress Semiconductor, Vitesse Semiconductor, Apple Computer, and Oracle. These are not stupid people! Merger and acquisition activity is also beginning in this space, exemplified by EMC's recent purchase of Documentum. And there is an entire magazine dedicated to Sarbanes-Oxley compliance – the *Sarbanes-Oxley Compliance Journal* (www.s-ox.com).

The Sarbanes-Oxley Act is changing the way the business world operates. High-quality staff, automation, and processes will be a must-have for all public companies. The long-term payback will ultimately be a significantly higher level of awareness and controls that will produce more stringent business processes throughout business units reporting to their corporate parents.

Sarbanes-Oxley will make the astute programmer, system administrator, and CIO indispensable within their organization. These positions cannot be rationally offshored or outsourced; the personal risk to the people running the corporation is too high. The requirement to attest that the systems are working as intended and described is so intense that only a madman would send this work halfway around the world to save a couple of bucks.

WebSphere Journal | Advertiser Index...

COMPANY	URL	PHONE	PG
BOWSTREET	WWW.BOWSTREET.COM/EASY	978-863-1500	52
CLEARNOVA	WWW.CLEARNOVA.COM/THINKCAP		27
CYANEA	WWW.CYANEA.COM/WSDJ/NOMOREMADNESS/HTML	877-CYANEA8	17
H&W COMPUTER SYSTEMS	WWW.HWCS.COM/WSJ4.ASP	800-338-6692	3
INFERDATA	WWW.INFERDATA.COM		9
ISSJ	WWW.ISSJOURNAL.COM	888-303-5282	35
IT SOLUTIONS GUIDE	WWW.SYS-CON.COM/IT	888-303-5282	41
MERCURY INTERACTIVE	WWW.MERCURYINTERACTIVE.COM/OPTIMIZEJ2EE	800-837-8911	23
PROLIFICS	WWW.PROLIFICS.COM	800-675-5419	4
QUEST SOFTWARE	HTTP://JAVA.QUEST.COM/JDJ_JAVA	800-663-4723	51
VERSATA	WWW.VERSATA.COM/BUSINESSLOGICDESIGNER	510-238-4100	31
WDJ	WWW.SYS-CON.COM/WEBSPHERE	888-303-5282	47
WEB SERVICES JOURNAL CD	WWW.SYS-CON.COM/FREECD	888-303-5282	45
WILY TECHNOLOGY	WWW.WILYTECH.COM	888-GET-WILY	5
WINMILL SOFTWARE	WWW.WINMILL.COM	888-711-6455	21

WebSphere Journal | Coming Next Month...

MESSAGING
JMS Support on the WebSphere Platform
BY COLIN YU

WAS SECURITY
Securing Access to the WAS Namespace
BY STEPHEN PIPES

PLATFORMS
IBM's Websphere Application Server vs Microsoft's .NET
BY DWIGHT PELTZER

PERFORMANCE TUNING
Optimizing Code to Minimize the Impact of Heap Management
BY MIKE STOREY

JDBC
Best Practices in Java Database Connectivity
BY DAVID MURPHY

Application Quality Is Vital

BY STEVE STOVER

I have noticed a need for a back-to-basics movement in the quality of J2EE and Web-based applications. Poor software development life-cycle processes, combined with the complexity of the J2EE technology stack, are resulting in IT applications that are too slow, fragile, or that are not scalable or maintainable. That's just not going to cut it. Problematic business applications mean loss of revenue for the business in one way or another, which ultimately means bad things for IT organizations (and you), one way or another.

Back in the days of monolithic "green screen" applications, there wasn't a lot between the application code and the OS, and those applications were, arguably, less complex. Now we have Internet architectures and the distributed J2EE technology stack, which enables much more complex applications to be built relatively easily, but also brings new and not-always-known dependencies to the system. Complexity and abstraction.

We really cannot get by without proper quality assurance and testing now. Most of us are aware of best practices to some degree, but in the rush to deliver more with less, we've often given them short shrift. Ensuring the quality of J2EE applications requires both developers and QA/test engineers to assume responsibility for some basic testing. It doesn't hurt to remind ourselves about these activities now and then.

Development Test Activities

Developers, responsible for creating the application set out by the requirements, need to test to ensure that their code is functionally correct, maintainable, and extensible.

A library of unit tests is an insurance policy against future changes breaking functionality in other areas. Performing code coverage of unit tests gives developers insight into the completeness of the unit test cases. Don't set a 100% code coverage goal initially, or you may spend more time writing test code than functional code.

Developers need to ensure that common coding mistakes are avoided. Code reviews and coding standards help ensure code correctness, maintainability, ease of understanding, and extensibility – automation of the process can increase productivity as well as quality.

The J2EE component model and service-oriented architectures have certainly made developers more productive. However, component-based development often occurs in a system-wide performance vacuum.

Developers should load-test their components and perform diagnostics to ensure response times and proper utilization of resources such as memory. A complete understanding of the performance characteristics of their components will help later during integrated performance testing.

QA Test Activities

Test engineers, responsible for verifying that the application meets requirements, test to ensure correct functionality and acceptable performance.

Functional and regression testing have long been staples of QA test activities. Automating regression testing is vital to productivity. Coupled with code coverage analysis, this can point out gaps in test-case coverage as well as uncover potential dead code that should no longer be maintained or unit-tested.

Beware of the temptation to forgo functional testing in favor of unit testing. Unit testing gives insights at the *class* level only, rather than into the application use cases. Skipping functional testing misses a key aspect of overall application quality.

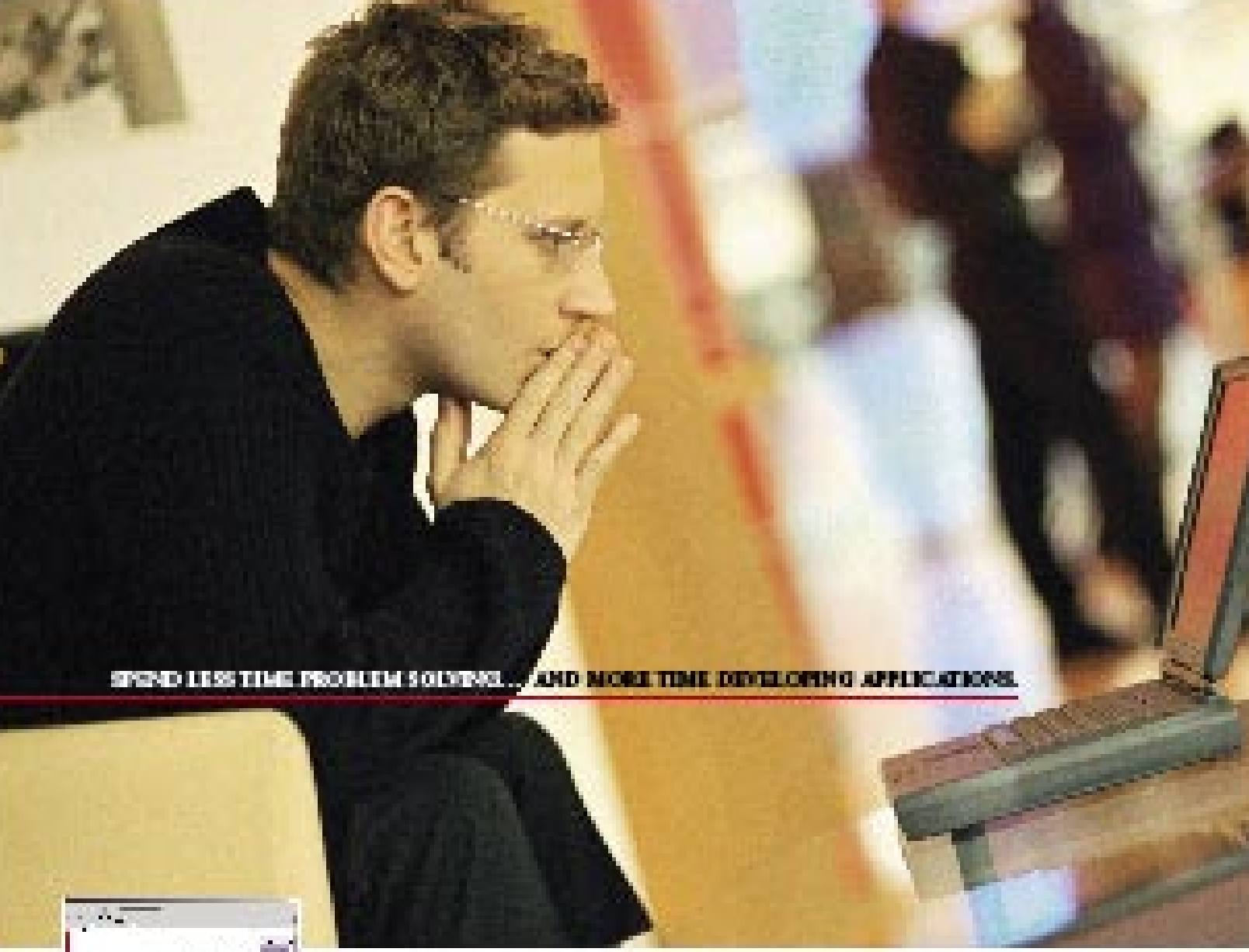
Mainly the province of QA/test engineers, performance testing of the assembled application under load is vital for everything from ensuring SLA compliance to diagnosing problems that will crop up only when the application is under real-world user load. Developers (and database administrators) can and should become involved here; their expertise can help diagnose and resolve performance bottlenecks.

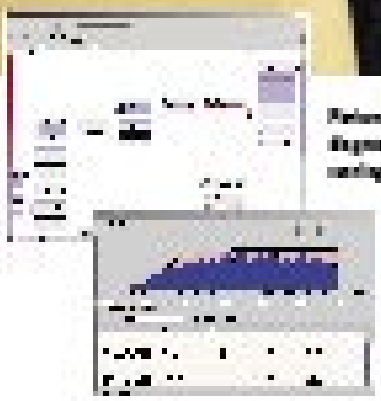
Create a performance acceptance test that includes not only response times but metrics for your other team members so they can quickly resolve problems. This data includes, but is not limited to, application server, method performance, operating system, and database metrics.

Why Bother?

Application confidence is something we all should strive for. Poor application quality is a risk to the business; it can easily drive away customers. Basic quality-assurance activities are an important investment to help ensure the success and longevity of J2EE and Web applications – and are vital when you consider the complexity and depth of today's J2EE applications.

Steve Stover is chief technology officer for Quest Software's Development Solutions. Steve has nearly a decade of experience in developing and architecting enterprise applications, including six years focused on Java and J2EE. Steve has conducted seminars and consulting engagements on architecting high-performance and scalable J2EE applications. steve.stover@quest.com





Join The Thousands of Companies Improving Java Application Performance with JProbe and DevPartner.

Whether it's a memory leak or other performance issues, Quest Software's award-winning Java products — including JProbe and DevPartner — help you spend less time trouble-shooting and more time on the things that matter. Quest's Java tools will identify and diagnose a problem all the way down to the line of code, so you no longer have to waste time pointing fingers or guessing where the problem lies. Maximize your team's productivity with Quest Software by downloading a free trial today from **http://www.quest.com/jb/j_jpro**.

QUEST SOFTWARE

© 2000 Quest Software, Inc.

Build WebSphere Portlets FAST*

**and change them even faster.*

Out-of-the-box portlets aren't enough.
But, you already know that.

Driving directions, stock quotes, and news are nice, but they're not what you need to run your business. What you need are customizable portlets that bring your enterprise IT assets to your enterprise portals.

Bowstreet™ Portlet Factory is the first product to automate the rapid development, deployment, and maintenance of highly adaptive portlets that plug right into the IBM WebSphere Portal stack.

BOWSTREET PORTLET FACTORY
TRY IT FREE FOR 60 DAYS.
WWW.BOWSTREET.COM/EASY

J2EE IS HARD. WE MAKE IT EASY.