SYS-CON

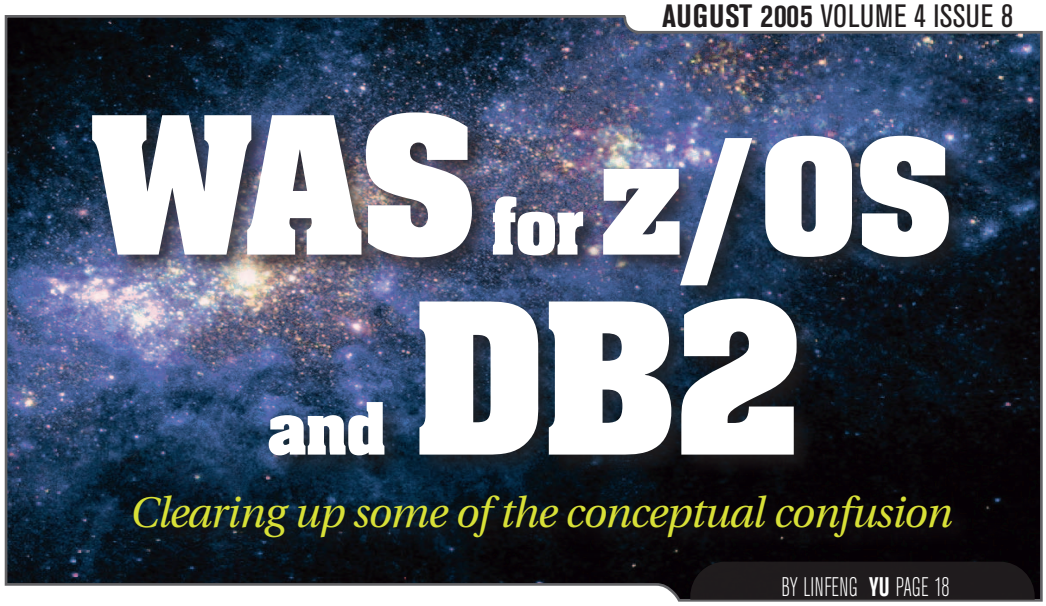
TESTING • INTEGRATION • WEB SERVICES

WebSphere® JOURNAL

www.SYS-CON.com/WebSphere

The World's Leading Independent WebSphere Resource

AUGUST 2005 VOLUME 4 ISSUE 8



NEW!

CodePro AnalytiX™

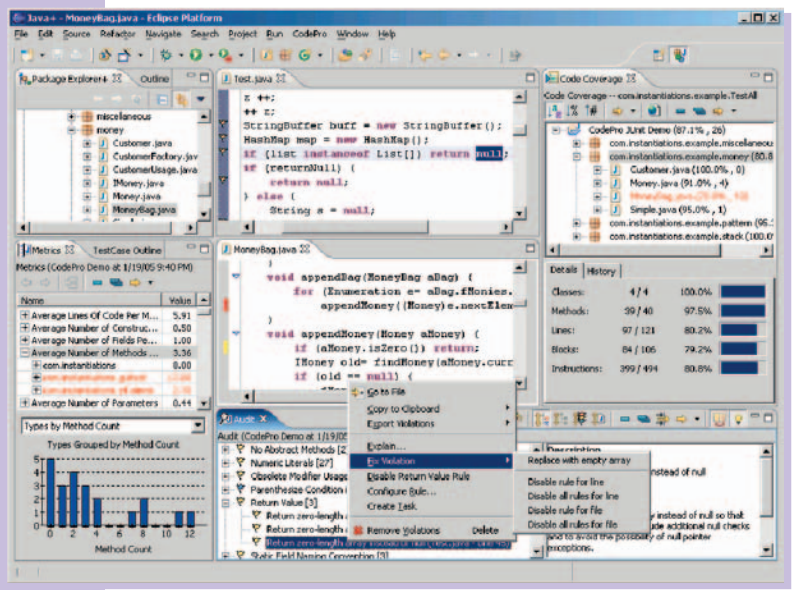
Continuous Collaborative Code Analysis for Rational®, WebSphere® & Eclipse

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

Sound like a fantasy?

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

Your money back.



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

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

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

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

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

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

Put our experience to work for your team.

www.instantiations.com

1-800-808-3737

Key Product Capabilities

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

Risk Free Trial Copy

www.instantiations.com/analytix

eclipse READY · IBM Business Partner · eclipse FOUNDATION MEMBER

instantiations

Movin' On Up

time to migrate to the next generation of WebSphere

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® IBM Business Partner

WebSphere Continuing to Lead

BY ROGER STRUKHOFF

Unification, integration, clustering, connectivity. These are words found commonly in enterprise IT and in this issue of WebSphere Journal. Picking up on last month's Non-Theme theme, we continue to provide you with a variety of articles that cover many parts of the application development ground, all made possible through the use of WebSphere.

As stated here previously (and obviously), we live in times of tremendous heterogeneity in most enterprise IT departments. We also live in times of industry rhetoric that borders on the political, sometimes with religious-sounding overtones. Whether an environment or approach is "open" is subject to severe debate, most of it self-serving.

Yet beneath the rhetoric lies the technical reality of designing, deploying, and maintaining applications, and IT developers and managers tend not to listen to the hype but simply ask what a technology can do and how well will it be supported.

We were reminded again recently of WebSphere's burgeoning dominance in many aspects of the markets it serves, with the release of an IDC report that "concludes that IBM Global Services is a leading global market maker for Service Oriented Architectures (SOA)."

Although this report's overview did not provide any market-share information, it pointed out that IBM's "leadership in SOA is attributed to a company-wide strategy, deep vertical industry expertise and Service Oriented Modeling and Architecture (SOMA), a unique methodology that aligns an SOA to a customer's business goals and directly ties business processes to underlying software applications that support them."

Although WebSphere and SOA, let alone application development and SOA, are not interchangeable terms, it is interesting to note that Blue has taken the lead in a very hot, very leading-edge of the market against serious competitors and an open-source community that is moving aggressively to tout itself as the answer to enterprise IT in the 21st century.

Combine the results of the IDC study with results reported by Gartner and others that show WebSphere taking the clear leadership position in the web services application development market worldwide, and you can see that IBM is helping enterprises make as much sense as possible out of the tasks developers face.

The polyglot nature of enterprise IT in most companies can give the impression that things are more messed up than they are. Because, despite the diversity of platforms, operating systems and environments, and application development approaches, it looks as if a significant number of people have found a solution that works for them within the world of WebSphere.

This issue of WebSphere Journal presents some pieces of the overall puzzle, as does every issue of the magazine, and we hope you enjoy working your way through the serious and eclectic variety of features presented here. As always, send us your story ideas, your stories (good and bad) about working with WebSphere, and your comments on the issue.

Roger Strukhoff, editor-in-chief of ***WebSphere Journal***, is West Coast Bureau Chief for the SYS-CON News Desk, and President of www.wdva.com. He spent 15 years with Miller Freeman Publications and The International Data Group (IDG), then co-founded CoverOne Media, a custom publishing agency that he sold in 2004. His work has won awards from the American Business Media, Western Press Association, Illinois Press Association, and the Magazine Publishers' Association. Read his blog at http://www.rssblog.linuxworld.com. Contact him at roger@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 ROGER STRUKHOFF
CONTRIBUTING EDITOR PATTI MARTIN
PRODUCT REVIEW EDITOR JAY JOHNSON
EDITOR NANCY VALENTINE
ASSOCIATE EDITOR SETA PAPAZIAN

WRITERS IN THIS ISSUE
Chris Lockhart, Serge Lucio, Tilak Mitra, Svetlana Petrova, Ade Rixon, Roger Struckhoff, Linfeng Yu

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

PRESIDENT AND CEO FUAT KIRCAALI
PRESIDENT, SYS-CON EVENTS GRISHA DAVIDA
GROUP PUBLISHER JEREMY GEELAN
SENIOR VP, SALES & MARKETING CARMEN GONZALEZ
VP, INFORMATION SYSTEMS ROBERT DIAMOND
VP, SALES & MARKETING MILES SILVERMAN
PRODUCTION CONSULTANT JIM MORGAN
FINANCIAL ANALYST JOAN LAROSE
CREDITS & ACCOUNTS RECEIVABLE GAIL NAPLES
ACCOUNTS PAYABLE BETTY WHITE
ADVERTISING DIRECTOR ROBYN FORMA
ADVERTISING MARKETING DIRECTOR DENNIS LEAVEY
ADVERTSING SALES MANAGER MEGAN MUSSA
ADVERTISING SALES MANAGERS DOROTHY GIL
KERRY MEALIA
LEAD DESIGNER ANDREA BODEN
ART DIRECTOR ALEX BOTERO
ASSOCIATE ART DIRECTORS ABRAHAM ADDO
TAMI BEATTY
LOUIS F. CUFFARI
VIDEO PRODUCTION FRANK MORICCO
WEB DESIGNERS STEPHEN KILMURRAY
VINCENT SANTAITI
SHAWN SLANEY
ONLINE EDITOR ROGER STRUKHOFF
CIRCULATION SERVICE COORDINATORS EDNA EARLE RUSSELL
LINDA LIPTON
CUSTOMER RELATIONS MANAGER BRUNI STAROPOLI

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

WORLDWIDE NEWSSTAND DISTRIBUTION
CURTIS CIRCULATION COMPANY, NEW MILFORD, NJ

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

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

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

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

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

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

Get Wily.™

1 888 GET WILY | wilytech.com

wily technology

Message-Driven Beans in WebSphere

Use MQ Messaging for Many Processes

BY JONATHAN FORCK

Jonathan Forck is a J2EE developer who has done key work on several online banking applications developed for a group of banks. Jonathan has written his own implementation of the security interface for WAS 5.1 and is currently working on converting 3270 applications to an Eclipse RCP.
jonforck@gmail.com

With WebSphere Application server 5.0+ you can now add Message-Driven Beans (MDBs) to your WebSphere arsenal. These beans are handy for many processes. I recently had to write a MDB using MQ Series. WebSphere Application Server (WAS) 5.0 had the same functionality built-in, but I had a hard time learning how to use it. I also encountered some issues while configuring it to run in a load-balanced environment.

First of all, there's no real way of balancing the load in a clustered environment for MDBs. All nodes in the cluster will fight for the messages as they enter the queue. This is a limitation that can't be worked around.

There's another pitfall that will affect you even if you don't plan to use load balancing. If for any reason the queue manager connection fails, your MDB will be rendered useless until someone restarts the message listener service on the application server.

That said let's build an MDB. First you'll want to get some information about the queue you're attaching to. The Queue Manager name, Host, Port, Channel name, and Queue name will most likely be enough information. You'll have to create both a Destination and a Connection Factory.

To create a destination you'll want to go to the WebSphere Console and go to: Resources – WebSphere MQ JMS Provider – a WebSphere MQ Queue Destinations. At this point you'll want to decide at which level you'd like this information to be available. If you're in a load-balanced environment there's only one that will allow all the cells to play nicely together, and that's the cell level. Choose a JNDI name and note it for later use. Then populate the appropriate queue info on the next screen.

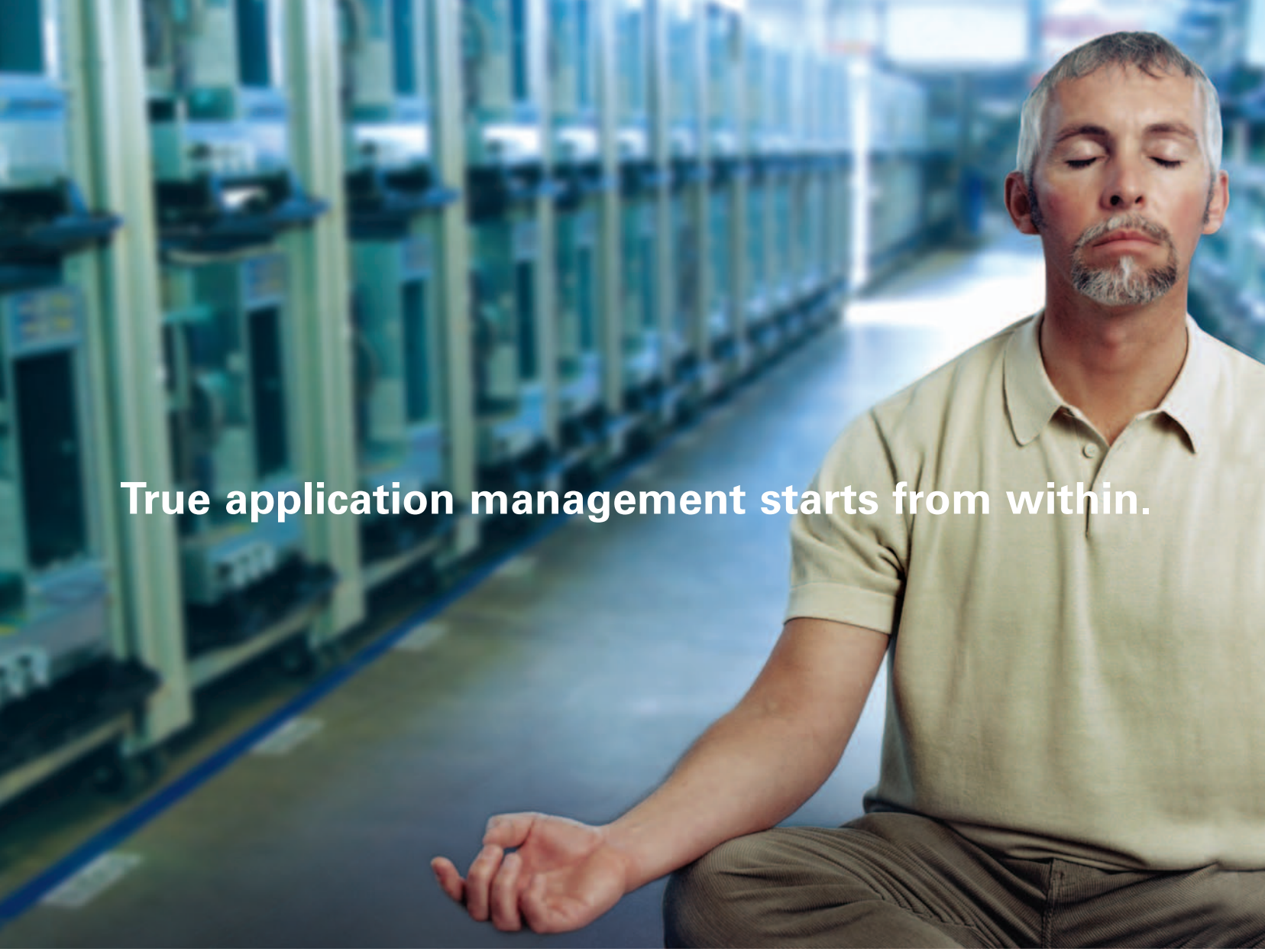
Once you've done that we'll have to create the Queue Connection Factory. This is found at: Resources – WebSphere MQ JMS Provider – a WebSphere MQ Queue Connection Factories. Here again, if you're in a load-balanced environment, make sure to create these at the cell level. Choose a JNDI name and note it for later use. Then, on the next page, populate the queue information.

At this point we're ready for the final piece of the puzzle, we have to create a Listener Port. All this really does is take the factory and bind it to the destination, and then manage the two. To create a port, go to Application Servers – (your server's name) – Message Listener Service. Here you'll create a port and assign it a JNDI name. This JNDI name will be used by your application to talk to the port. The guts of the port is pretty straightforward, just give it the JNDI names for the Connection Factory and Destination. The rest should work with the default values. Once you have that created, go back to the Listener Ports screen.

It's here that you'll probably want to add a little extra info in case your queue manager ever needs to be rebooted or taken down for short periods. We're going to add two custom properties: "MAX.RECOVERY.RETRIES" and "RECOVERY.RETRY.INTERVAL." This will let the Listener Port retry for the specified number of tries, and wait for however long you've set the interval to between tries. The interval value is in seconds.

I've created a simple MDB that will print the contents of a message to the console. This would be a good app to start your testing with. It can be found with this article online at http://websphere.sys-con.com/ (*WebSphere Journal* archives Vol: 4, Iss: 8).

"WebSphere Application Server 5.0 had the same functionality built-in, but I had a hard time learning how to use it"



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

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

motive

www.motive.com

Testing IFX Messaging Middleware with an Open Source Test Tool – Part One of a Two-Part Article

WebSphere System Integration Testing with OpenSTA

BY: GREG HERRINGER

Greg Herringer is an IT Architect with 15 years of experience in customer relationship management and contact center technologies, with a focus on the financial services and public sector industries. His background cuts acoss the entire application development lifecycle.
gherring@ca.ibm.com

This article describes how the Open Source tool OpenSTA facilitated the system integration testing of a middleware solution based on IBM WebSphere Application Server and the Websphere Branch Transformation Toolkit (BTT).

OpenSTA automation improved test execution accuracy and reproducibility while preserving the project's tight test budget. With the benefit of the lessons learned and the key OpenSTA workarounds documented here, you may want to consider OpenSTA for your own distributed application testing strategy.

Why an Open Source Test Tool?

A recent project for a financial institution delivered a middleware infrastructure to support an Interactive Financial Exchange (IFX) messaging interface between line-of-business applications and the business's core financial systems. Figure 1 illustrates the middleware architecture, with the components in bold representing the initial release and future components in grey.

The project team had to devise a method for testing the integrity of the transactions passing through the middleware to satisfy both the system integration testing criteria and the demands of the corporate audit department. The project sponsor wanted the testing approach selected to be repeatable and extensible for testing future additions in the middleware's transaction inventory. Besides these challenges, the project had no budget for buying commercial testing tools. Without a test tool to structure, automate, and report on test case execution, it would be very hard to provide reproducible results and meet the testing deadline for the hundreds of test cases defined for the project.

With a tight budget, a looming deadline, and no test automation tool, it was clear this was a chance to explore what Open Source had to offer.

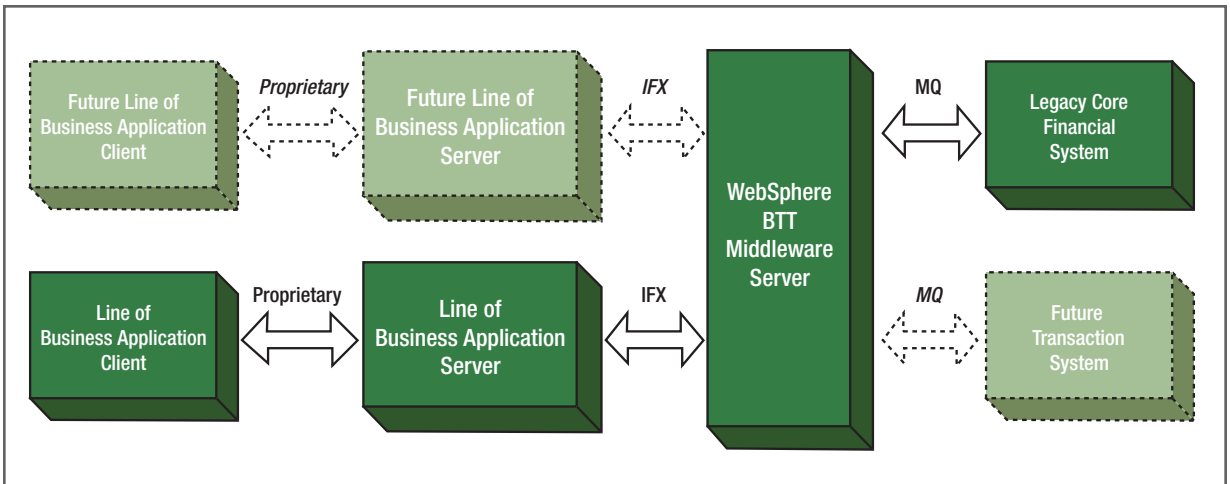


FIG 1: MIDDLEWARE ARCHITECTURE

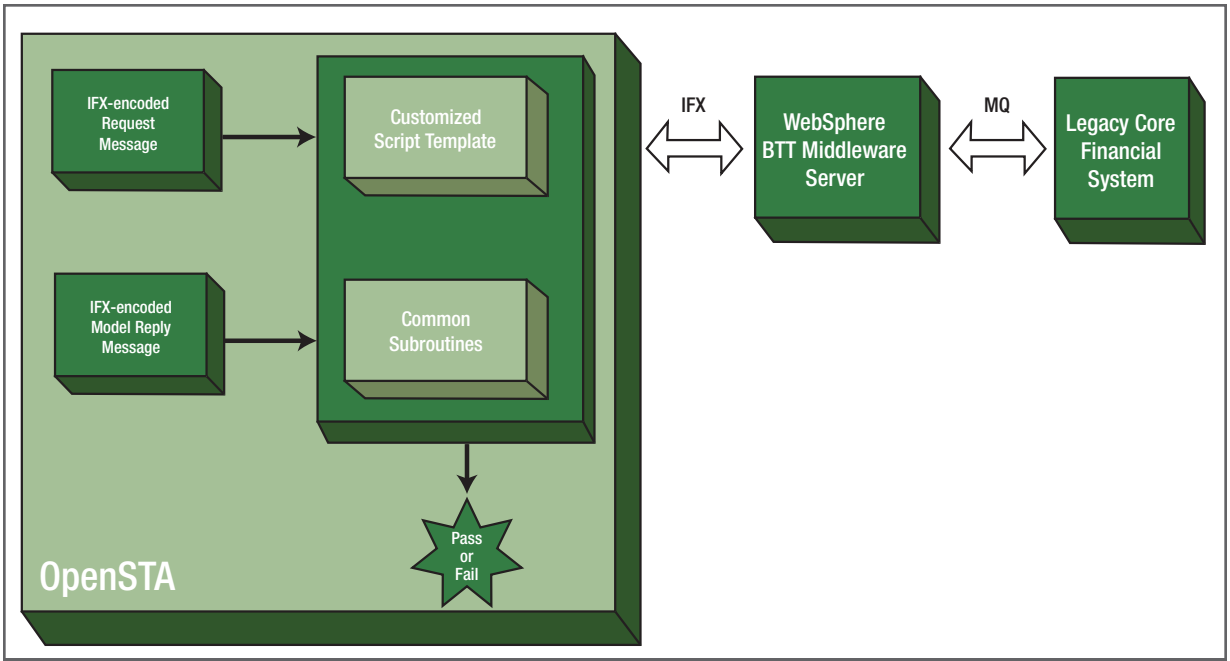


FIG 2: OPENSTA SCRIPT ELEMENTS DEVELOPED FOR MIDDLEWARE TESTING

Selecting OpenSTA

Research on the available Open Source testing tools revealed that OpenSTA would give the team a no-cost testing platform with a rich set of features to support the middleware system integration testing requirements. OpenSTA was originally developed as a commercial product but has since been open sourced. Because of its commercial heritage, OpenSTA offers a more polished user interface than other Open Source tools, comprehensive help, and an online reference for its proprietary Script Control Language.

For this particular project, a key feature of OpenSTA was its Script Control Language, which let the team execute a number of routines that were reusable for each test case. This feature, combined with OpenSTA's ability to report on test case results including the IFX-encoded response, helped the test team meet its testing schedule. Unfortunately no tool could reduce the work involved in collecting test data for the hundreds of test cases defined for the project.

Designing the Test Scripts

The object of the system integration test was to ensure that the middleware and legacy system interoperated and that IFX-encoded requests and the resulting IFX-encoded responses accurately reflected the results of the legacy system's transactions. The system integration test cases specified the request message content and expected response message content for a variety of business scenarios and test case conditions. The OpenSTA test scripts were designed to meet these test case criteria.

To support the test case execution, a set of IFX-encoded requests was prepared with test data from the legacy system's test region. A complementary set of IFX-encoded responses, one each for the expected results of the IFX-encoded requests, was also developed. So each test case was represented by a request/response pair. It should be noted that the design and implementation of these messages was completely independent of the test tool.

OpenSTA scripts were then developed so each test case would:

- read the IFX-encoded request
- read the expected test results in the form of an IFX-encoded response (referred to as a model response)
- generate a unique message identifier (RqUID) as required by MQ per the project's IFX specification and update both the IFX-encoded request and IFX-encoded model response with the identifier
- submit the IFX-encoded request to the middleware for processing
- capture the actual IFX-encoded response from the middleware
- depending on the test case, remove the time-sensitive elements (timestamps) from the actual IFX-encoded response and the IFX-encoded model response
- compare the actual response with the model response: if identical, flag the test case as passed; otherwise, flag the test case as failed

Since much of this test script logic was common to all the test scripts, it was decided that the test scripts would be written in a modular format in a basic mainline script calling OpenSTA's built-in commands and custom subroutines as required.

A Modular Approach

OpenSTA provided the architec-

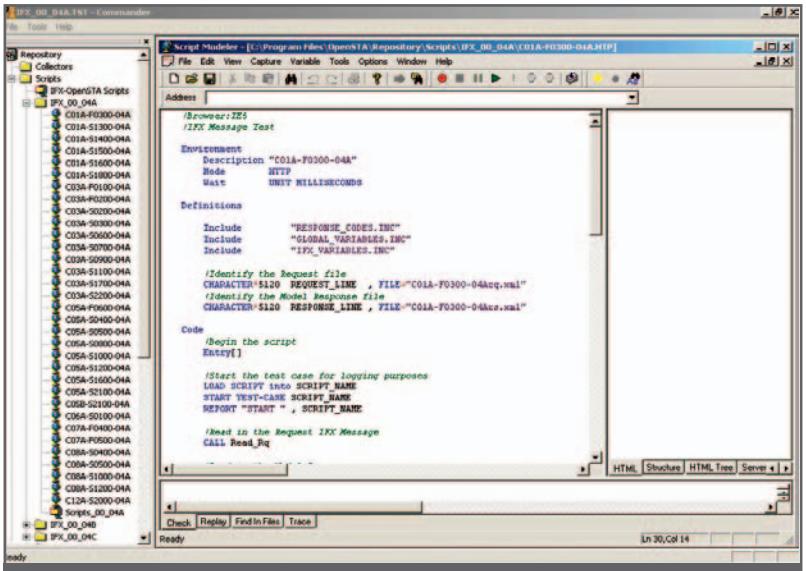


FIG 3: OPENSTA SCRIPT MODELER WITH CUSTOMIZED TEST SCRIPT

ture to design modular scripts so that common logic could be reused for each test case category. After some experimentation, a scripting approach was developed using four script elements:

- a single test script template slightly customized for each test script
- a set of common subroutines referenced by each test script as required
- an IFX-encoded request message for each test script
- an IFX-encoded model response message for each test script

Figure 2 illustrates how these elements interrelate.

In this project's OpenSTA deployment, each test script referenced two executable components: the customized script template that acts as the scripting mainline function and the common subroutines script file that provides common logic across test cases. When executed, the test script reads the IFX-encoded request and model response messages specified in the customized script template, sends the IFX-encoded request to the middleware, captures the response, compares it to the model response, and flags the test script execution as having passed or failed. A link to the script template and common sub-routine files used in this project can be found with this article online at http://websphere.sys-con.com/ (*WebSphere Journal* archives Vol: 4, Iss: 8).

The following sections describe the details behind the design of each test script component, including insight on how to avoid some of OpenSTA's quirks and limitations.

Test Script Templates (Sort Of)

The test script templates referenced above are more conceptual than physical, since OpenSTA doesn't support a script template mechanism. The test team defined a "template" by defining a generic set of steps that each test script would need to execute based on the test case criteria specified by the project and then "cut, pasted, and modified" specific test scripts several hundred times. This may sound tedious, but the effort was quite small compared to the benefits of having a common test script pattern and automated test script execution. The specific test script modifications are described in the section "Customizing the Test Script Template" (in part 2 of this article, which will run in *WebSphere Journal* Vol: 4, Iss: 9).

Test Script Example

Figure 3 illustrates a test script for a specific test case that follows this test script template approach. The script is displayed in the OpenSTA Script Modeler editor.

Note that the Script Modeler provides color-coding of Script Control Language keywords, commands, and comments. This feature came in handy while learning the commands and syntax of the Script Control Language. Also note that the test scripts are displayed in the left-hand frame of the OpenSTA user interface and managed using a directory-like structure that groups test cases into test case categories. This arrangement had to be set up outside of OpenSTA by using sub-directories in the OpenSTA directory "Scripts." The OpenSTA user interface didn't provide a way to create this hierarchy, but it could use it once it was deployed.

"With a tight budget, a looming deadline, and no test automation tool, it was clear this was a chance to explore what Open Source had to offer"

Listing 1 at the end of this article shows the contents of the test script shown in the Script Modeler graphic (Figure 3).

Based on the defined template, each test script followed the same execution pattern of OpenSTA commands and common subroutines (described in the section "Test Script Subroutines" in part 2 of this article, which will run in *WebSphere Journal* Vol: 4, Iss: 9). Referring to the line numbers shown in Listing 1, the script execution pattern is as follows:

- **Lines 1 to 13** – initialization commands including the set of common variables provided by the file "IFX_VARIABLES.INC." The OpenSTA script compiler requires that Line 1 specify a target browser and browser version for the test even though this test doesn't involve any sort of browser whatsoever.
- **Lines 14 to 18** – the commands to identify character strings that will be initialized by the IFX-encoded request and response messages specified in the File option. It took some trial and error for the test team to realize that even though the file name is specified in the command syntax, OpenSTA requires that the final extension of the file be ".fvr." Therefore, all of the IFX-encoded messages use the format *test-script-name.xml.fvr* for the message name. Another limitation of OpenSTA is that the File option has to be provided with a hard-coded value. If this could have been a variable value, then a single test script could have been written to completely parameterize test case execution.
- **Lines 19 to 43** – the commands and sub-routine calls to read the IFX-encoded request and response messages issue an HTTP POST to send the message, receive the reply, and report the results to the test results file.
- **Lines 44 to 58** – the commands and sub-routine calls to compare the actual IFX-encoded response to the model IFX-encoded response. If the comparison is successful, the test case is flagged as passed; otherwise, it's flagged as failed.
- **Line 59 to 60** – the command to reference the common subroutines file "IFX_SUBROUTINES.INC" that provide the logic for the Read_Rq, Read_Model_Rs, Post_Request, REMOVE_CRLF, and CHECK_4A sub-routines. Unlike the conventions followed by some other languages, the OpenSTA Script Control Language requires that an Include command that references sub-routines be placed inline and not within the Definitions section of the script.

Part 2 of this article will appear in the September 2005 issue of WebSphere Journal Vol: 4, Iss: 9.

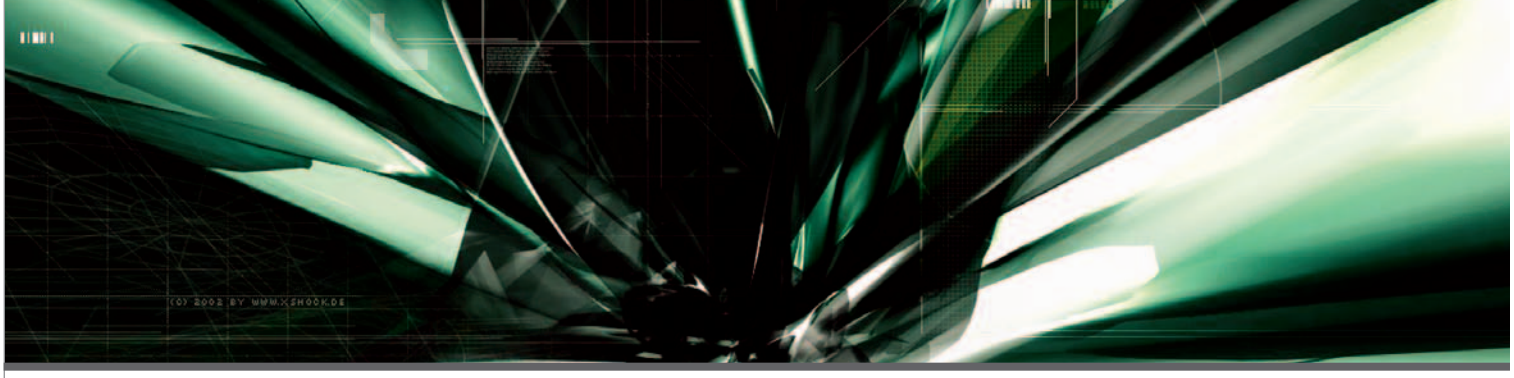
Resources

- http://opensta.org – the homepage for the OpenSTA project
- http://www.ifxforum.org – the IFX forum and source for the latest IFX specification
- http://www-306.ibm.com/software/awdtools/studiobranchtransformation/ – the homepage for the IBM WebSphere Branch Transformation Toolkit

LISTING 1 – CUSTOMIZED TEST SCRIPT

```
1   !Browser:IE5
2   !IFX Message Test

3   Environment
4          Description "C01A-F0300-04A"
5  Mode              HTTP
6  Wait              UNIT MILLISECONDS
7
8   Definitions
9
10 Include                    "RESPONSE_CODES.INC"
11 Include                    "GLOBAL_VARIABLES.INC"
12 Include                    "IFX_VARIABLES.INC"
13
14 !Identify the Request file
15 CHARACTER*5120   REQUEST_LINE   , FILE="C01A-F0300-04Arq.
   xml"
16 !Identify the Model Response file
17 CHARACTER*5120   RESPONSE_LINE , FILE="C01A-F0300-04Ars.
   xml"
18
19   Code
20 !Begin the script
21 Entry[]
22
23 !Start the test case for logging purposes
24 LOAD SCRIPT into SCRIPT_NAME
25 START TEST-CASE SCRIPT_NAME
26 REPORT "START " , SCRIPT_NAME
27
28 !Read in the Request IFX Message
29 CALL Read_Rq
30
31 !Read in the Model Response
32 CALL Read_Model_Rs
33
34   !Post the request message to the IFX server
35   CALL Post_Request
36
37 !Report on the Request and Response
38 REPORT SCRIPT_NAME , " Request: ", THEXML
39 REPORT SCRIPT_NAME , " Response: ", THERESPONSE
40
41 !Remove all CRLF pairs from the response
42 CALL REMOVE_CRLF [THERESPONSE, NOCRLF]
43
44 !Compare Actual with Model Response
45 Set passtest = 0
46 CALL CHECK_4A
47
48   !Record whether test passed or failed
49   If (passtest > 0) THEN
50         PASS TEST-CASE
51   ELSE
52         FAIL TEST-CASE
53   ENDIF
54
55 END TEST-CASE
56
57   Exit
58
59   !Include common subroutines referenced in all scripts
60   Include                              "IFX_SUBROUTINES.INC"
```



Monitoring WebSphere MQ–Connected Applications

Gaining Real-Time, End-to-End Visibility

BY SATEESH NARAHARI

jshissler@wilytech.com

Enterprises need to think about a performance management solution that will give them real-time end-to-end visibility into the entire Web application environment from browser to back-end systems, including messaging middleware such as WebSphere MQ. By providing an application-centric view of MQ-connected environments, such a tool allows application administrators to work collaboratively with MQ administrators without displacing existing MQ monitoring tools. The result is an effective, seamless process for detecting, triaging, and diagnosing application performance issues in complex interconnected environments.

Intended for Web application administrators, this article will discuss the core requirements for monitoring and managing the performance of WebSphere MQ-connected Web application environments and illustrate the advantages in having a single tool that can provide deep visibility into the application and its connecting components.

MQ Adoption Is Increasing

Enterprise IT is under constant pressure to deliver business value and optimize business processes without increasing costs. Tighter IT budgets limit IT directors' spending choices, causing many to look much harder at leveraging existing infrastructure and in-house software such as mainframes and ERP systems.

At the same time, IT directors are increasingly required to meet IT governance demands and successfully implement additional standardization programs such as ITIL initiatives. One result of these trends on the development and deployment of composite application architectures is an increasing reliance on integration middleware that is reliable and provides a proven way of tracking business-critical transactions.

Rather than creating proprietary point-to-point integration solutions, many IT organizations are implementing IBM WebSphere MQ, which provides an event-driven architecture that decouples systems and allows independent upgrades of IT system components. WebSphere MQ also establishes the foundation for higher-level application integration solutions such as publish/subscribe systems and message transformation engines. The increasing reliance on WebSphere MQ to integrate Web applications with various enterprise systems presents IT organizations with new challenges when it comes to ensuring composite application performance and availability.

What Is WebSphere MQ?

WebSphere MQ is IBM's middleware solution for commercial messaging and queuing. Used by thousands of customers around the world in every major industry, it's a key component in distributed applications and is supported on a variety of platforms. Estimated to have a 70% market share, WebSphere MQ is the Message Oriented Middleware market leader and has become the de facto industry standard for messaging middleware.

WebSphere MQ use has been growing rapidly in enterprises that are building and deploying highly integrated composite J2EE applications for a number of reasons:

- As companies face the need to comply with various regulations and corporate governance mandates, they are standardizing their technology infrastructures.

Reach Over 100,000

Enterprise Development Managers & Decision Makers with...

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

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

Get Listed as a Top 20* Solutions Provider

For Advertising Details Call 201 802-3021 Today!

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

SYS-CON MEDIA

The World's Leading i-Technology Publisher

- The integration of Web applications with existing ERP, CRM and SCM packages continues to be an imperative.
- Also, as IT teams take on strategic enterprise-specific projects that require application integration, they are realizing that they can leverage their existing investment in WebSphere MQ.

WebSphere MQ provides messaging backbone for applications by enabling reliable delivery of the key transactional messages between various systems. WebSphere MQ enables applications to communicate reliably across a network of processors, subsystems, and operating systems.

Queue Manager

The heart of WebSphere MQ is the message queue manager (MQM), its runtime program. Its job is to manage queues and messages for applications. It provides the Message Queuing Interface (MQI) for communication with applications.

Figure 1 shows interaction between three different applications (App1, App2, and App3) and illustrates message flow between applications. In the first case, the message msg1 is being sent from App1 to App2, both of which are running on the same WebSphere MQ system. In the second case, msg2 is being sent from App1 to App3, but App3 is running on a different MQ system. Msg2 goes through a Message Channel Agent and the network to reach the second server. WebSphere MQ provides support for communicating between applications whether they are on same server or different physical servers and different operating systems.

Queues

Message queues are used to store messages sent by programs. Queues are defined as objects belonging to a queue manager. WebSphere MQ supports different queue types, each with a specific purpose. These queues are set up for different purposes, and knowing the purpose of each queue would enable application administrators to monitor the queues that are most important to their application. Table 1 shows the various types of queues and an explanation of their purpose.

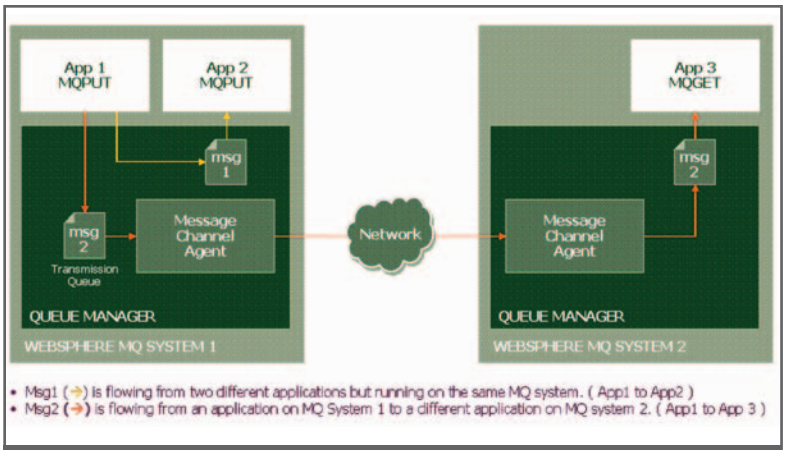


FIG 1: THE WEBSPHERE MQ MESSAGING INFRASTRUCTURE PROVIDES A CRITICAL POINT OF INTEGRATION BETWEEN J2EE APPLICATIONS AND BACK-END SYSTEMS.

Channels

WebSphere MQ provides intercommunication between queue managers (or between queue managers and clients) using channels. In WebSphere MQ, there are two different kinds of channels:

1. Message channels connect two queue managers via message channel agents (MCAs). Such a channel is unidirectional.
2. MQI channels connect a WebSphere MQ client to a queue manager in its server machine. Clients don't have a queue manager of their own. An MQI channel is bidirectional.

A channel can use the following transport types: SNA LU 6.2, TCP/IP, NetBIOS, SPX, and DEC Net.

Java Connectors

The applications running on an application server such as WebSphere Application Server (WAS) also use a Java connector to link to WebSphere MQ. The MQ Java connector has evolved over the years and so there's a history to consider. The original MQ Java connector (MQCCF) is based on the older-type IBM Common Connector Framework. The newer MQ Java connector (MQJCA) is based on the J2EE Java Connector Architecture (JCA) standard.

The JCA V1.5 standard defines an architecture to connect the J2EE platform to heterogeneous Enterprise Information Systems (EIS.) The MQJCA connector represents an implementation of the JCA specifically tailored to operating with WebSphere MQ.

The packages listed below make up the current MQJCA distribution:

```
com.ibm.mq.jar
com.ibm.mq.pcf.jar
com.ibm.mqbind.jar
com.ibm.mqjms.jar
connector.jar
fscontext.jar
jms.jar
jndi.jar
jta.jar
ldap.jar
```

Monitoring Requirements

The increasing adoption of WebSphere MQ-based messaging infrastructures is driving an increase in the need for ensuring that these systems are up and running at peak performance. So enterprise IT shops need to implement a management solution that provides comprehensive visibility into production and pre-production application environments as well as the ability

QUEUE TYPE	PURPOSE
LOCAL QUEUE	A queue is local if it's owned by the queue manager to which the application program is connected.
REMOTE QUEUE	A queue is "remote" if it's owned by a different queue manager. A remote queue definition is the local definition of a remote queue. A remote queue isn't a real queue. It's a structure that contains some of the characteristics of a queue hosted by a different queue manager.
TRANSMISSION QUEUE	Transmission queue are used as intermediate steps when sending messages to queues that are owned by a different queue manager.
INITIATION QUEUE	An initiation queue is a local queue to which the queue manager writes a trigger message when certain conditions are met on another local queue, for example, when a message is put into an empty message queue or in a transmission queue.
DYNAMIC QUEUE	A dynamic queue is defined "on the fly" when the application needs it. Dynamic queues can be retained by the queue manager or automatically deleted when the application program ends. Dynamic queues are local queues.
ALIAS QUEUE	Alias queues aren't real queues; they're definitions used to assign different names to the same physical queue.
DEAD LETTER QUEUE	When a queue manager can't deliver a message, the messages are written to the dead letter queue. Such a queue is defined when the queue manager is created.
REPLY-TO-QUEUE	A request message must contain the name of the queue into which the responding program must put the reply message. This can be considered a "return address." The name of this queue together with the name of the queue manager that owns it is stored in the message header.
MODEL QUEUE	A model queue isn't a real queue. It's a collection of attributes that are used when a dynamic queue is created.
REPOSITORY QUEUE	These are used in conjunction with clustering and hold either a full or a partial repository of queue managers and queue manager objects in a cluster (or group) of queue managers.

TABLE 1

to share critical data with each member of the IT team who has a stake in the performance of the application and its subsystems.

Comprehensive Visibility

While there are tools that can monitor each piece of a composite application, they often present a disjointed view of the application environment. To achieve effective monitoring of Web transactions, it's important to have end-to-end visibility starting from the browser to the back-end systems and everything in between so that IT personnel can proactively detect slowdowns and failures, determine the nature of the problem, and diagnose the root cause of performance problems quickly. This whole application visibility should include the browser, JVM, application server, databases, MQ Java connectors, and WebSphere MQ itself as well as connection to back-end transaction servers and connected third-party systems.

The management solution should monitor 100% of the transactions and let you track component interactions in the individual transaction path. Another key requirement is customizable dashboards that can be tailored to fit the individual needs of users whether they're operations personnel responsible for production monitoring, application administrators responsible for problem triage and diagnosis, or WebSphere MQ administrators responsible for ensuring service levels and system availability.

The monitoring technology should use a non-intrusive monitoring mechanism so it can provide deep visibility into production systems without negatively impacting performance. The metrics reported should reflect the performance of the system in response to real user behavior rather than synthetic transactions that can't give an accurate picture of system performance.

Another key element for an effective solution is a proactive alerting mechanism that can provide incident notification before end users are affected. Whether via pager, on-screen notification, or through a systems management framework console, operations and application support personnel need to know about problems early so that performance issues can be eliminated as soon as possible. The solution should also be able to integrate performance data with framework solutions so that it fits with existing systems management processes.

Monitoring WebSphere MQ

Application administrators responsible for the health and availability of systems that use WebSphere MQ as the messaging backbone are keenly aware of the need to monitor both WebSphere MQ and the Java applications to which they connect. But these administrators tend not to be messaging system specialists, so for them the important question becomes what MQ metrics do I need to monitor? What data can help me pinpoint performance issues to the Java application, to the MQ connectors, or to the MQ system itself so that I can bet-

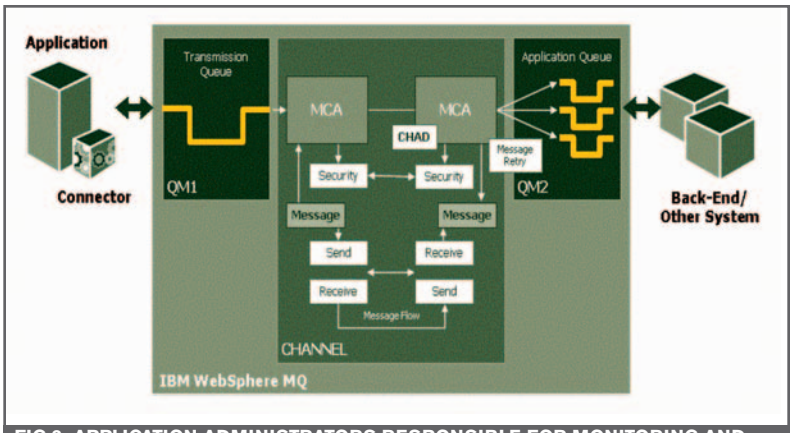


FIG 2: APPLICATION ADMINISTRATORS RESPONSIBLE FOR MONITORING AND MANAGING WEBSPHERE MQ-CONNECTED SYSTEMS NEED LIVE VIEWS OF THE MESSAGING INFRASTRUCTURE AS WELL AS THE APPLICATION ENVIRONMENT

COMPONENT	SIGNIFICANCE
JAVA CONNECTOR	What is the rate of message flow into the application through Java Connector and how responsive has this message flow been?
	What is the rate of message flow from the application to Java Connector and how responsive has this message flow been?
	How responsive is WebSphere MQ when opening and closing a connection?
QUEUE	If PUT ops are inhibited, your applications may not be able to send messages.
	If GET ops are inhibited, your applications may not be able to retrieve messages.
	If the Queue is full, it may indicate a problem with one component on one side or the other.
	Watch the Queue Depth to ensure that it's not too high resulting in a 'Queue Full.'
CHANNEL	If the Channel is in doubt, messages may not be exchanging between Queues.
	If the Channel is stopped and it's unknown to the application, it could result in problems completing your Web transactions.
	Is the Channel running or paused or initializing, etc.?
	How many total messages were handled by the Channel?
	When was the last message on this Channel?
QUEUE MANAGERS	Which Queue managers are currently active?

TABLE 2

ter communicate the exact nature of the problem to the person who must fix it? And further, the tools available to application administrators tend to be best suited to MQ administration, not MQ monitoring. How do I get the metrics that I need to monitor? Many of these administrators have built their own scripts to gather data from WebSphere MQ but the process of correlating these MQ metrics with J2EE application performance metrics is often painstaking and messy.

The administrator should monitor the Java connectors such as MQCCF (based on the Common Connector Framework) and MQJCA (based on the Java Connector Architecture) to ensure that the connection between the application and MQ is running smoothly with no bottlenecks affecting performance. The application administrator should also monitor WebSphere MQ components such as the Queue Manager, Queues, and Channels used by the application being monitored. This is to make sure the application messages are flowing smoothly across WebSphere MQ.

Messaging software such as WebSphere MQ can be shared between multiple applications. This poses an interesting problem for the application administrator. Administrators don't want to monitor every queue and channel in WebSphere MQ; they just want to monitor the queues that their application uses. So it's important for the monitoring software to enable administrators to filter the metrics they want to monitor by queues. The monitoring software should also be efficient in reporting static metrics that don't change. (For example, Queue Manager Name – it's not required to report the same value every monitoring cycle.)

Table 2 illustrates some of the key components of the messaging infrastructure for a given WebSphere MQ instance and their significance from an application monitoring perspective. Monitoring the performance of these components and having an understanding of their affect on overall application performance would enable an application administrator to communicate more effectively with an MQ administrator.

"An integrated management strategy that lets administrators monitor the entire composite application environment with one tool helps to avoid playing the blame game"

Collaborative and Consistent Monitoring

The fact that applications are becoming more complex and interconnected makes it difficult for one person or even a single IT team to manage the entire application environment. It's not uncommon to see various administrators working to ensure the availability of an IT application. For example, one person may be administering the application server, while a different person

may be administering the messaging system. Moreover, QA managers will want to monitor the system during testing and then hand the monitoring task over to operators once the application has been deployed into production. The goals of each stakeholder are to ensure that their systems are up and running and successfully meeting performance and business objectives.

However, monitoring these components in isolation with different tools presents only a fraction of the whole picture and doesn't reveal any potential problems that can happen as a direct result of the way these components interact with each other in different environments (staging versus live production for example). For this reason, it's very important for these administrators to collaborate with each other by using a single performance management solution that can let them speak the same language and share the same data. This allows more constructive interactions, i.e., "The messaging system isn't working because this particular channel is down" as opposed to "The messages don't seem to be getting delivered."

Having a consistent mechanism for monitoring various application components is a first step towards effective communication. When administrators are monitoring applications, they shouldn't have to switch among multiple tools with different terminology or be required to use disparate tools that have been squeezed into same packaging (often the result of a poorly executed acquisition by an IT vendor). An administrator's life is hard and is becoming harder with the increasing complexity of applications. Adding extra tools can only make it worse.

Another key requirement for effective collaboration and information sharing is historical data analysis. An optimal monitoring solution would be able to store performance data for periods of up to a year so that capacity planners and application administrators can retrieve it whenever needed to perform trend analysis for future planning.

Conclusion

To ensure maximum availability, application administrators need to extend performance management visibility beyond the applications into systems such as WebSphere MQ, which are becoming increasingly important to enterprise IT organizations. When devising a monitoring strategy, IT teams should develop an understanding of how the application uses WebSphere MQ and then determine what components of the messaging infrastructure to monitor. An integrated management strategy that lets administrators monitor the entire composite application environment with one tool provides an effective means for detecting and diagnosing production problems and helps IT teams avoid playing the blame game when slowdowns occur.

WAS for z/OS and DB2

Clearing up some of the conceptual confusion

BY LINFENG YU

Linfeng Yu is a software architect with ISO, Inc. He has extensive experiences in developing large-scale, complex enterprise-wide architectures and cross platform software development. He has been working with WebSphere for both distributed platform and z/OS since version 3. He can be reached at lyu@iso.com.

Most WebSphere Application Server (WAS) for z/OS customers use DB2 for z/OS as the backend or data store. DB2 for z/OS is a high-performance DBMS, with a strong reputation for handling high-volume data access transactions. Picking the right WAS for z/OS and DB2 Subsystem connectivity architecture can greatly improve the system's performance, availability, scalability, security, and transactional capability.

On the other hand, application developers are always confused by some of the WAS for z/OS DB2 application development concepts such as local transaction, JDBC connection sharing, locking, and isolation-level control. It's necessary to clarify these concepts.

This article will discuss the WAS for z/OS and DB2 Subsystem connectivity architectures and clear up the most confusing WAS for z/OS DB2 application development concepts. If not specifically stated, I'm speaking here of WAS for z/OS V5 (or later) and DB2 for z/OS V7 (or later).

To make the article self-contain, I'll start with some zSeries terminologies even though most of them have appeared in my previous articles. If you're familiar with them, you can jump directly to the JDBC Providers and Drivers on z/OS section.

z/OS Concepts and Terminology

zSeries hardware allows the allocation of resources to multiple logical partitions (***LPARs***) in a single zSeries machine, with each partition supporting a single z/OS image.

A ***SysPlex*** is a collection of LPARs joined together to form a single logical entity or view to an external observer.

A Coupling Facility (***CF***) is a zSeries machine with microcode that allows high-speed communication between LPARs in a SysPlex as well as a common repository for sharing data by subsystems like DB2 that are in different LAPRs in the SysPlex.

Resource Recovery Service (***RRS***) is a z/OS component that can perform transaction management for multiple subsystems such as CICS, DB2, WebSphere, and WMQ on the same LPAR.

An application program can use the Resource Recovery Services Attachment Facility (***RRSAF***) to connect to and use DB2 to process *SQL* statements, commands, or *IFI* calls.

Workload Manager (***zWLM***) uses installation-defined policies and service-level commitments to govern the performance of a workload in the system.

Dynamic Virtual IP Address (***DVIPA***) is a common external IP address for an application residing or executing on different LPARs in the SysPlex.

Sysplex Distributor (***SD***) is a z/OS component (part of the TCP/IP stack) that consults zWLM to distribute inbound DVIPA requests to the most suitable LPAR in the SysPlex.

Automatic Restart Management (***ARM***) is a z/OS component that will try to restart a job or task after a failure.

Data sharing lets multiple DB2 subsystems have concurrent full read and write access to databases on shared direct access storage devices (DASDs).

Distributed Relational Database Architecture (***DRDA***) is the DB2 database communication protocol.

JDBC Providers and Drivers on z/OS

JDBC is an Application Programming Interface (API) that the Java programming language uses to access different forms of tabular data, as well as some hierarchical systems. Applications developed for WAS for z/OS use JDBC to access data in DB2 Subsystem.

WAS for z/OS provides a Relational Resource Adapter (***RRA***) implementation. The RRA uses a JDBC driver to access data through JDBC calls to the database. The connection management is based on a JCA connection management architecture. It provides connection pooling, transaction, and security support. WAS for z/OS V5.x supports JCA 1.0, whereas WAS for

z/OS V6.0.1 supports both JCA 1.0 and JCA 1.5.
In WAS for z/OS a JDBC provider defines a set of JDBC drivers that relates to a particular database type; a Data Source is defined "under" the JDBC provider definition; it contains specific information about the database to which it connects. Applications are mapped to Data Sources.

Three DB2 JDBC providers exist in WAS for z/OS V5.X:

- DB2 Universal JDBC Driver Provider
- DB2 Universal JDBC Driver Provider (XA)
- DB2 for z/OS Local JDBC Driver Provider (RRS)

In WAS for z/OS V6.0.1 the first two have been combined into one called DB2 Universal JDBC Driver Provider.

Table 1 lists the providers and what they offer. JDBC providers can be "scoped" in WAS for z/OS. Three levels of scopes can be set for a JDBC provider: Cell, Node, and Server. Scoping at the Node level is the common approach, though there are reasons to scope lower still. Currently the two DB2 JDBC providers in WAS for z/OS can't coexist because they have same classes with the same names but different implementations, so their scopes mustn't overlap at all. The only two possibilities are:

- One scoped to "Node A" and other scoped to "Node B"
- One scoped to "Server A" and other scoped to "Server B" in same node

So which provider should you use? In general, you should use the DB2 Universal JDBC Driver Provider unless the DB2 for z/OS installation doesn't support the DB2 Universal Driver.

The DB2 Universal JDBC Driver supports JDBC 3.0 with both Type 2 driver and Type 4 driver implementations. The Type 4 driver implementation that connects to the DB2 Subsystem uses DRDA over TCP/IP, whereas the Type 2 driver implementation connects to the DB2 Subsystem through RRSAF (it requires DLLs that are included with the DB2 for z/OS V8 and APAR PQ80841 distribution for the driver). So the Type 2 driver implementation's performance is equivalent to that of the "Legacy" Local JDBC Driver.

Considering that the two DB2 JDBC providers can't coexist there's no point in using the "Legacy" Local JDBC Driver if the DB2 Universal JDBC Driver is available in the installation. The migration from the "Legacy" Local JDBC Driver Provider to the Universal JDBC Driver Provider is transparent to application developers.

Then which Type of driver implementation of the DB2 Universal JDBC Driver Provider should you use to connect to the DB2 Subsystem? The answer is, "it depends." Let's take a look at the connectivity options.

WAS for z/OS DB2 Connectivity Options

Note: The Type 2 driver and connection mentioned in this section all refer to the Type 2 driver implementation in the DB2 Universal JDBC Driver Provider.
The general recommendation is to use the DB2 Universal JDBC Driver Type 2 implementation for local DB2 connections and Type 4 implementation for remote DB2 connections. "Local" means that the DB2 Subsystem locates in the same LPAR as WAS for z/OS. Whereas the "remote" refers to the DB2 Subsystem located on a different LPAR from the WAS for z/OS.

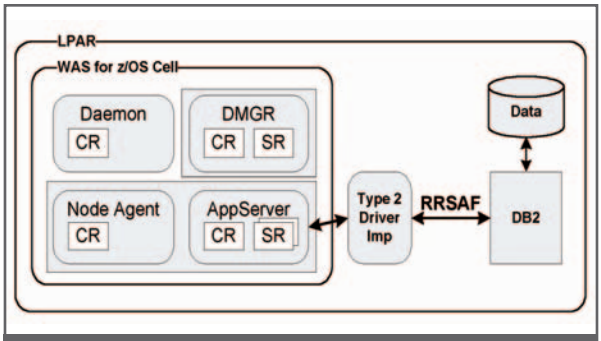


FIG 1: WAS FOR Z/OS DB2 LOCAL CONNECTION

OPTION 1

As shown in Figure 1, a Type2 connection to a local DB2 Subsystem offers the best performance because of the connection through RRSAF (memory to memory communication). The scalability and availability are somehow limited to the DB2Subsystem on a single LPAR. Since the Type 2 connection supports thread-level security, it has some special security advantages. RRS is exploited to handle two-phase commits in this architecture. Theoretically nothing prevents you from using a Type 4 driver for the local DB2 subsystem connection, but the performance will be 50% worse and thread-level security support is no longer available. So it's not recommended.

PROVIDER	DB2 UNIVERSAL JDBC DRIVER PROVIDER			DB2 FOR Z/OS LOCAL JDBC PROVIDER (RRS)
IMPLEMENTATION	XA Data Source	Connection Pool Data Source		Connection Pool Data Source
DRIVER TYPE	Type 4	Type 2	Type 4	Type 2
CONNECTIONS	IP	Local	IP	Local
COMMIT PHASE	1 phase 2 phase	2 phase (RRS)	1 phase	2 Phase (RRS)
THREAD-LEVEL SECURITY	No	Yes	No	Yes

TABLE 1: DB2 JDBC DRIVER PROVIDERS

OPTION 2

A Type 4 connection is the better choice in connecting to a remote DB2 Subsystem. It performs better than a Type 2 driver connecting to a local DB2 Subsystem and subsequently to the remote target DB2 Subsystem. The connectivity architecture is shown in Figure 2. Compared to Option 1, it has worse performance and worse availability (more single points of failure) but better scalability (subsystems on different LPARs can scale independently). Type 4 XA connection can be established to handle two-phase commit. Unfortunately, thread-level security isn't supported in this architecture option.

OPTION 3

Of course, you can combine *Option1* and *Option2* – connect to the local DB2 Subsystem through a Type 2 connection and the remote DB2 Subsystem through a Type 4 connection. It applies when your application has to access both the local and remote DB2 Subsystems at the same time. To do so, you should define different Data Sources in WAS for z/OS.

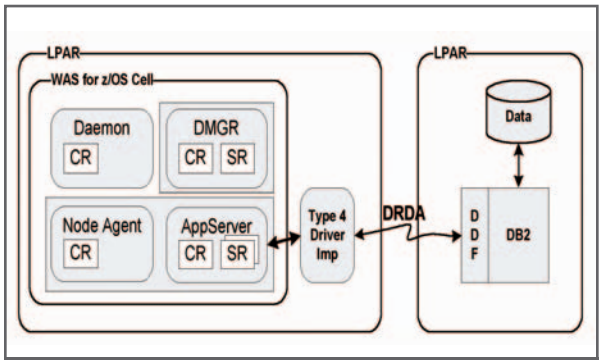


FIG 2: WAS FOR Z/OS DB2 REMOTE CONNECTION

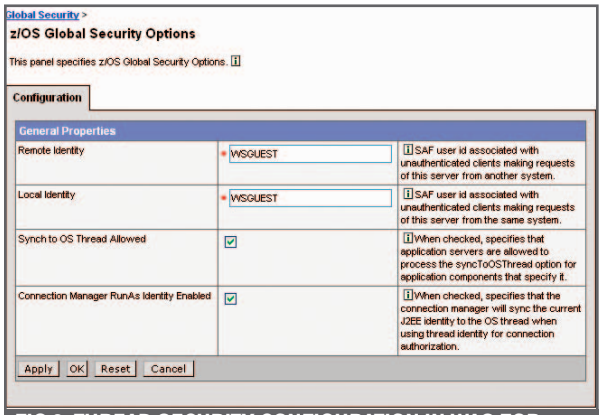


FIG 3: THREAD SECURITY CONFIGURATION IN WAS FOR Z/OS V5.1

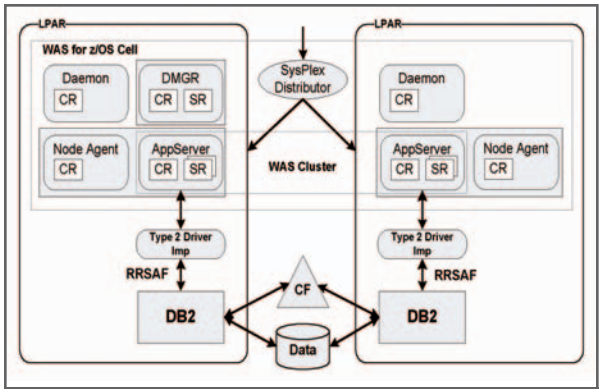


FIG 4: DB2 DATA-SHARING GROUP TYPE 2 CONNECTION

From a performance and security perspective, it's recommended that WAS for z/OS co-locates in the same LPAR with the DB2 Subsystem so that Type 2 connections can be used by the applications. As previously stated, a Type 2 connection greatly out-performs a Type 4 connection. Using thread-level security you can propagate a customer id to DB2 rather than a common proxy id. Applications that need tracing and auditing at the individual customer level can exploit this nice feature in the Type 2 driver implementation.

To use thread-level security you have to understand the thread identity and thread security concepts exclusive to WAS for z/OS available for JCA connectors. Thread identity support is used to flow the current thread identity to the backend EIS (Enterprise Information System). Once the user is authenticated, his id flows to any work he does in the z/OS system, cutting an SMF(System Management Facility) audit trail records as it goes. Thread security can be used to "push down" the current thread identity onto the OS thread. The Type 2 driver implementation captures the id from the OS thread.

Certain conditions should be met to use thread-level security. The WAS for z/OS user repository must be a local OS registry. The JCA connection must be container-managed. And the Connection Manager RunAs Identity Enabe has to be enabled as shown in Figure 3.

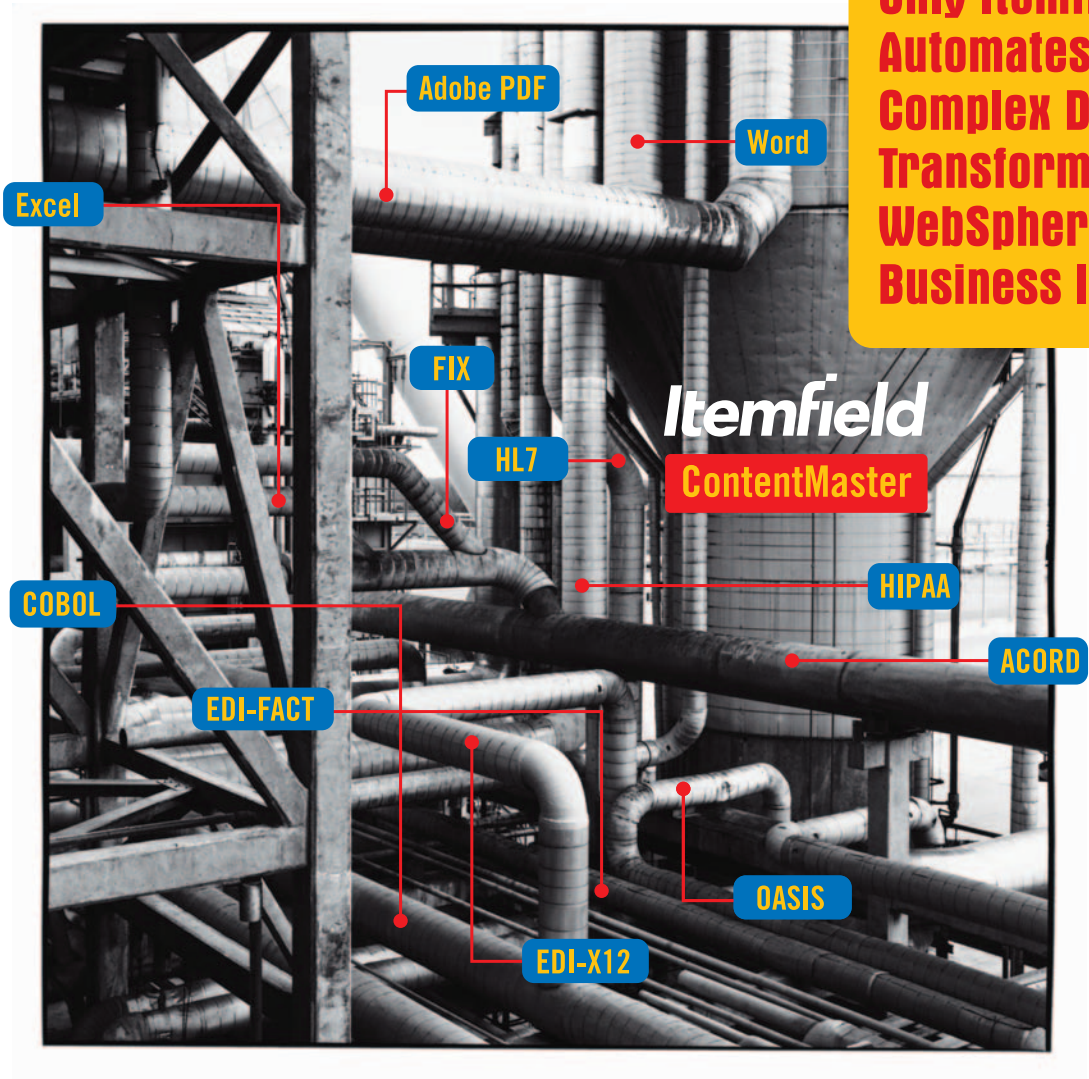
So far we've discussed WAS for z/OS and DB2 for z/OS in the context of a single LPAR. To improve overall scalability and availability the WAS for z/OS Network Deployment (ND) and DB2 data-sharing group in a SysPlex environment has to be exploited.

OPTION 4

Look at Figure 4. It depicts a Type 2 local connection architecture in a SysPlex environment in which each WAS for z/OS connects to the DB2 Subsystem in the same LPAR. Besides the performance and security advantages, the cross-LPAR WAS cluster combined with the DB2 data-sharing group has better scalability and availability than that in a single LPAR.

There's a major problem with the architecture when the WAS server is up running but the DB2 Subsystem isn't available. User requests are distributed to WAS servers by the SD based on each WAS server's workload and status (provided by the zWLM). Because the WAS server is running fine, more and more workloads are sent to the WAS server, creating a storm drain. The problem can be solved to a certain extent by using ARM to restart the DB2 Subsystem. When the DB2 Subsystem is recycled the JDBC Type 2 driver can

Only Itemfield Automates Complex Data Transformation For WebSphere® Business Integration



Extend The Power And Reach of WebSphere Business Integration

Transform Any Data Format to XML and Back Again—Without Writing Code

Together, IBM and Itemfield solve the costly problem of data transformation to reduce the cost of business integration with a combination of products:

From Itemfield

Itemfield ContentMaster™ Agent for WebSphere Business Integration Message Broker simplifies solutions to a broad range of integration needs and data formats with WebSphere Business Integration Message Broker, providing a high throughput interface between any data formats.

From IBM

IBM WebSphere® Business Integration Data Handler for Complex Data leverages Itemfield ContentMaster to enable the bidirectional conversion of a vast number of text and binary formats to and from WebSphere Business Integration business objects and can be used with any server in the WebSphere Business Integration product family.

Visit our website **www.itemfield.com/wbi** to obtain a copy of our latest white paper *"The Complexity Advantage: Solving the Requirements of Complex Data Transformation in Business Integration"* and to view the ContentMaster Solution for WebSphere Business Integration demo.

Itemfield | 1 800 838 1055 | www.itemfield.com/wbi

reconnect to the same DB2 Subsystem and purge the dead connections.

There are two facts about this architecture worth noting: the two-phase commit is handled through RRS and the DB2 connections are attached to a specific member of the DB2 data-sharing group instead of to the group itself.

OPTION 5

If the Type 2 connections in Option 4 are replaced by Type 4 connections and use SD, the connections are directed to the DB2 data-sharing group. So all the single points of failure are removed. The new architecture is depicted in Figure 5. It provides the highest level of availability and DB2 Subsystem location transparency. Taking the Type 4 driver's XA support into consideration, it seems like an attractive solution for DB2 connectivity. Unfortunately the architecture is 't recommended because global transactions can be deadlocked.

When multiple JDBC connections participate in a global transaction, all branches of the transaction share locks. Uncommitted changes made by one branch are visible to every branch. Unfortunately SysPlex doesn't support global transactions; all JDBC connections must process against a single subsystem (a single member of the data-sharing group) to share locks. Since Type 4 connections are directed to the DB2 data-sharing group they can be routed to different DB2 Subsystems by the SD. Whenever that happens, you're exposed to deadlocks. To avoid them, you have to serially reuse the same connection just like using a local connection. But you lose the level of availability added by the SD.

However the performance of a local DB2 Subsystem suffers in a Type 4 connection versus a Type 2 connection. Why use a Type 4 connection if you're forced to use local connection?

Normally Option 4 would be the recommended connectivity architecture in a production environment. But you should still evaluate the application needs and infrastructure constraints before making a decision. You may be wonder how to handle two-phase commit with XA resources if you choose Option 4. Actually WAS for z/OS provides two-phase commit support for transactions that access both XA resources and RRS resources in the same transaction.

WAS for z/OS Local Transaction

Local transaction is the first concept that has to be clarified. To understand local transaction, let's make a detour to global transaction first. A global transaction is a logical unit of work (LUW) that spans multiple resource managers such as DB2 and MQ. The J2EE specification defines a global transaction's behavior. As you can guess, a local transaction refers to a LUW that doesn't span multiple resource managers.

A resource manger local transaction (RMLT) is a resource manager's view of a local transaction; that is, it represents a unit of recovery on a single connection that's managed by the resource manager.

An LTC is a bounded unit-of-work scope in which zero, one, or more resource manager local transactions (RMLTs) can be accessed. The LTC defines the boundary at which all RMLTs must be complete; any incomplete RMLTs are resolved according to policy by the container. An LTC is local to a bean instance. The LTC defines the application server's behavior in an unspecified transaction context. The unspecified transaction context is defined in the EJB 2.0 (or later) specification. An example of an unspecified transaction context is the EJB transaction attribute *Not supported*.

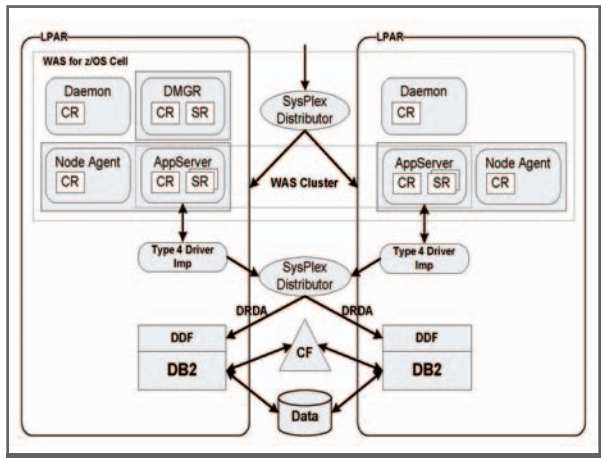


FIG 5: A DB2 DATA-SHARING GROUP TYPE 4 CONNECTION

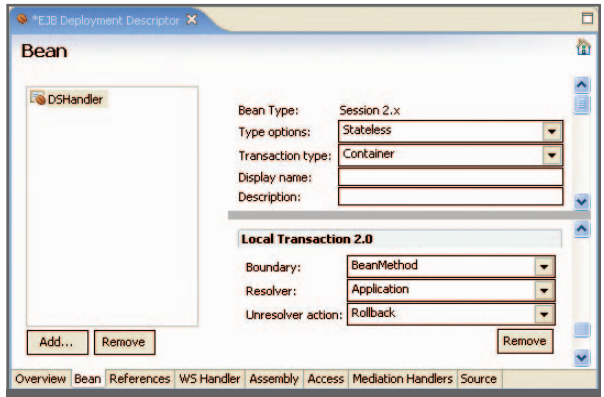


FIG 6: LOCAL TRANSACTION CONFIGURATION IN RAD 6

If you want to access several non-XA resources in a method and manage them independently, you should use LTC. As shown in Figure 6, **Local Transactions – Resolver** should be set to *Application* and **Local Transactions – Unresolver action** should be set to *Rollback* in the component's deployment descriptor. In the Container transaction deployment descriptor, the Transaction should be set to *Not supported*.

However, you should be aware of a restriction if you plan to use LTC with DB2. The RMLTs in an LTC don't commit until the transaction boundary defined in LTC is reached. Be careful about lock duration. And if you manipulate the same table using different database

Imagine seeing EAI's impact on business process performance in real-time...

Now imagine implementing it in minutes, *instead of* months

To get your free copy of the informative new article

"3 Steps to Uncomplicating Business Process Management"

go to www.nastel.com/uncomplicate
or call 800-580-2344 or
+631-761-9100

Business happens in real time.
If something impacts your business processes, you need to know. *Now.*

AutoPilot™ from Nastel monitors and manages all of the complex events that comprise your business-critical transactions, giving you deeply granular visibility from end to end for all your business processes. AutoPilot lets you define customized, actionable alerts in the context of each user's specific business function to streamline your business processes and eliminate costly failures, bottlenecks and redundancies. With AutoPilot, you can see the impact of EAI on your business process performance in real time, on-demand.

Unlike other monitoring software, though, AutoPilot's unique architecture takes the complexity out of implementation. It doesn't require months of programmers and consultant's time before you reap the benefits. With AutoPilot, you simply point and click, and within minutes you're in total control.

NASTEL™ TECHNOLOGIES, INC.

Put Your Application Infrastructure On **AutoPilot**

connections other than serially reusing the same connection, you can be exposed to database deadlocks (under certain database isolation levels).

Sharing Database Connections

Sharing database connections in WAS for z/OS refers to multiple *getConnection* calls in the application returning multiple connection handles with the same physical connection (a physical connection corresponds to a DB2 thread). Sharing connections make an application more scalable. It also reduces resource allocation and the chances of deadlocks.

To share database connections, certain conditions have to be met. In WAS for z/OS you can only share a connection scope in a global transaction. However you can serially reuse connections in an LTC. The application's deployment descriptor element **Sharing scope** has to be set to shareable as shown in Figure 7. The following database connection properties have to be the same:

- Java Naming and Directory Interface (JNDI) name
- Resource Authentication setting
- Principal
- Connection transaction isolation-level property
- Connection readOnly, catalog, and typeMap properties

Listing 1 shows two connection handles sharing the same physical connections in a user transaction.

If the resource reference of the Data Source is marked Unshareable, the Connection handles aConn and bConn will have different physical connections. The application can fail if the Data Source doesn't

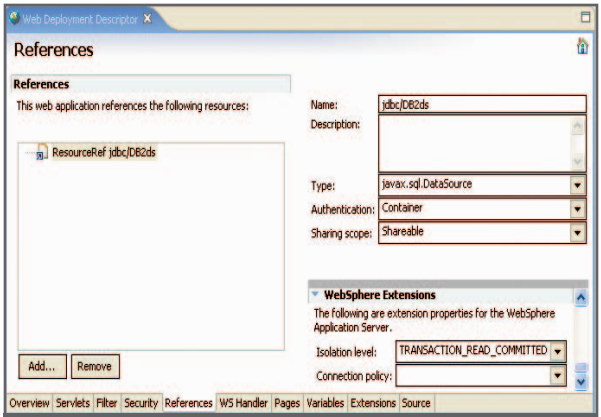


FIG 7: DATA SOURCE ISOLATION-LEVEL SET-UP IN RAD 6

support two-phase commit. Even if the Data Source has two-phase commit support, a deadlock is possible. Using shared connection can avoid this problem because of a strategy that calls for multiple work items to be performed on the same connection.

The use of shareable connections in WAS for z/OS is recommended. However you can still use unshareable connections if your application has to change connection properties. The get/use/close pattern is preferred to share connections. Should you need more information about sharing connections, please refer to the resource.

DB2 UDB FOR Z/OS	JDBC/WEBSPHERE
UNCOMMITED READ (UR)	TRANSACTION_READ_UNCOMMITED
CURSOR STABILITY (CS)	TRANSACTION_READ_COMMITED
READ STABILITY (RS)	TRANSACTION_REPEATABLE_READ
REPEATABLE READ	TRANSACTION_SERIALIZABLE

TABLE 2: DB2 FORZ/OS AND JDBC ISOLATION MAPPING

It's worth pointing out that WAS for z/OS connection manager doesn't reset database connections returned back to the connection pool. The connection returned by the *getConnection* call will carry the properties set by the application component before it's returned back to the connection pool. For example, a connection is returned back to the connection pool with *AutoCommit* value *FALSE*. If the connection is acquired from the connection pool again, the *AutoCommit* value of the connection is still *FALSE* rather than *TRUE*. Initializing the connection before using it can avoid undesired behavior.

Locking and Isolation Level

Transactional locks in DB2 for z/OS are a key component in ensuring database integrity. During database access, transaction isolation determines the nature of the locks to be acquired, which ultimately determines the transactional integrity.

There are four isolation levels in DB2 for z/OS: Repeatable Read (RR), Read Stability (RS), Cursor Stability (CS), and Uncommitted Read (UR). Table 2 shows the DB2 for z/OS and JDBC/SQLJ isolation mapping.

The isolation levels in a Java application are determined by the following precedence order:

1. The SQL statement WITH clause. This overrides all other forms of setting the isolation level, but offers little flexibility for deploying the same application with different isolation characteristics.
2. The last *setTransactionisolation*( ) on the current connection. This method is only available in WAS for z/OS V5.1 (or later). It can't be used for shared connections.
3. The WAS for z/OS resource reference value. Figure 7 shows how to set the isolation level of a Data Source.
4. The default for JDBC in WAS for z/OS. The default is (SQL) Read Stability (JDBC) Repeatable Read. This is different from WAS and DB2 for distributed platforms. It can have negative concurrency side effects especially for a DB2 data-sharing environment. Consider the use of isolation CS and MAXROWS 1 with LOCKSIZE ANY to simulate row-level locking.

To maintain the integrity with isolation Cursor Stability (CS), the application must use a mechanism

beyond the isolation level to make sure that other transactions can't update data between the time the data is selected and when the target data is updated. This can be done by using either an update cursor or overqualified predicates on an updated statement.

In two-phase commit processing, if the other resources in the transaction don't maintain locks, such as WebSphere MQ, the application should exploit DB2 for z/OS as the last resource enlisted in the transaction to reduce the lock duration.

Conclusion

WAS for z/OS and DB2 for z/OS are a perfect couple for developing high-end business applications. This article discussed the connectivity architectures of the two products and clarified several WAS for z/OS DB2 application development concepts. The best solutions are determined on a case-by-case basis sensitive to specific application needs and infrastructure constraints.

LISTING 1: SHARING CONNECTIONS IN A USER TRANSACTION

```
//start a trnasaction
userTransaction.begin();

//get the first connection
aConn = ds.getConnection();
aStmt = aConn.createStatement();
...

//get the second connection
bConn = ds.getConnection();
bStmt = bConn.createStatement();
...

//Commit the transaction
userTransaction.commit();
```

Resources

WebSphere for z/OS information center:

- http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/welcome_nd.html
- http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1//index.jsp?topic=/com.ibm.websphere.zseries.doc/info/welcome_zos.html
- http://publib.boulder.ibm.com/infocenter/wasinfo/v5r0/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/welcome_zos.html

Application Programming and Reference for Java (SC26-9933-04):

- http://publib.boulder.ibm.com/cgi-bin/bookmgr/download/dsnjvh17.pdf
- http://publib.boulder.ibm.com/epubs/pdf/dsnjvj12.pdf

WebSphere Portal Server 5.1

Your Guide to Portal Clustering

There are some things with WebSphere PortalServer that work well and are well documented. There are other things that are well-documented and work well in theory. There are still other things that have okay documentation and will work well when all of the WebSphere stars are aligned. Depending on your implementation, Portal Clustering can fit into all three categories.

BY: CHRIS LOCKHART

Chris Lockhart is a senior technical resource at Perficient, a firm recognized for its expertise around IBM technologies. Chris has worked with IBM's WebSphere, Tivoli, and Lotus Software platforms for more than six years. For more information, please visit www.perficient.com. chris.lockhart@perficient.com

De-Mystification

Given the broad range of configurations open to an infrastructure planner when dealing with WebSphere software, it's no wonder that there's often a good deal of confusion when it comes to making a complex product like Portal highly available. Naturally, there can be no single "Learn Clustering in 20 Minutes" manual that conveys all knowledge. But there are some general steps to follow and some to avoid.

Let's start with a very basic explanation of the WebSphere Cell and how the Portal fits into it (see Figure 1).

A basic install of WebSphere AppServer is often on a single machine, typically referred to as a node. A simple enterprise infrastructure might include several AppServer nodes each performing their own tasks. As the infrastructure evolves and grows more complex, the need arises to simplify operational management of the multiple nodes. The Network Deployment incarnation of WebSphere Application Server lets you centrally manage and control multiple Application Servers. The configuration of the security parameters, user directories, JNDI resources, etc. is greatly simplified by this centralized operational management. Instead of deploying code to three different WebSphere machines, I can now deploy once and have the Network Deployment services distribute the code for me. The combination of the Network Deployment Manager and the Application Servers it manages represent the WebSphere Cell (see Figure 2).

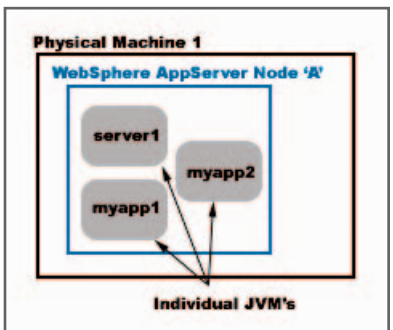


FIG 1: A SIMPLE WAS INSTALL

The process of adding Application Server nodes to the cell is called federation. Its opposite, unfederation, describes removing a node from a cell and returning it to standalone status.

The Deployment Manager communicates with cell members via a small JVM called the nodeagent located on each of the managed nodes. The Deployment Manager will keep track of all configuration changes occurring and utilize the nodeagents on the federated nodes to "synchronize" the configurations so that they all represent what is defined in the master configuration on the Deployment Manager itself.

In many respects Portal is no different than any other Web application that would be configured to run in a WebSphere Cell. Just like a regular Web app, I could choose to run Portal on a single node in the cell or on multiple nodes or even have multiple instances running on a single node (see Figure 3).

Unlike a regular Web application, Portal utilizes several other services that aren't part of the regular WebSphere Application Server product set. As such, there's a completely separate installation that must exist on each node that you want to run Portal.

A Network Deployment infrastructure consisting of several WebSphere AppServer nodes gives you tremendous advantages that could (and do) fill entire Red Books. For purposes of the Portal, there are four that are of primary importance:

- Simplification of operational management
- Central management of Portal configurations

- Highly available environment
- Session and cache data replicated between all Portal nodes

So the assumption at this point is that you've decided you want to cluster Portal for some good reason and there's no real need to go on ad nauseum about the wonderful benefits to be derived from a clustered topology. Right, so let's get to it.

Which Way to Cluster?

Starting with Portal 5.1, there are two ways to establish a cluster of Portals. The first way is to build a WebSphere Cell of multiple federated nodes and then install Portal onto each node in the existing cell. The second way is to take a set of standalone nodes with Portal already installed on them and federate them into a cell. Which way is better? Well, that depends.

Using an existing cell of AppServers and simply installing Portal into that topology is easiest, but has its drawbacks. When an Application Server node is federated into a cell, it loses its default configuration and assumes the configuration dictated by the Network Deployment Manager. If the node is ever unfederated, the original configuration is loaded from a backup copy and the node reverts to its standalone state. So, if I have a node that's already been federated and I install an application on it, if that node ever gets unfederated it will revert to its default state and the application I installed on it will go away. You can see the limitation here. I couldn't unfederate a Portal node and still have it be a Portal node.

Conversely, if I have a standalone node on which I install Portal and *then* I federate it, the Portal is part of the node's original pre-federated configuration and will always be present on the node, even if it gets unfederated. Unfortunately, doing it this way is more complicated and prone to error and endless frustration. Historically this has been the only way to cluster the Portal and it has led to many late nights and high consulting fees.

A real-world recommendation, one that recognizes impossible deadlines, unrealistic demands, and the constant need to document procedures, would be to go the first route. Build your cell of federated WebSphere nodes, verify that they function, *then* add Portal to the mix. It's less flexible, but I think you'll find yourself going home at 5pm more often.

Federation

Naturally you first have to install WebSphere Application Server. Let's assume you can get this far and that you have two WAS nodes (WAS1 and WAS2) installed and ready to go at version 5.1.x. There are no Web applications installed other than the default ones. You have also installed and configured a Network Deployment Manager (DM1) at version 5.1.x. For argument's sake, let's say each component is installed on a separate Windows machine.

In our scenario, WebSphere Global Security isn't enabled (yet). This makes the process of federation very easy. On each of the two WAS nodes you'd execute the following:

```
WAS_HOME\bin\addNode.bat DM1_
HostName DM1_SOAP_Port
```

By default, the SOAP port of the Deployment Manager out-of-the-box is 8879.

Once executed, you'll see the console output describing the process of federation. When it's complete on both nodes, you'll have a cell. See? So easy. Now let's put Portal into the mix.

Installation of Portal

Our plan is to install Portal first on WAS1, do some basic configuration then install what's called a "secondary" Portal on WAS2. Prior to installing Portal anywhere, however, we must ensure that the WAS1 and WAS2 nodes have any and all patches/efixes required by the release of Portal that we're installing. In our example, we're going to install version 5.1 of Portal.

At this point you go ahead and install Portal on WAS1. This is a custom install. You'll be choosing the Application Server that's already installed on this node and federated into the cell. The installer will recognize that this is a federated node and will prompt you to select whether this is a primary or secondary node. Since this is the first install, you must select pri-

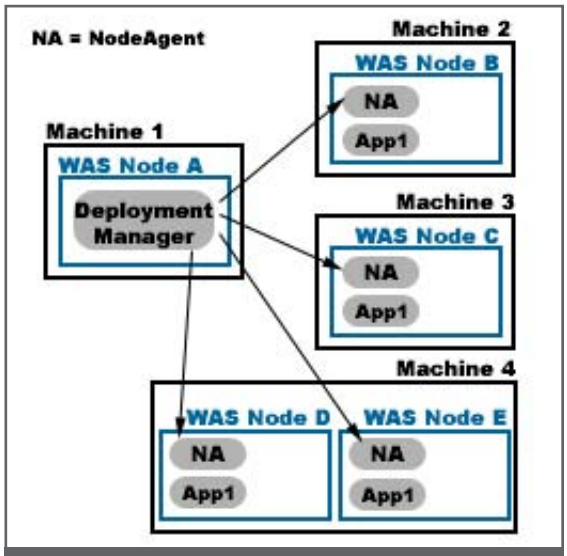


FIG 2: A NETWORK DEP TOPOLOGY

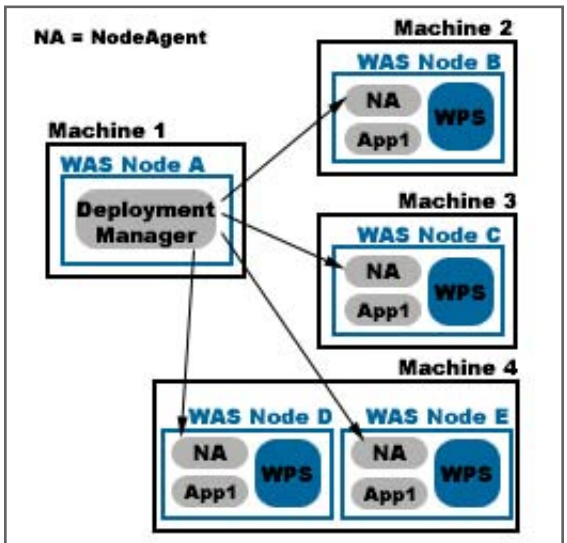


FIG 3: A NETWORK DEP TOPOLOGY INCLUDING PORTAL

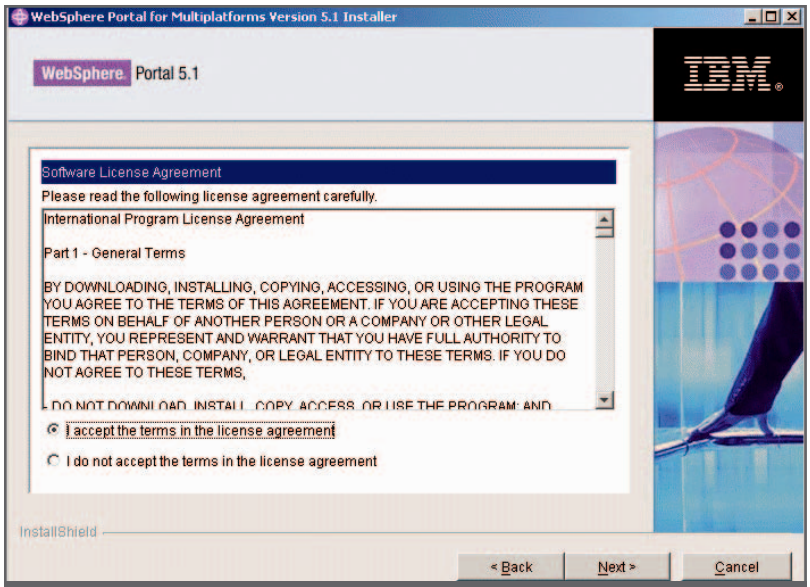


FIG 4: YOU MUST AGREE!

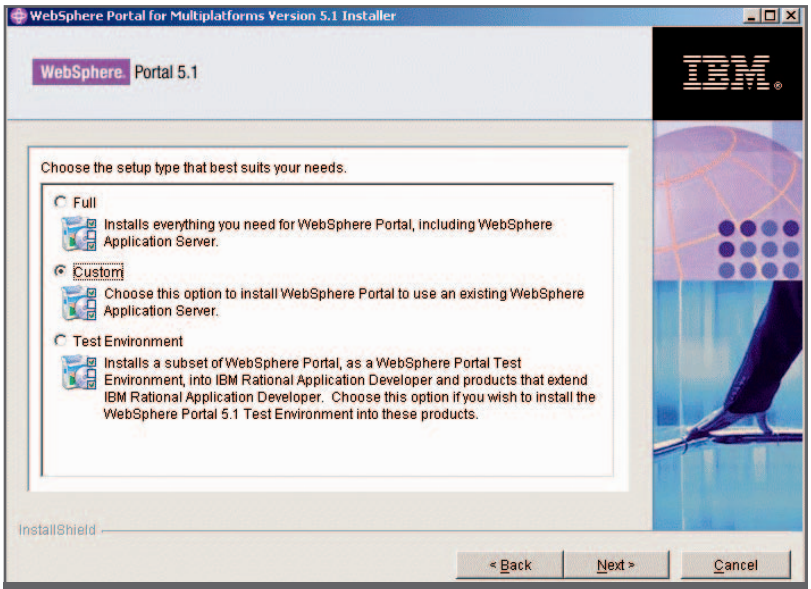


FIG 5: SELECT THE CUSTOM INSTALL SEND

mary node. The installer will interface with the Deployment Manager controlling this node and install the appropriate files.

1. Execute the installer on WAS1
2. Agree to the terms. Click Next. (See Figure 4.)
3. Select Custom, click Next. (See Figure 5.)
4. Select the instance that you want to use from the list. If the installation program doesn't detect an instance of WebSphere Application Server but you know that it's on the machine, you can enter the directory path to the WebSphere Application Server. Click Next. (See Figure 6.)
5. Select Primary and click Next. (See Figure 7.)
6. Specify the directory where you want to install WebSphere Portal. Click Next. (See Figure 8.)
7. Verify the components to be installed and click Next.
8. When the installation is finished, the installation program displays a confirmation panel listing the components that have been installed.
9. Click Finish.

Remote Database

For the Portal on federated node WAS1 to share its configuration information with any additional Portal nodes we choose to install, we must transfer its configuration to an external datastore such as a remote DB2 or Oracle database.

For the sake of this exercise, we'll assume that you've configured a remote DB2 database, as defined in the Portal Infocenter, to accept the Portal configuration data. We'll also assume that you've followed the appropriate steps in the Infocenter to transfer the Portal configuration data to that remote database.

This is a fairly straightforward step and is the same whether you're clustering Portal or not. It involves exporting the Portal configuration data from the bundled Cloudscape database running locally on WAS1:

```
WPS_HOME\config\WPSconfig.bat
database-transfer-export
```

The WPS_HOME\config\wpconfig.properties file is then edited to reflect the necessary database settings for the remote database instance. The data is then loaded to this remote database by invoking:

```
WPS_HOME\config\WPSconfig.bat
database-transfer-import
```

The wpconfig.properties file is read and the data is written out to the remote database server.

Voila, database export/import complete.

Security

The configuration steps for enabling Portal security and WebSphere Global Security are largely the same whether you're talking about a cluster or a standalone environment. However, there are some key differences. For this

example, we'll assume that you've chosen to enable security for Portal and that the user directory will be a remote IBM Directory Server (LDAP). We're also assuming here that LDAP is set up properly with the necessary Portal users.

First, edit the wpconfig.properties file on WAS1 as detailed in the Infocenter. Then execute:

```
WPS_HOME\config\WPSconfig.bat
enable-security-ldap
```

This will set up security for the Portal and put the details in the Portal's configuration now stored in the remote database. It will also enable Global Security for the Deployment Manager and both federated nodes WAS1 and WAS2. By restarting the nodeagents on both WAS1 and WAS2, the DM and the Portal AppServer on WAS1 will put our security policy into effect.

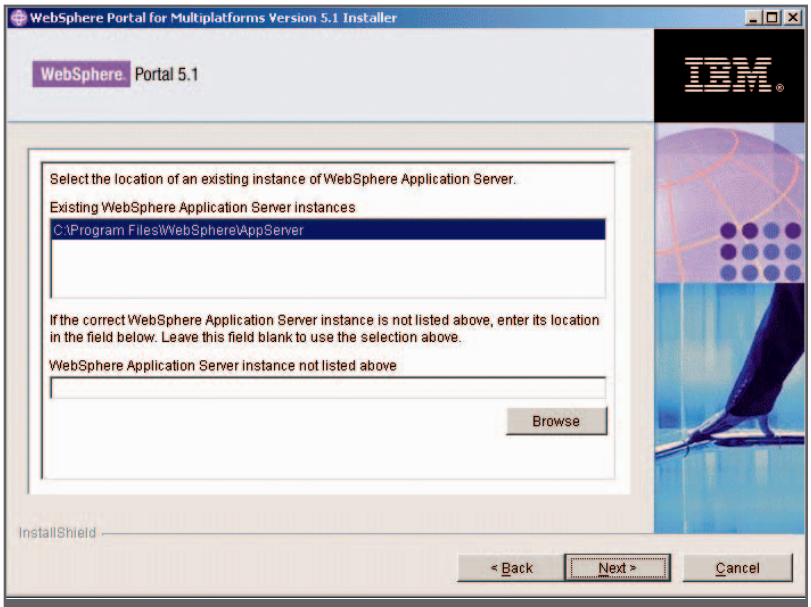


FIG 6: PICK AN INSTANCE, ANY INSTANCE

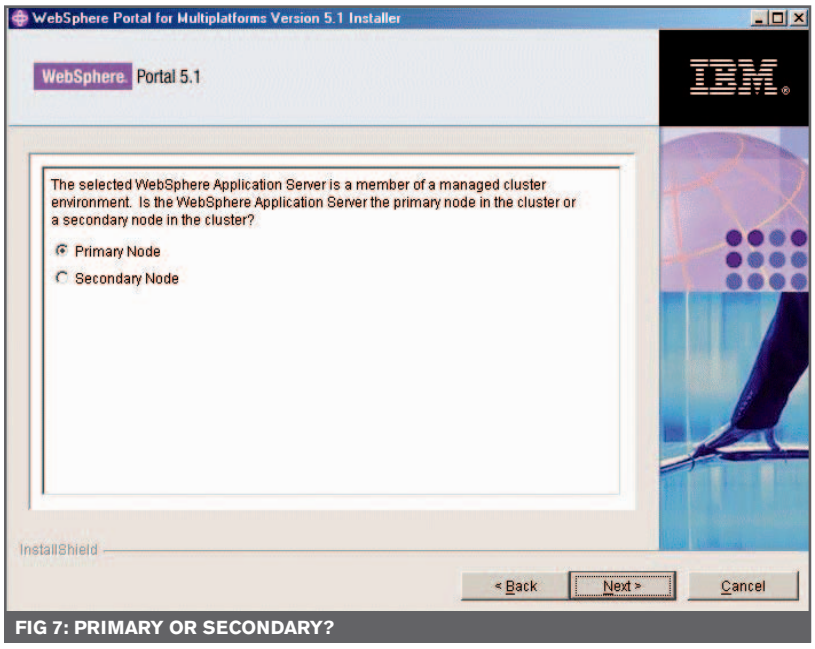


FIG 7: PRIMARY OR SECONDARY?

Portlet Install and Activation

During the installation process, the default portlets will be installed via the Deployment Manager just as if they were any other WAR file. However, portlets are special forms of Web applications that require an additional step called activation. This step will inform the portal configuration that the portlet is available for general use. Since this is not a normal step for installing a Web application, the Deployment Manager can't activate the portlets during the install process. We must do that manually via the command line.

This must be done only on the primary node (WAS1):

```
WPS_HOME\config\
WPSConfig.bat portlets –
DPortalAdminPwd=password
```

Of course, because security is enabled, we must provide the Portal admin password when invoking these commands.

```
WPS_HOME\config\WPSConfig.
bat activate-portlets –
DPortalAdminPwd=password
```

You'll see the called ANT tasks activating each of the default portlets (including the all-important admin portlets) and indicating whether or not they're ready to use. The "active" status is a state defined in the Portal repository currently stored locally on WAS1.

Without this step, you won't see any portlets on the Portal home page and won't be able to administer the Portal.

Ensuring that the Portal continues to function after this step is crucial. You can do this simply by logging into the Portal running on WAS1. If you can log in without errors and browse the Portal, then you're good to go to the next section.

Technically you could stop at this point, skip the next section, and go straight to creating the cluster itself. This would end up giving you a vertical cluster of multiple Portals installed on WAS1. This would be a perfectly fine Portal cluster, but in the event that the WAS1 server imploded you'd lose all of your Portals in one shot. Horizontal clustering is really where it's at when it comes to high availability. So read on!

The Second Node

We should now be able to coast the rest of the way toward our cluster. By this point, the heavy lifting has been done and we are ready to add a second Portal on WAS2.

As noted earlier, we'll be installing a secondary node on WAS2. This will install the requisite Portal files but leave out the default portlets since we'll be using the ones already installed on the cell (the ones derived from WAS1).

1. Execute the installer on WAS2.
2. When prompted, select Custom.
3. When prompted, select Secondary node.
4. The installation program will detect that WebSphere Application Server security is enabled and give you the option of continuing the installation without performing WebSphere Portal configuration. Click **Next**.

The installation of the secondary node bypasses the installation of the portlets since it'll be using the ones that were installed on WAS1.

Once installation is complete, we have to tell this Portal node that security is enabled by editing the WPS_HOME\config\wpconfig.properties file on WAS2. We want the security-related parameters to match what we configured on WAS1. Refer to the InfoCenter for the list of relevant parameters for this step (there are many of them).

Once we're certain that we have the correct values for security, we want to execute the following task on WAS2:

```
WPS_HOME\config\WPSconfig.bat
secure-portal-ldap
```

This command will secure the newly installed Portal with the same values used for WAS1 but this time without affecting WebSphere Global Security definitions on the Deployment Manager. Restart the Portal on WAS2 for the security settings to take effect.

We must also tell our new Portal node to talk to the same configuration database that's being used by the Portal on WAS1. To do this, we need to edit the wpconfig.properties file on WAS2. We want the database-related parameters to match what we configured on WAS1. Refer to the InfoCenter for the list of relevant parameters for this step.

Once the database parameters are configured in the wpconfig.properties file, put them into effect by executing:

```
WPS_HOME\config\WPSconfig.bat
connect-database
```

This command will hook us up so that both Portals are reading the same configuration information.

Now we have two Portals installed, both secured using LDAP, both using the same remote data-

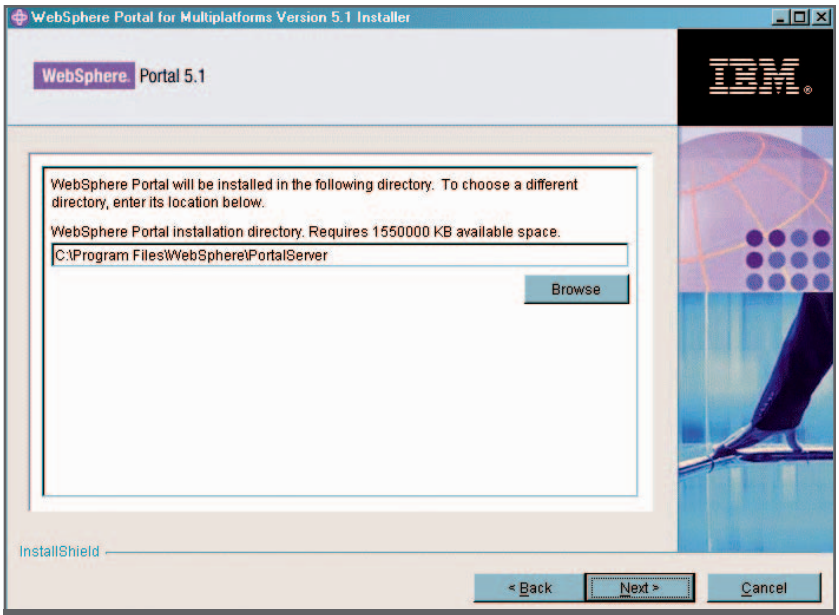


FIG 8: GIDDY-UP! CLICK INSTALL

"Unlike a regular Web application, Portal utilizes other services that aren't part of the regular WebSphere Application Server product set"

"One of the chief benefits that the Network Deployment topology gives us is the ability to have our cluster members on WAS1 and WAS2 share information between them"

base to store config data, and both ready to be clustered.

Cluster Time

The Portal cluster is defined by using the Deployment Manager Admin Console. Log into the console and navigate to the Servers -> Clusters section of the navigation tree.

1. Click New
   a. Define the cluster name
   b. Check the box **Prefer local enabled**
   c. Check the box **Create Replication Domain for this cluster**
   d. **Check the box for "Select an existing server to add to this cluster"** and then choose server **WebSphere_Portal on node WAS1** from the list
   e. Check the box **Create Replication Entry in this Server**
   f. Click **Next**

2. Create the second cluster member
   a. Define the name of cluster member (make sure it's different from the name of cluster member 1 above)
   b. Select the node WAS2
   c. Uncheck the box **Generate Unique HTTP Ports**
   d. Check the box **Create Replication Entry in this Server**
   e. Click **Apply** and then click **Next** to view the summary.
3. To create the new cluster, click **Finish**
4. Save the changes

The cluster is created! See, the actual cluster creation itself is very brief. So easy!

When we create the second cluster member above, the Deployment Manager is actually copying portlets and other configuration files from WAS1 to WAS2. It's synchronizing the nodes so they both contain the same application information. This is why we didn't have to install portlets during the Portal install on WAS2.

You may be asking yourself what the replication entries are that we're creating in this step. One of the chief benefits that the Network Deployment topology gives us is the ability to have our cluster members on WAS1 and WAS2 share information between them. This includes session information as well as dynacache information. With this step, all members in the defined Replication Domain that have Replication Entries will be able to share this valuable information.

If you're load balancing these Portals, having the user session shared between all the cluster members would be part of the high-availability configuration. If one Portal goes down, the other still has your session data active in memory. Excellent!

Tidy Up

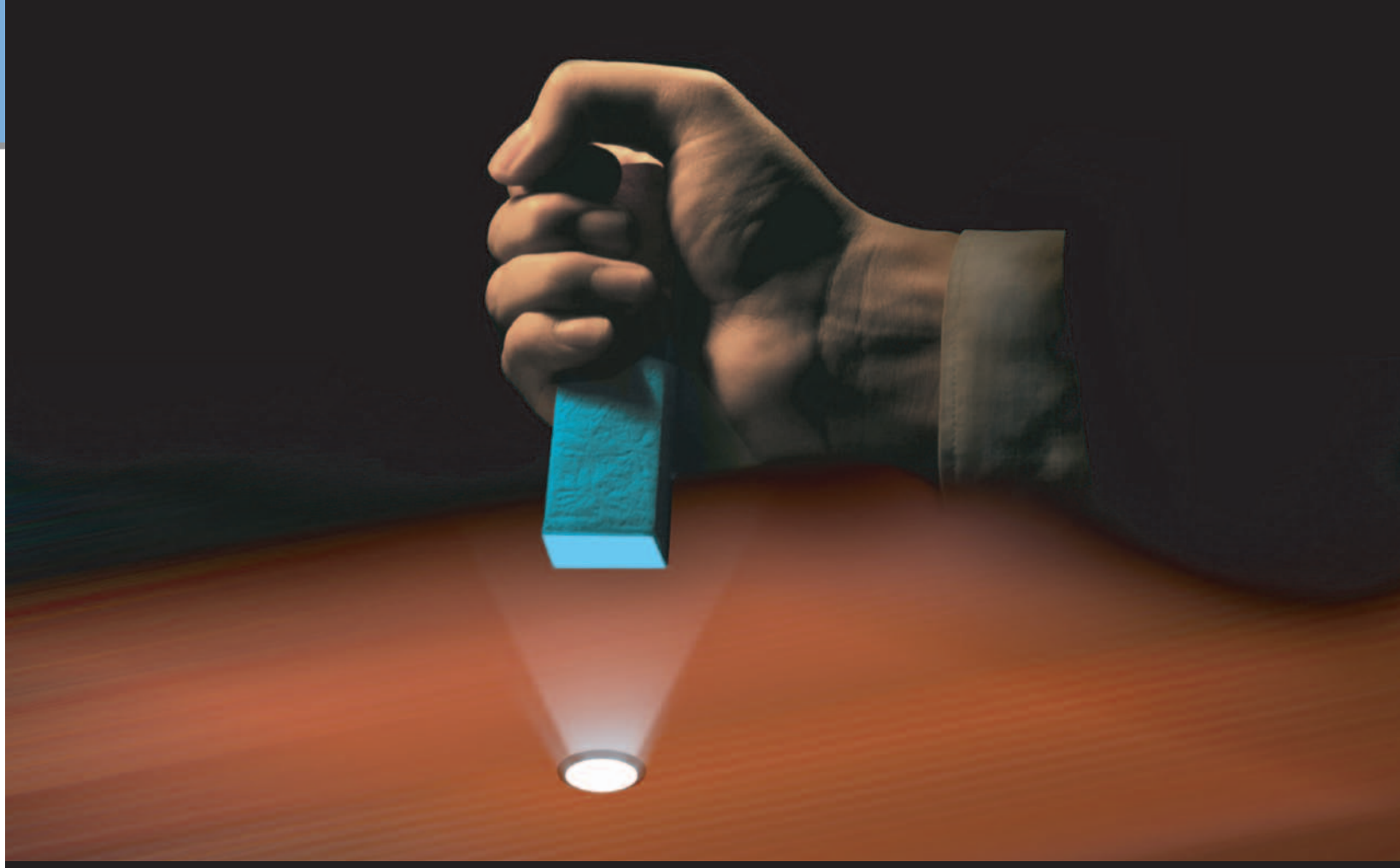
But of course it couldn't be THAT simple. There are a few additional tasks left to perform to ensure the smooth operation of our cluster.

First, to be able to deploy portlets properly, we have to edit a file located on each of our Portal servers. This file is called *DeploymentService.properties* and it's located in WPS_HOME\shared\app\config\services.

Open this file up and set the *wps.appserver.name* property to the name of the cluster you defined in step 1a above. Save and close this file. We're done with it.

Next, we have to update the wpconfig.properties file on each Portal node with the correct cluster member information. The server name is no longer what it was when we installed Portal. Each cluster member has a specific name that we have to refer to it by. So open up the wpconfig.properties file and edit the *ServerName* property. Set this to match the cluster member name used in the Deployment Manager. To determine the cluster member name, click *Servers* > *Cluster Topology* in the Deployment Manager Admin Console and expand the cluster to view the cluster members.

Last, we must enable our Portals to accept the dynamic cache replication that we enabled when we



Think Crystal is a good fit for your Java application? Think again.

Think JReport.

Forcing a Windows reporting solution into a J2EE environment is never going to result in a perfect fit. Only JReport is built from the ground up to leverage J2EE standards and modular components for seamless embedding in any Web application.

JReport is ready out-of-the-box, with all the tools necessary to empower your application with any reporting requirement. From reusable, shared report components to flexible APIs, JReport is the most complete embedded reporting solution available.

With the ability to scale to multi-CPU and server clusters, JReport is a perfect fit for any reporting workload. Load balancing and failover protection provide peak performance and uninterrupted access to critical business data. In addition, JReport integrates with any external security scheme for single sign-on.

JReport's ad hoc reporting lets users access and analyze data on demand, from any browser. And, with cascading parameters, embedded web controls for dynamic sorting and filtering, drill-down and pivot, JReport ensures users get the information they want, when they want it, and how they want it.

See for yourself why over half of the world's largest organizations have turned to JReport to enable their J2EE applications with actionable reporting.

When it comes to embedded Java reporting, JReport is the perfect fit. Download a FREE copy of JReport today at www.jinfonet.com/wj4.

JReport
JINFONET SOFTWARE

created the cluster. In a normal Web application you wouldn't have to do this, but this isn't a normal Web application after all.

If this step is not completed, situations could occur in which users have different views or different access rights, depending which cluster member handles the user's request.

On WAS1, execute the following:

```
WPS_HOME\config\WPSconfig.bat
action-set-dynacache
–DserverName=cluster_member –DRep
licatorName=replicator_name
```

In this syntax, the value of cluster_member is the name of the cluster member to update, in this case the cluster member on WAS1. The value of replicator_name is the name of the cluster member with which to replicate, in this case the cluster member on WAS2.

Be sure to run the same command (but with the values reversed) on the WAS2 node.

After making the configuration changes detailed above, it would be a good idea to do a Full Synchronization of all the changes. This will instruct the Deployment Manager to copy any and all changes out to the two federated nodes via the nodeagents.

In the admin console, select *System Administration* > *Nodes*, select the two nodes from the list, and click *Full Resynchronize*. The admin console will display a message indicating that a *request* for full synchronization has occurred successfully. Be sure to check to the status messages in the Runtime Messages panel at the bottom of the screen to be sure that the request *completed* successfully.

We haven't really talked about a remote HTTP server to handle static content and the incumbent installation and configuration of the WebSphere Plugin, but that's really incidental to clustering Portal. Suffice it to say that the plug-in located on the remote Web servers will contain knowledge of the two Portal cluster members and will route traffic to them based on the policies set in the cluster administration area of the Admin Console.

HTTP Session Replication

When dealing with a cluster, once a user creates a session with a portal by logging in, the user is returned to that WebSphere Portal cluster member for the rest of his session. There's a portlet in the global settings section of the Portal Administration panel that will tell you which cluster member is handling the current session.

The cluster member currently handling the session is referred to as the session owner. If this cluster member were to fail, then the plug-in on the Web servers will route the next request to another cluster member. The new cluster member either retrieves the session from a server that has the backup copy of the session or it retrieves the session from its own backup copy table. The server now becomes the owner of the session and affinity is now maintained to this server.

Whenever a session is modified in any of the cluster members, that session data is replicated to each of the other members of that cluster. So in our example, WAS1 would replicate the session with WAS2. By default, in a cluster the sessions are replicated to each of the cluster members that are using same replicator domain (defined when the cluster was created).

However, because this gets defined during the creation of the cluster and not the cluster member (AppServer) itself, we must go back and tell the Web container on each cluster member's AppServer that it should use this replication domain to store session data.

To enable this "memory-to-memory" session replication:

a. Click **Application Servers > *WebSphere_Portal* > Web Container > Session Management**
b. Click **Distributed Environment Settings** under Additional Properties
c. Select **Memory to Memory Replication**
d. Click **Apply**
e. Repeat these steps for each cluster member

Save the changes to the Deployment Manager master config and synchronize the changes. Restart the cluster members for replication to take effect. You're all set! No fear now of losing a session.

Clustered!

Now that you have a functioning cluster of Portal servers, you may be tempted to believe that nothing will ever go wrong and that your life as a portal admin is worry-free. You have every reason to believe this. Your Portal cluster *is* highly available, it *is* running on the industry's leading application server, it *is* extremely powerful and flexible, but there are some inherent frailties to be aware of. And we'll discuss them next month!

Hopefully you'll come back to hear about some practical techniques for managing real-world portal clusters. This is stuff that you probably won't find in the InfoCenter.

Speaking of the InfoCenter, there's good information regarding the establishment of a Portal cluster currently located at http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/wpf/clus_install.html.

Visit the *New* www.SYS-CON.com Website Today!

The World's Leading *i*-Technology News and Information Source

24/7

FREE NEWSLETTERS
Stay ahead of the i-Technology curve with
E-mail updates on what's happening in your industry

SYS-CON.TV
Watch video of breaking news, interviews with industry leaders, and how-to tutorials

BLOG-N-PLAY!
Read web logs from the movers and shakers or create your own blog to be read by millions

WEBCAST
Streaming video on today's i-Technology news, events, and webinars

EDUCATION
The world's leading online i-Technology university

RESEARCH
i-Technology data "and" analysis for business decision-makers

MAGAZINES
View the current issue and past archives of your favorite i-Technology journal

INTERNATIONAL SITES
Get all the news and information happening in other countries worldwide

JUMP TO THE LEADING i-TECHNOLOGY WEBSITES:

IT Solutions Guide
Information Storage+Security Journal
JDJ
Web Services Journal
.NET Developer's Journal
LinuxWorld Magazine
Linux Business News
Eclipse Developer's Journal
MX Developer's Journal
ColdFusion Developer's Journal
XML Journal
Wireless Business & Technology
Symbian Developer's Journal
WebSphere Journal
WLDJ
PowerBuilder Developer's Journal

The Synergy of Tools

How the IBM Rational Team Unifying Platform Works with IBM WebSphere Tools

BY: REID GERSON

Reid Gerson is a Market Manager with IBM Rational focusing in the areas of business analysis and requirements management, design and construction, and is responsible for marketing the Rational software development solution for Linux.
reid.gerson@sys-con.com

There is a great synergy between the Rational Team Unifying Platform and the IBM development tools, such as IBM Rational Application Developer, IBM Rational Web Developer, and IBM WebSphere Studio Application Developer Integration Edition. While WebSphere developers build, test, and deploy Web, Web services, and Java applications, their entire team needs to ensure the code they write is in line with the project requirements, that versions of their code are maintained, that they are timely informed when requests for change occur, that they are aware of the overall project status, and that the QA team is building test cases validating the code against the requirements set forth by the project. The Rational Team Unifying Platform unifies software development teams by providing guidance in how to write good software and on automating these best practices with tools.

Viewing Team Unifying Platform Tools in Eclipse-based Rational Practitioner Tools

The Team Unifying Platform tools have started to migrate into the Eclipse framework. IBM WebSphere developers can view requirements from their environment. Using Eclipse-based Rational Software Architect or Rational Software Modeler (no direct integration in Eclipse), developers can have up to date access to the latest state of requirements stored in Rational RequisitePro (see Figure 1).

Similarly, developers can have access to defects, change requests and their personal to-do list with the Rational ClearQuest Eclipse plugin. Rational ClearCase and Rational ClearQuest can plug-into either the Eclipse 3.0 SDK or Rational Web Developer, Rational Application Developer, Rational Software Modeler, or Rational Software Architect (see Figure 2).

Using the ClearCase Eclipse plug-in, developers can view the various versions of their code (see Figure 3).

The Rational Unified Process (RUP) is available as a process advisor in Rational Software Architect and Rational Application Developer – it brings contextual process guidance to developers' fingertips. The three ways to view RUP content are through the Process Advisor, Process Browser, and Search. You do not have to start a separate RUP browser (see Figure 4).

Summary

The Rational Team Unifying Platform is extremely valuable to any software development organizations, especially WebSphere software teams, because it unifies the team through a common set of artifacts, tools and processes – providing teams the necessary infrastructure to collaborate successfully…all within their IDE.

The IBM Rational Team Unifying Platform for WebSphere equips WebSphere software teams with:

- Access to the latest requirements, defects, test plans and results
- Visibility into software project data dependencies
- Accurate project status and reports
- Effective change management
- Collaborative software development process

What Is the IBM Rational Team Unifying Platform?

An integrated suite of software development tools:

• Requirements management	IBM Rational RequisitePro
• Defect and change tracking	IBM Rational ClearQuest
• Software configuration management	IBM Rational ClearCase LT
• Test planning	IBM Rational TestManager
• Project documentation automation	IBM SoDA

A process

• Software development best practices	IBM Rational Unified Process

And a project measurement tool

• Project measurement	IBM Rational ProjectConsole

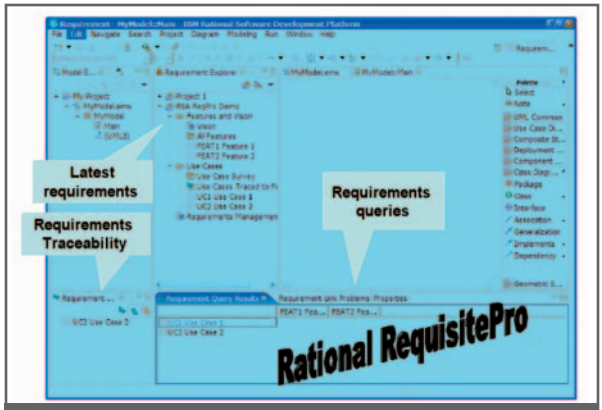


FIG 1: VIEWING REQUIREMENTS IN RATIONAL SOFTWARE ARCHITECT AND RATIONAL SOFTWARE MODELER

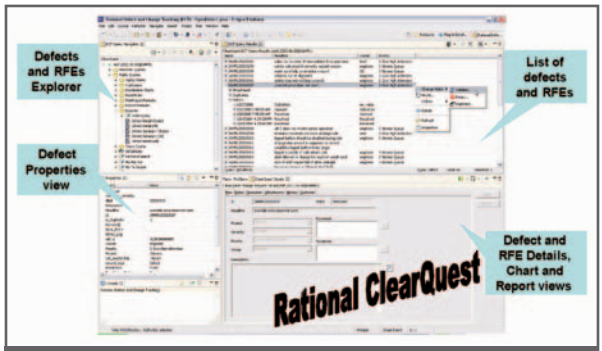


FIG 2: VIEWING CHANGE REQUESTS IN ECLIPSE-BASED TOOLS

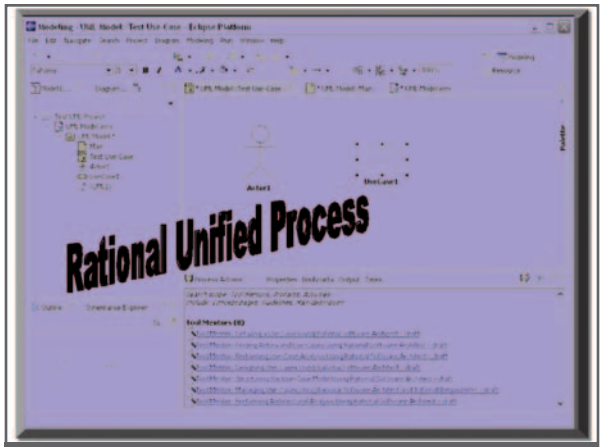


FIG 3: INTEGRATED CONTEXT-SENSITIVE PROCESS GUIDANCE IN RATIONAL SOFTWARE ARCHITECT AND RATIONAL APPLICATION DEVELOPER

Introducing IBM Workplace Designer

Developing applications for the IBM Workplace platform

BY DICK McCARRICK

This article will introduce you to Workplace Designer 2.5. We begin with a brief overview of Workplace Designer and how you can use it to create Workplace applications. We then take a closer look at some of the many features included in Workplace Designer. We assume that you have experience with Web application development. Some familiarity with IBM Workplace products (such as IBM Workplace Collaboration Services 2.5 and IBM Workplace Services Express 2.5) would also be helpful.

Dick McCarrick is a content developer for developerWorks: Lotus. He was previously a member of the Domino/Notes Documentation team for over 11 years, playing a variety of roles in documenting many major components of Domino and Notes. He also wrote the occasional article for *Iris Today* (including Ask Professor INI) before joining the Notes.net/Lotus Developer Domain team permanently in 2002. In his spare time, Dick's leisure activities include running, fishing, woodworking, and reading about the natural sciences. An avid astronomer, he's the former director of the Bridgewater (Mass.) State College Observatory. Dick lives in Vermont.
david.mccarrick@sys-con.com

What is Workplace Designer?

If your background is in Notes/Domino programming, you can think of Workplace Designer as a tool for providing Domino Designer-type application development functionality to the IBM Workplace family of products. In fact, those familiar with Domino Designer will notice a lot of similarities with many IBM Workplace Designer features and concepts described later in this article. This similarity is not coincidental – Workplace Designer was created with Domino Designer in mind. This lets experienced Domino developers leverage their existing skills to quickly create new applications for IBM Workplace products. Figure 1 shows a screen shot of Workplace Designer.

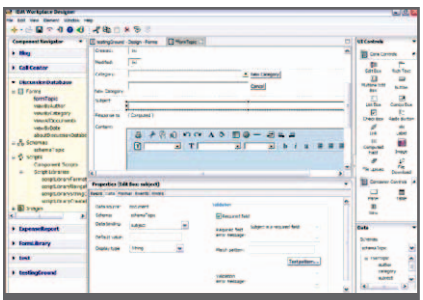


FIG 1: WORKPLACE DESIGNER

Bear in mind, however, that Workplace Designer isn't intended to replace Domino Designer. Domino Designer will still be available as the primary tool for developing applications in a Notes/Domino environment. Workplace Designer lets Notes/Domino developers extend their expertise to IBM Workplace products. (For an overview of new features in the upcoming release of Domino Designer 7, see the developerWorks Lotus article, "New Features in Lotus Notes and Domino Designer 7.0."

Even if you don't have a great deal of experience with development platforms such as Domino Designer, Workplace Designer will give you an easy way to use document-oriented programming for collaborative components without an in-depth knowledge of Java. Workplace Designer's underlying J2EE technologies aren't exposed, so expertise in this area isn't necessary to develop production applications – although if you need the power and flexibility of Java, there are extensive APIs available that let you access the data and services provided by IBM Workplace servers. Developers who have to customize and access the code directly can use tools such as Rational Application Developer. Workplace Designer offers a number of extension points that more experienced Java developers can use to share code artifacts created with Rational Application Developer and other IBM tools.

Inside Workplace Designer

Internally, the components created with Workplace Designer contain artifacts that support the MVC (Model, View, Controller) architecture used in standard Web architectures. Additionally, any documents created from the deployed components are stored in a relational data-

base as XML documents. These documents are defined by XSD schemas and can be accessed through the XPath language. Workplace Designer includes a schema editor for modifying the schemas.

Note that Workplace Designer applications aren't self-contained like Notes databases. When you design an application in Notes/Domino, you can immediately deploy and run it. In the Workplace model, you must deploy components to the Workplace server where they will be run, and then do additional setup to include the component in one or more Workplace composite applications.

Workplace Designer, based on Eclipse 3.0, is completely integrated with IBM Workplace Collaboration Services 2.5 and IBM Workplace Services Express 2.5. (See the developerWorks Lotus article, "Introducing IBM Workplace Services Express.") It installs on top of the IBM Workplace Managed Client. Workplace Designer includes a developer's license for IBM Workplace Collaboration Services 2.5 and WebSphere Portal, and can seamlessly deploy components to the server for testing and assembly into templates and applications. You can authenticate with the Workplace server, and launch the tool from the IBM Workplace client. You can use the components in conjunction with Workplace application builder.

Workplace Designer incorporates the following industry standards:

- XML for data access and exchange
- HTML/CSS for portal development
- JavaScript for the programming language

On the workstation, Workplace Designer runs in IBM Workplace on Linux and Microsoft Windows 2000 or XP. On the server, Workplace Designer applications run on Workplace Services Express 2.5 and Workplace Collaboration Services 2.5 on all supported platforms.

Workplace Designer supports several database platforms, including IBM Cloudscape, DB2, Oracle, and Microsoft SQL Server. In addition, Workplace Designer supports the LDAP servers supported by IBM Workplace products.

The first release of Workplace Designer includes everything you need to get started:

- The product code
- Documentation and tutorials
- Example application code
- IBM Workplace Collaboration Services for previewing and testing in-progress applications
- IBM Cloudscape (technically part of Workplace Collaboration Services)

Workplace Designer Features

Workplace Designer 2.5 includes the following features:

- Components (collections of forms, schemas, scripts, and images) that comprise a single portlet on the Workplace server.
- Design elements. These are parts of Workplace Designer that let you create and modify components visually.
- Form editor for creating forms for entering information into and displaying information from various data sources. Forms are critical elements of a collaborative system, providing a flexible way to collect and process information. They are the main user interface element and contain the controls that comprise the interface of the application. Forms are used to create, edit, and display data stored in documents.
- UI control palette for presenting all available UI controls/components. UI controls are visual elements that let users work with data on a form. UI controls include edit fields, buttons, drop-down lists, and links.
- Events for providing full programming capabilities, such as a button click or field event, document open or close event, and so on.
- Scripting with full JavaScript interfaces to built-in functions. This includes a script editor, offering many of the features found in the Eclipse script editor, including syntax coloring and type-ahead.
- Easy deployment options for testing and production scenarios.

"You don't need to know Java to use document-oriented programming for collaborative components"

These are discussed in more detail in the sections below.

Components

As mentioned, Workplace Designer components are sets of forms, scripts, and other elements from which you construct portlets. (Workplace applications consist of one or several portlets.) You can use Workplace Designer to build components, deploy them on the Workplace server as portlets then add them to Workplace applications. These applications are then available to Web users. You can create new components "from scratch" or import an existing Domino database as a starting point for the new component. You can write scripts to associate with events in a component. To do this, select a component event then write a script for that event. (Scripting and events are discussed below.)

Workplace Designer components support the role-based security model of Workplace applications. After you deploy the component and add the component to an

application or application template, you can set access to components by mapping component roles to the roles in that specific Workplace application. For instance, if you map the template role Contributor to the component role Reader, users with Contributor access to the application will have Reader access to the component.

Workplace Designer provides a methodology to easily deploy components to the Workplace server to be used in Workplace applications. This involves creating a deployment profile in Workplace Designer. For more information, consult the Workplace Designer online help.

Forms

Similar to Notes/Domino forms, a Workplace Designer form is the primary user interface for interacting with IBM Workplace components. Forms in the component let users input and display information. A component can have numerous forms. To allow the component user to navigate from form to form in the current component, the component developer can leverage the form properties or use UI controls.

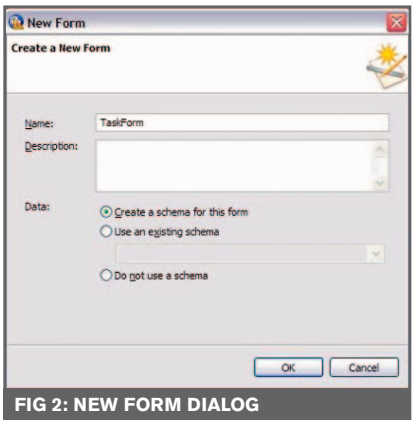


FIG 2: NEW FORM DIALOG

The form is the central design element in Workplace Designer. In Notes/Domino application development, there are many different standalone design elements, including views, agents, and so on. Workplace Designer has one basic element, the form. To create a view, you must embed it in a form; to run the Workplace equivalent of a Notes/Domino agent, you have to make it an event on the form, and so on. In addition, Notes/Domino defines data as part of the form design. Workplace Designer defines data in separate schemas then the developer binds the UI controls to the schema elements. If you don't have an existing schema during form design, Workplace Designer can build the schema for you as you specify the data binding for the UI controls. Note that you need to create a component before you can create a form.

Workplace Designer offers several different ways for you to create a new form. You can start with a blank form, import a form from a Domino database, or copy an existing form. When you create a new blank form, you are prompted for the name of the form, a comment, and whether or not to:

- Create a schema for this form
- Use an existing schema
- Not use a schema

Form Editor

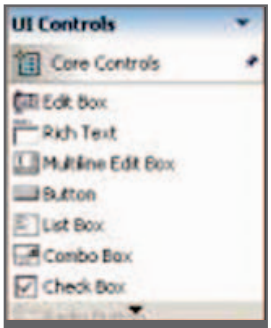


FIG 3: UI CONTROL PALETTE

To create, view, and edit forms, Workplace Designer includes the Form editor (visible in Figure 1). You can use this graphical editor to arrange artifacts such as UI controls, buttons, graphics, and so on. These elements are added using the UI control palette (described below). You can then drag-and-drop these components into the form. Workplace Designer lets you do data validation at several levels. Field-level validation is related to field type and allowed values. Document-level validation happens when multiple fields are involved in the validation process. Depending on the type of validation, it can occur on the client prior to submitting the data, or on the server prior to updating the backend document.

You can select UI controls from the palette and insert them into the form you are designing.

View Controls

As mentioned previously, one area in which Workplace Designer differs from Domino Designer is how it handles views. Views are UI controls that can be added to a form to display a collection of documents. In Workplace Designer, views are controlled through forms. To create a view, you have to insert a view control into a form and then associate one or more schemas and individual fields to the view. View controls provide a way to separate the presentation of the view from the data that will eventually populate it. After inserting a view control into a form, you can modify the format of the control, similar to any other user interface control, using styles or custom formatting properties.

Container Controls

A container control (as the name implies) is a UI control that can contain other UI controls. There are three kinds of container controls:

- **Panel** is a rectangular area on a form to which you can add core controls. Panels can also be used to refer to other documents and to repeat over a selected XML node.
- **Table** contains individual cells. You can add controls into the table cells.
- **View** is a rectangular section on the form that contains rows and columns of data extracted from XML, based on a defined query.

UI Controls

An important concept in Workplace Designer development is the UI control. These are graphical elements (for instance, a button or link) that users can interact with. Workplace Designer offers a large set of UI controls that you can add to your forms. These include edit box,

multi-line edit box, button, list box, combo box, check box, radio button, link, computed field, and image. These UI controls are accessible through the UI control palette (Figure 3), located in the upper-right section of the Form editor interface.

Events

A vital part of any development environment is event handling. Workplace Designer supports several types of event controls, including:

- **UI control events**. Each UI control can have a set of client-side events. For instance, a button can produce an "on click" event. The code can be executed at either server side or client side.
- **Component events**. These are standard IBM Workplace Collaboration Services events; for example, when a component is instantiated or when a member is added or removed to the Workplace application. These events are defined and handled at the component level, using a set of event global properties. (Note: An iFix is required to be applied to the server for this to work.)
- **Form/Document events**. Form events support events on edit controls, such as open, close, onSave, postSave, and so on. Document events support the same events as form events.

Scripting

As we mentioned earlier, Workplace Designer includes a complete JavaScript scripting environment for developing applications. This lets you create relatively complex programs in an easily understood language. The scripts can run on either the server or the client.

The scripting environment includes script libraries for storing reusable code. Workplace Designer's JavaScript interpreter comes with a set of predefined, native libraries that expose all the major IBM Workplace Collaboration Services APIs. (The JavaScript interpreter also features a Java bridge that let you directly call any Java API.) You can also create your own script libraries of reusable script code, or copy/paste script libraries between components.

Workplace Designer's scripting environment also includes a script editor (see Figure 4).

This editor is based on the Eclipse script editor, and includes features such as syntax coloring, type-ahead (code completion), and JavaScript class browsing.

After you deploy the component to the server, it can be added to Workplace applications and application templates.

Other Workplace Designer Features

In addition to the features described in the preceding sections, Workplace Designer offers the following.

SAMPLE APPLICATIONS

Workplace Designer comes with four complete sample applications that you can use to familiarize yourself with IBM Workplace application development. IBM will also make other applications available via the Web that you can download for free and adapt to your own requirements.

DOCUMENTATION AND TUTORIALS

Along with the online help documentation describing features and functions, Workplace Designer includes:

- A step–by-step tutorial for building an IBM Workplace component (this is in the help)
- IDE Reference describing the tool itself and its modules
- API reference listing public APIs, usage, and sample code

IBM Workplace Designer: New Yet Familiar

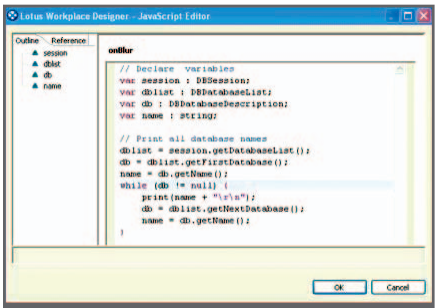


FIG 4: JAVASCRIPT EDITOR

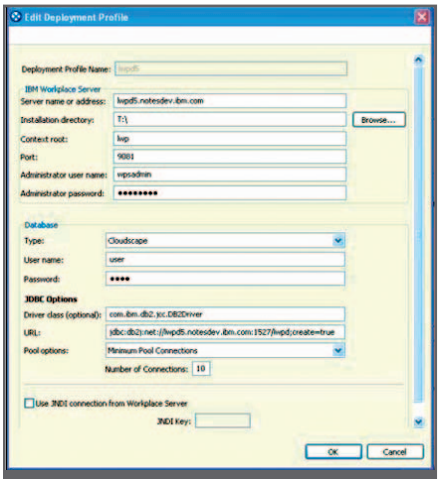


FIG 5: EDIT DEPLOYMENT PROFILE SCREEN

As you can see, Workplace Designer has plenty to offer anyone interested in developing applications for the IBM Workplace platform. Long-time Domino developers can apply their current skills in an environment that functions like Domino Designer and is based on industry-standard technology. And advanced programmers can take advantage of Workplace Designer's extensibility to integrate with Java-based tools, while less experienced developers with no knowledge of Java can create full-featured programs using the script interface. Whatever your background, Workplace Designer will give you an easy-to-use yet powerful tool for creating Workplace applications. To learn more about IBM Workplace Designer, visit the IBM Workplace resource center.

Resources

- To find out more about how IBM Workplace Designer was built, see the developerWorks Lotus interview, "Designing Workplace Designer: An interview with the IBM Workplace Designer design team."
- Learn more about IBM Workplace Designer by visiting the IBM Workplace application development page.
- For information about all IBM Workplace products, visit the IBM Workplace resource center.
- Get involved in the developerWorks community by participating in developerWorks blogs.

WebSphere Integration Reference Architecture

Part One of Two Parts

BY SCOTT SIMMONS

The emergence of service-oriented frameworks results from the evolution of software development and implementation over the last 20 years. Our industry has evolved from monolithic applications and hard-to-manage client/server solutions and has now discovered that the incremental development of components, enabled via a Service Oriented Architecture (SOA), increases the quality of applications, speeds development of new solutions, and addresses the requirements of business stakeholders better.

Scott Simmons is the worldwide technical lead architect for B2B integration for Worldwide WebSphere Business Integration and is an IBM Certified Senior IT Architect. Scott joined IBM in March 2002 from Peregrine/Extricity, where he was the director of Solution Technologies for Peregrine's Office of the CTO. In this role, Scott specialized in B2B solutions for the high technical manufacturing sector. Scott has over 20 years of experience in the IT industry.
scottim@us.ibm.com

The concepts embodied in SOA enable an enterprise to integrate its business processes across its lines of business and their supporting information systems. The ability to coordinate the challenges of enterprise-level integration requires an architecture to facilitate the modeling and managing of services spanning information, applications, and people. Through a component-based approach, organizations can build more flexible integration solutions that leverage a common set of core infrastructure services. This solution framework enables the increased agility and adaptability of IT solutions through a more simplified loosely coupled approach. An SOA-based integration foundation provides support for technology co-existence by supporting standards-based design, development, and implementation. This approach provides a scalable infrastructure across existing and future technology assets to provide a solid foundation for the enterprise integration architecture.

Although many integration products claim to support this architectural approach, many products fall short in their ability to provide enterprise integration requirements. This article focuses on the breadth of capabilities required to support the myriad integration styles required for an enterprise-level approach to integration, and discusses the need for a comprehensive Service Oriented Architecture built on open development and runtime standards as the critical component for integration. Through this architectural foundation, enterprises can extend their solutions by combining existing components with new components and by implementing composite applications as integration requirements emerge.

Enterprise Integration Strategy

An enterprise integration strategy requires, at a minimum, the exchange of messages as the foundation of the solution architecture to enable the integration of people, applications/processes, and information. Through this approach, integration solutions can be implemented at departmental levels and easily extended across enterprise boundaries. This approach to integration provides flexibility and a time-to-market advantage for an enterprise and becomes a critical differentiator for success in today's complex and competitive business environment.

Currently a shift is occurring in enterprise integration strategies. Although tactical initiatives continue to be deployed to solve departmental requirements, organizations are defining enterprise integration architecture initiatives based on an SOA approach. This approach requires organizations to identify core integration assets/components and advocates an approach to encourage the reuse and refinement of these assets for integration projects. In this manner, integration across departments can be approached as an enterprise-level initiative that promotes reuse through a standards-based approach to definition, discovery, and invocation.

An enterprise approach to integration is compatible with a more tactical approach to developing departmental/line-of-business integration solutions. In fact,

the tactical solutions become assets for the overall integration repository. It's important to provide a governance model that ensures that the development of tactical solutions is done within the context and rules of the enterprise's integration architecture. Without this rigor, enterprises will continue to build departmental integration solutions in isolation and fail to achieve any discernible long-term benefits from integration. Given the rate of change in business and technology, the lack of governance in integration approaches will hinder the ability to effectively and proactively support integration objectives across the enterprise.

Service-Oriented Integration Architectures

Coupled with the need to provide a disciplined approach to integration, it's essential that the enterprise integration architecture leverage the strengths of a service-oriented architecture (SOA). SOA enables the creation of modular composite applications by packaging new as well as existing functions as reusable service components using open and available standards. Service Oriented Architectures are said to provide a "set of business-aligned services that are combined (composed and choreographed) to fulfill business goals... services are manifested as a set of interfaces without any dependencies on the implementation mechanism or location." Loosely coupled component architectures are powerful because they let components act as service consumers and/or providers by exposing interfaces in a standard format across a distributed topology. Composite applications improve the flexibility of IT systems through function isolation, an increased ability to reuse components in multiple contexts, a simpler model for integration, and flexibility in constructing integration processes.

The application of a Service Oriented Approach for integration provides many benefits. It:

- Leverages open standards to surface integration assets as services, fostering the reuse of existing assets and helping avoid vendor lock-in.
- Provides a standard way to represent and interact with integration components (such as maps, processes, discrete transactions/services, or interfaces), providing the flexibility to reuse assets across multiple business process solutions.
- Shifts the integration focus to application assembly rather than implementation details, enabling the efficient implementation of new and modified business solutions.

An SOA provides for the connectivity of applications and organizational resources by representing these assets as services that can be composed into higher-order service components. From a technical standpoint, a service-oriented framework for integration enables resources to exchange information (messages, documents, business objects) through a standard interchange framework. By expressing new and existing applications as services, services can be built to address integration requirements.

The SOA approach enables an architectural style consisting of service providers, requestors, and service descriptions enumerated through the familiar publish/find/bind service framework. SOA approaches enable and encourage design principles and patterns encompassing encapsulation, service composition, loose coupling, and reuse.

The service-oriented framework has to be pervasive across all of the aspects of the integration solution. Some current middleware approaches support defining or invoking integration components via SOAP wrappers, but integration architects have to recognize that an SOA approach is far more than just SOAP-based integration. The solution has to support an ongoing evolution of interface definitions (both user-defined and industry-based) as they emerge from standards bodies such as OASIS and OAG. The underlying integration assets of the architecture must conform to an SOA-based approach:

- Service description and definition via a standard interface; for example, WSDL.
- Service invocation/interaction over a standards-based transport/mediation layer.
- Service choreography for orchestrating service interactions; for example, BPEL4WS.
- Service discovery through integrated registry/directory services; for example, UDDI.

"A shift is occurring in enterprise integration strategies"

Composite applications enable the creation of coarse-grained services with flow logic to determine the execution sequence of the services. These composite applications become the IT analogy of business process in the business domain. The SOA design leverages middleware and operating environment functions to separate the flow logic from the underlying business logic implemented in the individual services. Services can be enabled from existing components (for example, mainframe CICS application) or be completely new functions.

A key challenge in a service-oriented integration approach is in the definition and implementation of services at the appropriate level of abstraction. The ability to apply sound architectural design in developing an enterprise approach to integration is tied directly to the concept of separation of concerns as well as the emerging concepts of model-driven design and architecture.

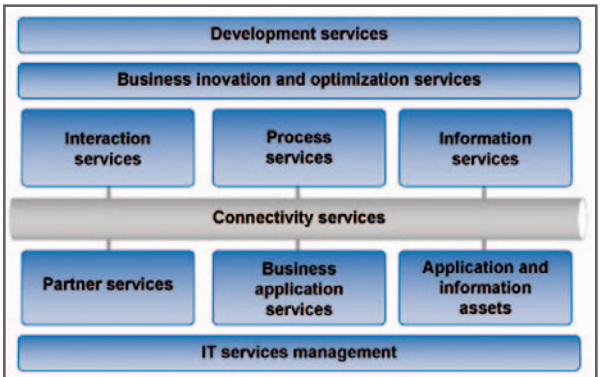


FIG 2: WEBSPHERE INTEGRATION REFERENCE ARCHITECTURE

Separation of Concerns Simplifies Integration Architecture Implementation

The development and implementation of information technology comprises multiple dimensions that have to be reconciled in an enterprise-level integration strategy. IT's role in defining and managing the boundary between business applications and the underlying operating environment becomes the key factor in implementing business solutions. The operating environment provides infrastructure services for business applications and represents common services to support the integration solutions.

When defining an enterprise's integration architecture, it's critical to consider the breadth of integration requirements. These integration requirements can include traditional workflow processing based on human task interaction, the choreography of activities between different systems, distributed data management involving structured and non-structured information, and user interaction capabilities. An enterprise requires the ability to differentiate and design service artifacts appropriate to specific styles of integration required to solve a particular integration problem. As a result, the concept of separation of concerns is the foundation of the definition of the integration services.

Separation of concerns suggests decomposing business needs into services and composing combinations of existing and new services that represent business processes. By separating concerns, integration requirements are decomposed into more granular services. This decomposition is accomplished through the identification of the specific functions that have to be defined to support business requirements, as shown in Figure 1.

A **separation-of-concerns** approach to defining services results in a design that isolates and characterizes functions and services that are implementation-independent. With this approach, building integration solutions becomes an iterative process and components are refined over successive integration projects. Enterprises give design governance and integration mentorship to development teams to promote best practices and the reuse of services and components.

In an SOA-based solution framework, service components are isolated and defined. Besides supporting a top-down model-driven approach, the solution approach enables a bottom-up approach to the development of new services. The resultant composite application consists of service invocations with flow logic to manage the execution order of the individual services. The composite application can be made available to other services as a service, itself the development of additional composite applications.

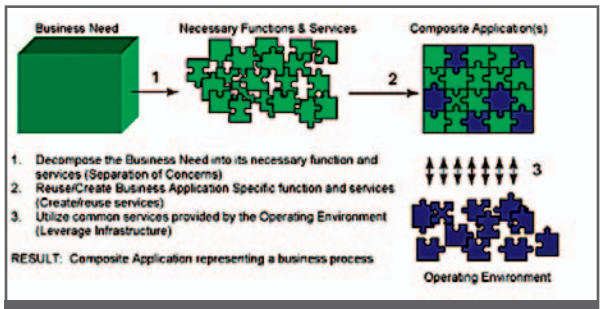


FIG 1: SEPARATION OF CONCERNS

The prevailing approach for building component-based architectures is via model-driven architecture (MDA), and the MDA approach provides a foundation for developing an SOA. In *An MDA Manifesto*, Grady Booch comments that an MDA is "a style of enterprise application development and integration based on using automated tools to build system-independent models and transform them into efficient implementations." The MDA approach provides greater efficiency in the IT development process through the tool-based generation of artifacts, as well as enabling business stakeholders to participate in developing business processes. MDA is rapidly converging with both SOMA (service-oriented modeling and architecture) and SODA (service-oriented development of applications) methodologies for Service Oriented Application development.

In *Guide to Enterprise IT Architecture*, it says, "IT persists in reinventing technical functions because the existing functions were not built by the current faction. At some times this approach may be perfectly justifiable; however in the many circumstances it is merely a whim of the project team. Application development is all about business functions, not building technical infrastructure." This comment underscores the need for an enterprise-governed approach to developing integration in tandem with the application of a separation-of-concerns perspective to building enterprise-level integration strategies.

The WebSphere Integration Reference Architecture

Business integration benefits from an enforcement of separation of concerns to integrate data and applications as appropriate for business needs. It's not enough to have a process solution or an application server foundation – it's the aggregation of multiple integration styles using common components and services that provides long-term sustainability, preserves investment

in existing IT assets, and avoids a "cobbled" integration approach. Effective business integration also requires an operating environment that supports the implementation of service-oriented solutions, can be implemented in a modular build-as-you-go fashion, and supports all functions required for enterprise-level implementations.

The WebSphere Integration Reference Architecture provides a comprehensive set of services to enable business integration. The services provide the breadth of functionality needed to solve integration requirements. More importantly, the component services can be implemented in stages to enable incremental evolution on a project-by-project basis while working towards an enterprise integration solution architecture. Although specific projects may not require all of these services, enterprise-level integration will require the ability to add these functional capabilities to the integration architecture. The resulting architecture provides for separation of concerns by enabling business logic, control logic, routing, and transformation logic to be loosely coupled and, as a result, more flexible to change. At the organizational level, this approach facilitates simpler integration solution development and enhances maintainability and operation of the solution.

The WebSphere Integration Reference Architecture (Figure 2) shows the key integration functions required for comprehensive enterprise-level solutions. These service groupings provide the ability to apply separation of concerns to enterprise integration requirements and lead to a convergence with the principles of SOA as they apply to integration.

This high-level architecture depicts the integration functions/services needed to enable a comprehensive approach to integration. Since these services are described by their interfaces and not by their implementation, a given solution can be made up of mainframe applications, local, or remote services, choreographed processes described by BPEL (the standard for business process description) or as components built with J2EE. The implementations of these integration components provide support for non-functional requirements including reliability, security, availability, and management at both the operating environment level, and the component/service level as well.

Connectivity Services

At the core of the WebSphere Integration Reference Architecture are the connectivity services. This component provides the infrastructure to support and instantiate the Enterprise Service Bus (ESB) architectural pattern. The ESB delivers inter-connectivity services across the distributed component topology. Transport services, event services, and mediation services are provided through the ESB. Transport services provide the fundamental connection layer; event services enable the system to respond to specific events arising as part of a business process; mediation services enable the loose coupling of interacting services in the system. The ESB

essentially becomes the extended enterprise's arterial system providing messaging, notification, and invocation services across the enterprise's various operating environments. The ESB is a key factor in leveraging the service orientation of the WebSphere Integration Reference Architecture for implementing service-oriented solutions today as well as in the future.

The ESB provides for multiple connection technology alternatives to support multiple messaging topologies and patterns, along with implementations such as SOAP/HTTP, RMI/IIOP, and others. In most cases, loosely coupled connections are required to facilitate managing the interactions between services. However, some transactions, by their nature and critical business functionality, can require tightly coupled connections. Both patterns are supported in the WebSphere ESB architecture. The need to support heterogeneous technologies is reflected in support for multiple protocols in a message flow instance, middleware interoperability, specification of different qualities of service (such as persistence, reliability, transactional management, availability), support for various message distribution, and routing (to include publish-subscribe, queue-based, and broadcast-based). Additionally, the connectivity services and ESB architecture provide support for specialized message/information delivery, such as points-of-sale, SCADA, and other pervasive device solutions. Customers using current solutions with WebSphere MQ can implement the ESB pattern today and be able to support new protocols as they emerge.

Within the ESB, there are three major service dimensions that are provided as part of the connectivity framework:

- **Event services** provide event-driven services to enable the components (as well as the organizational resources) to respond to stimuli; business events, for example.
- **Transport services** provide communication services across wired and wireless networks for synchronous and asynchronous message delivery with varying levels of delivery assurance to provide for both location and protocol-independence.
- **Mediation services** enable dynamic in-flight message transformation, dynamic routing, and service-binding resolution services.

An SOA approach to integration isn't possible without the fundamental capabilities of connectivity, routing, and transformation. These functions are enabled via the ESB in the WebSphere Integration Reference Architecture and form the base of the solution architecture. The ESB, in many cases, is implemented using traditional and new emerging middleware solutions to provide access to integration components.

Business Logic Services

Business logic services provide the capabilities required for executing the business logic in the form of service endpoints. The implementation of service endpoints is a critical part of comprehensive integration architecture. Services may be provided through any combination of existing applications; through newly implemented components and through external connections to third-party systems:

- **Partner services** enable the integration of service endpoints provided by trading partners over different transports and using document formats. This layer of services provides support for traditional B2B partner integration solutions with outside partners and suppliers. It's the component of the architecture in which the concerns of cross-enterprise interactions are isolated. These services provide the document, protocol, and partner management services required for efficient implementation of business-to-business processes and interactions:
  - **Community services** enable the management of the trading community for the trading hub, as well as enabling partner self-management.
  - **Document services** enable support for business protocols such as RosettaNet, EDI, and AS1/AS2, as well as associated state management to support conversational processes.
  - **Protocol services** provide transport-level services, including authentication, document routing, and

"The ability to coordinate the challenges of enterprise-level integration requires an architecture to facilitate the modeling and managing of services spanning information, applications, and people"

A LIMITED TIME SAVINGS OFFER FROM *SYS-CON MEDIA*

SUBSCRIBE TODAY TO MULTIPLE MAGAZINES

AND SAVE UP TO $340 AND RECEIVE UP TO 3 FREE CDs!*

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

CALL TODAY! 888-303-5282

Pick a 3-Pack, a 6-Pack or a 9-Pack						
☐ 3-Pack	☐ 1YR	☐ 2YR	☐ U.S.	☐ Can/Mex	☐ Intl.
☐ 6-Pack	☐ 1YR	☐ 2YR	☐ U.S.	☐ Can/Mex	☐ Intl.
☐ 9-Pack	☐ 1YR	☐ 2YR	☐ U.S.	☐ Can/Mex	☐ Intl.

TO ORDER •Choose the Multi-Pack you want to order by checking next to it below. •Check the number of years you want to order. •Indicate your location by checking either U.S., Canada/Mexico or International. •Then choose which magazines you want to include with your Multi-Pack order.

☐ **LinuxWorld Magazine**

U.S. - Two Years (24) Cover: $143	You Pay: $79.99 /	Save: $63 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $32
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **JDJ**

U.S. - Two Years (24) Cover: $144	You Pay: $99.99 /	Save: $45 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $69.99 /	Save: $12
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $89.99 /	Save: $40
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **Web Services Journal**

U.S.- Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **.NET Developer's Journal**

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **Information Storage + Security Journal**

U.S. - Two Years (24) Cover: $143	You Pay: $49.99 /	Save: $93 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $39
Can/Mex - Two Years (24) $168	You Pay: $79.99 /	Save: $88 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $49.99 /	Save: $34
Int'l - Two Years (24) $216	You Pay: $89.99 /	Save: $126 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $59.99 /	Save: $48

☐ **Wireless Business & Technology**

U.S. - Two Years (12) Cover: $120	You Pay: $49.00 /	Save: $71 + FREE $198 CD
U.S. - One Year (6) Cover: $60	You Pay: $29.99 /	Save: $30
Can/Mex - Two Years (12) $120	You Pay: $69.99 /	Save: $51 + FREE $198 CD
Can/Mex - One Year (6) $60	You Pay: $49.99 /	Save: $10
Int'l - Two Years (12) $120	You Pay: $99.99 /	Save: $20 + FREE $198 CD
Int'l - One Year (6) $72	You Pay: $69.99 /	Save: $2

☐ **MX Developer's Journal**

U.S. - Two Years (24) Cover: $143	You Pay: $49.99 /	Save: $93 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $32
Can/Mex - Two Years (24) $168	You Pay: $79.99 /	Save: $88 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $49.99 /	Save: $34
Int'l - Two Years (24) $216	You Pay: $89.99 /	Save: $126 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $59.99 /	Save: $48

☐ **ColdFusion Developer's Journal**

U.S. - Two Years (24) Cover: $216	You Pay: $129 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189 /	Save: $75 + FREE $198 CD
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

☐ **WebSphere Journal**

U.S. - Two Years (24) Cover: $216	You Pay: $129.00 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189.00 /	Save: $75
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

☐ **PowerBuilder Developer's Journal**

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

☐ **WLDJ**

U.S. - Four Years (24) Cover: $240	You Pay: $99.99 /	Save: $140 + FREE $198 CD
U.S. - Two Year (12) Cover: $120	You Pay: $49.99 /	Save: $70
Can/Mex - Four Years (24) $240	You Pay: $99.99 /	Save: $140 + FREE $198 CD
Can/Mex - Two Year (12) $120	You Pay: $69.99 /	Save: $50
Int'l - Four Years (24) $240	You Pay: $120 /	Save: $120 + FREE $198 CD
Int'l - Two Year (12) $120	You Pay: $79.99 /	Save: $40

*WHILE SUPPILES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE

3-Pack

Pick any 3 of our magazines and save up to $210.00

Pay only $99 for a 1 year subscription plus a FREE CD

- 2 Year – $179.00
- Canada/Mexico – $189.00
- International – $199.00

6-Pack

Pick any 6 of our magazines and save up to $340.00

Pay only $199 for a 1 year subscription plus 2 FREE CDs

- 2 Year – $379.00
- Canada/Mexico – $399.00
- International – $449.00

9-Pack

Pick 9 of our magazines and save up to $270.00

Pay only $399 for a 1 year subscription plus 3 FREE CDs

- 2 Year – $699.00
- Canada/Mexico – $749.00
- International – $849.00

Subscribe Online Today **www.sys-con.com/2001/sub.cfm**

SYS-CON MEDIA

edge service functions for automated document interchange.

- **Business application services** provide the J2EE runtime environment for integration components developed as custom application components coded in Java and running in the application server environment. This layer of services provides the framework and runtime operating environment for managing custom application components developed with J2EE, XML, and Web Services programming models. These components provide the new functions required to implement fully modern business processes that meet the requirements of today's business environment. The implementation of these components as services within the WebSphere Integration Reference Architecture enables reuse directly across new solutions. Business application services include functions important to the traditional programmer for building maintainable, flexible, and reuseable business logic components, as well as the runtime integration for higher-levels of autonomic administration and management. The service functions reflected in this layer include:
  - **Component services**, which provide the runtime container management services that automatically handles issues such as object persistence, relationship navigation, and object query and transaction management in a J2EE framework.
  - **Core services**, which provide the runtime services, such as memory management, object instantiation and pooling, performance management and load balancing, event notification, availability, directory, and security as part of the J2EE, XML, messaging, and the Web Services programming model.
  - **Interface services**, which provide the services for bi-directional integration with databases, messaging systems, management frameworks, and enterprise applications.

- **Application and information access** provides the capabilities for interacting with third-party applications (for example, ERP, CRM), mainframe interfaces (for example, CICS, iSeries), custom applications (via technology bridges such as messaging, application programming interfaces, data handlers), as well as heterogeneous data sources (like RDBMS, XML, non-RDBMS data sources; for example, IMS, text files, and content management systems). This functional layer provides for the access interface to existing applications and data with support for transactional services and connection services for databases, messaging systems, and other data sources. Existing host-based applications and enterprise data are accessible from the ESB through a set of access services. These access services provide the bridging capabilities between legacy applications, pre-packaged applications, enterprise data stores (including relational, hierarchical, and non-traditional, unstructured sources, such as XML, text, and content management systems), and the ESB. This layer can integrate mainframe systems through multiple runtime solution patterns such as Web facing, communication-level integration, messaging, and Web Service-enabling. As applications and data implementations evolve to become more flexible participants in business processes enhanced capabilities of their underlying operating environments will continue to increase in utilization. The following services are isolated and enabled in this layer:
  - **Event detect services** provide event notification services based on the event framework that is enabled through the specific application/data source interface. For example, the creation of a new customer in a CRM system would generate an event that the ESB would distribute to the subscriber to this event type.
  - **On-ramp services** enable application integration patterns, including one-way inbound, request-reply, and solicit-reply message patterns to support application and data wrapper functions, including query execution planning and data retrieval to support data integration. For example, if one step in a business process needs to update the order, a message with the order data would be sent through the ESB to the mainframe CICS application, which would then return an acknowledgement.

Control Services

Control services provide the capabilities to effectively integrate people, processes, and information within the WebSphere Integration Reference Architecture. These services control the flow of interactions and data among people, process, and information services appropriate to the development and run time implementation of business processes:

- **Interaction services** provide the capabilities required to deliver functions and data to end users, meeting the user's specific usage preferences. Interaction services also provide the capabilities required to integrate various pervasive devices (sensors and actuators, for example) with the other components of the integrated system. This layer of services provides the external interaction services for uses (including browser and voice interaction) and pervasive device integration. The following services are provided as part of interaction services:
  - **Delivery services** enable a runtime interaction framework for users to interact with the integration components via portlets, voice, and pervasive messaging; this set of capabilities includes specialized technologies such as multi-device support, page aggregation, markup transcoding, language translation, and internationalization, as well as integration with wireless communication technologies to support mobile/remote user/device interactions.
  - **Experience services** provide the user-centric services responsible for the delivery of a robust user experience, including personalization, collaboration, authentication, authorization, self-service functions

(customization and registration), and single sign-on functions.
  - **Resource services** provide for the runtime management of the interaction components supporting, for example, security and entitlements via components such as pages, themes/skins, principles, and portlets.

- **Process services** provide the control services required to manage the flow and interactions of services that implement business processes. This layer of services can broker process execution through the aggregation of integration components to support coarse-grained business functions. In the WebSphere Integration Reference Architecture, the BPEL4WS standard is used to describe the orchestration of services. The following services are provided in process services:
  - **Choreography services** are composed of other services, and so choreography services provide the ability to execute these other services in a defined sequence and recover from errors.
  - **Transaction services** provide error recovery for both short-running and long-running processes. An entire process can be transactional (short-running). In a long-running process, each service is typically treated as a transaction if it changes data.
  - **Staff services** enable the integration of people into a process providing the creation of work items based on rules and information in a staffing directory. It supports task assignment, delegation, and administration in conjunction with interaction services.

- **Information services** provide the capabilities to federate, replicate, query, analyze, and transform data sources that can be structured (RDBMS or other data sources, such as IDMS or VSAM) or non-structured (such as spreadsheets, text files, document formats like Adobe PDF). This layer of services provides data integration and aggregation for heterogeneous data sources, thereby enabling transparent access to multiple data contexts using advanced techniques for distributed optimization. The following services are part of the information services layer:
  - **Federation services** enable the ability to aggregate data from traditional (such as RDBMS) and non-traditional sources (such as XML data stores, text data, and content stores) while leaving the data under the control of its native store.
  - **Replication services** provide support for automated real-time data synchronization and source/target transformation of structured data sources, enabling locality of access for data access regardless of source implementation.
  - **Transformation services** transform data from source formats into target formats for batch and record-oriented access/transformation.
  - **Search services** enable full query and search integration through conventional (database and structured data sources) and non-conventional (such as PDF, spreadsheets, word processing documents) data sources.

(Next month's concluding article will cover development services and the WebSphere Integration Reference Architecture in action.)

Acknowledgements

The author would like to acknowledge the work of the Worldwide WebSphere Technical Sales team in the creation and evolution of the WebSphere Integration Reference Architecture, and particularly the contributions of Bill Hassell, Bob Liburdi, and Bob Knaus. The author would also like to acknowledge Dan Wolfson, IBM CTO for business integration, for his thorough review of this article.

Resources

- M. Endrei (and others), *Patterns: Service-oriented Architecture and Web Services,* (Redbook), IBM TSO, 2004.
- C. Perks and T. Beveridge, *Guide to Enterprise IT Architecture,* Springer-Verlag, New York, 2003.
- G. Booch et al, *An MDA Manifesto,* MDA Journal, May 2004.
- M. Keen and others, *Patterns: Implementing an SOA using an Enterprise Service Bus,* IBM Redbook SG2463, July 2004.
- *IBM Service-Oriented Modeling and Architecture*, IBM Business Consulting Services Overview, 2004.

More on Globalization

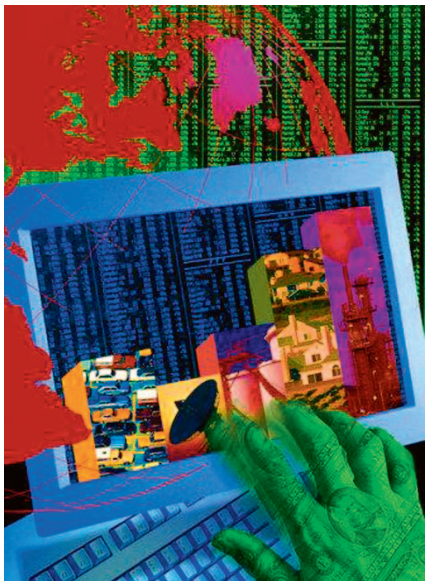
BY ROGER STRUKHOFF

Where technology marketers could at one time predict what was going to happen in Europe today simply by looking at what happened in the U.S. 18 months ago, then add different measures of additional months for other global markets, today they actually have to do some real work to understand the international scene.

The U.S. fell way behind most of Europe and much of Asia in wireless adaptation and usage, for example, some years ago. With organizations such as RIM and Eclipse.org spreading their message from bases in Canada, the U.S. is hardly the unchallenged innovation leader in North America these days, either. (Insert gratuitous hockey joke here.)

Indian technology parks, Chinese search, technology corridors in faraway lands such as Malaysia, Brazil, and France, and scattered hotspots from Bulgaria to the Philippines all demonstrate that the our sphere is becoming ever more well-rounded when it comes to technological innovation.

This global technology pool, fed by myriad streams, can make for some tough sailing. Take the recent European Parliament vote against new software patent law. Thought to be a close call until the eve of the vote, the measure lost overwhelmingly when the tide turned due to intense lobbying from European open-source advocates in alliance with political Greens across the Continent.



Or so it may have seemed. Was this just a simple open source vs. proprietary debate? Is there such a thing as a *simple* open source vs. proprietary debate? The measure's opponents said they feared the power of large, proprietary companies to blow smaller companies out of the water through sophisticated patent strategies rather than by pure innovation.

Most of the companies that would fall into this class today would seem to be the large, American technology companies. So this vote was probably an anti-U.S. vote, right? If so, it would come as no big surprise. Yet at least one large, proprietary, American technology company, Sun Microsystems, hailed the defeat as well.

This leads to the idea that the whole brouhaha has to do with stopping the one very large, proprietary, American technology company based in Redmond, Washington. Yet this company certainly has its own troubles these days, with a long-delayed Longhorn operating system, a recent propensity to step in it with respect to political matters, and unquestioned non-leadership in search, the most dynamic software market today.

Meanwhile, back in the U.S., open-source leaders are calling for existing patent and trademark law to be rewritten. In an era when intellectual property is replacing internet protocol as the de facto owner of the acronym *IP*, companies of any size should be shuddering at this thought.

Sure, anyone who has ever filed with the Patent and Trademark Office (PTO) in the U.S. is quite aware that the office is understaffed, pedantically process-oriented, and often uncomprehending. But who, then, should decide how to allocate and enforce intellectual property rights? Should this be put through some sort of oligarchic process? Should there be some sort of peer review?

And what sort of harmonization, to use a Euro-term, should there be to create, enforce, and defend IP rights on a global scale? This is one of the key issues involved in globalization, and it seems commonsensical that a fairly specific, enforceable global patenting and trademarking standard would be in the best interests of technology developers, large and small, in the U.S., the Faeroe Islands, or Mauritania.

It's time for a big sitdown, to see if the technologists of the world can agree to check their politics at the door, even if for a short time, and try to begin to figure out a problem that transcends the day-to-day challenges of application development.

Roger Strukhoff, editor-in-chief of ***WebSphere Journal***, is West Coast Bureau Chief for the SYS-CON News Desk, and President of www.wdva.com. He spent 15 years with Miller Freeman Publications and The International Data Group (IDG), then co-founded CoverOne Media, a custom publishing agency that he sold in 2004. His work has won awards from the American Business Media, Western Press Association, Illinois Press Association, and the Magazine Publishers' Association. Read his blog at http://www.rssblog.linuxworld.com. Contact him at roger@sys-con.com.

YOUR TESTING SUCKS.

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

LISA from iTKO.

Because everyone should own quality.

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

We understand.

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

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

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

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

LISA is different.

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

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

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

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


IBM
YOU
VS
APPLICATION
INTEGRATION
COMPLICATION
WebSphere
IBM WebSphere middleware is the easy and affordable way to help you overcome virtually any integration challenge. In fact, you can CONNECT ANY APPLICATION ON ANY PLATFORM WITH OPEN STANDARDS-BASED IBM MIDDLEWARE. Only IBM has years of proven, trusted experience helping customers build composite applications. The open standards answer to complex application, platform and IT infrastructure combinations, IBM WebSphere lets you re-use your existing IT investments. Imagine increasing efficiencies and making your business more flexible.
SEE HOW AT IBM.COM/MIDDLEWARE/CONNECT
IBM, the IBM logo and WebSphere are registered trademarks or trademarks of International Business Machines Corporation in the United States and/or other countries. ©2005 IBM Corporation. All rights reserved.