SYS-CON

WEBSPHERE APP SERVER • BPM• PORTAL SCRIPTING

WebSphere® JOURNAL

www.SYS-CON.com/WebSphere

The World's Leading Independent WebSphere Resource

MAY 2005 VOLUME 4 ISSUE 5


Middleware is Everywhere.
Can you see it?
1
4
Rational.
Key
1. Profiles code within IDE.
2. Develops globally, 24 x 7 x 365.
3. Shares test assets with U.K. team.
4. Manages projects without meetings.
5. Writes innovative patterns and code.
MIDDLEWARE IS IBM SOFTWARE. Rational software. The most complete set of tools for systems and software development, it makes it easier for you to create, to innovate, to excel. Seamlessly integrating into multiple programming environments, the Rational Software Development Platform makes the most of open standards. From requirements to deployment. So your team can work faster. Easier. Better.
Evaluate IBM software development tools. Learn more at ibm.com/middleware/build
ON DEMAND BUSINESS
IBM, the IBM logo, the On Demand logo and Rational are registered trademarks or trademarks of International Business Machines Corporation in the United States and/or other countries. ©2004 IBM Corporation. All rights reserved.


IBM
2
3
5

Weekend Warriors

BY ROGER STRUKHOFF

The amazing thing is that anything gets done at all. Software architects and developers, analysts and administrators, and C-level executives all share this core belief, whether they publicly state it or not. As we approach the summer months, one thinks of weekends in the Hamptons or the Wine Country, at the Shore or the Cape, maybe a trip to the Dells or to Mackinac, the Gulf or the Mountains.

Or maybe to the office. Blessed weekends, when the phones are not ringing, meetings are merely bad memories, and some real work can get done. One of the great ironies of IT is that it sucks in millions of people with the intellectual apparatus to handle the most abstract concepts and theories, often in several languages, people who are familiar with the great Discovers and Creators of human history, yet whose day-to-day existence is now consumed with writing, checking, and rewriting line after line of the driest and dreariest of low-level code.

"Seamless interoperability," ritually extolled in an invariably chirpy tone in innumerable press releases (and reflexively passed on by magazine editors), must ring as a cruel, hollow joke to anyone who has ever tried to get a process written in year 19xx at Company A on the Acme Platform to work with another written in year 200x at Company B on the Whizzo Platform.

This issue is, in a sense, dedicated to these people and those weekends. It represents a tremendous cross-section of what is going on within WebSphere in particular, and IT development and management in general.

In no particular order, let's start with Sveta Petrova's piece on WebCM. This relatively short but complex feature takes a high-level view and low-level discipline to present the challenges—and solutions—inherent in moving from Vignette to WebCM workflows. Well-known WJ writer Tilak Mitra chips in with the second part of his web services implementation piece, with a detailed, step-by-step tutorial on how to integrate business logic into the process. Serge Lucio contributes a piece about the importance of quality-testing SOAs early on, pointing out that SOAs are always in flux (and flux is your enemy)!

Chris Lockhart's contribution to world knowledge this month is a look at the portal scripting interface, in which he comes to the rescue of "the poor, overworked Portal administrator who doesn't want to the art of XMLA access…"

We also have the second half of our tremendous interview with IBM's Doug Wilson and Richard Gornitsky (which can also be found on SYS-CON.TV), and a contribution from Linfeng Yu about z/OS and Web Application Server. (I must admit the first time I ever heard the term z/OS, I thought someone was talking about the old Zilog chip days. I thought, now there's someone after my own heart. Then I woke up and found myself back in the 21st Century.)

We found Ade Rixon's "Putting WAS on Unix" in a serendipitous way, simply by surfing the web in search of ideas one blessed weekend. "But I only posted it on my website two days ago," he said. "How did you find me?" It's simple, we are omniscient, I reminded him, and therefore knowledgeable enough to know a good subject and good writing when we see it, and give it a much wider audience.

That's it. Nasty, difficult, highly useful material made simple in the hands of tremendous technologist/writers, presented for our audience of presumably the same. Have a good week, have a good weekend, and let us know if you have something to contribute!

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, KIM HUGHES
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, PERCY YIP, VINCENT SANTAITI
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

Putting WAS on Unix

Getting you from A to B by the quickest, easiest route

BY ADE RIXON

Ade Rixon is a UNIX System Administrator working in the UK on e-commerce and related systems. His personal web site is at http://www.big-bubbles.fluff.org. He may be e-mailed at info@big-bubbles.fluff.org

This article describes how to install and use IBM WebSphere Application Server (WAS) on Unix systems (chiefly Sun Solaris but most of the information is generally applicable). It's for Unix system administrators who must support WebSphere in an enterprise or production context.

It assumes that you're generally familiar with Unix system administration, TCP/IP networking, and the operation of any additional components you may require, such as Web servers and database management systems.

The goal is get you from A to B along the quickest, simplest route, where 'A' is the Unix system and a WebSphere installation CD and 'B' is an installed WebSphere application server with a working Web application. It deliberately avoids explaining J2EE technology, discussing the many implementation scenarios and other related subjects unless such information is explicitly required to complete a task.

Large-scale WebSphere deployments (i.e., the Network Deployment product) aren't described in detail either, although there's an overview of relevant issues towards the end.

We're talking specifically about WebSphere Application Server V5.1. Some of the information will also be valid for V5.0 and V6.0. Previous major releases of WebSphere differ markedly from the V5.1 release and are quickly becoming obsolete.

Note that the configurations chosen here aren't the only options and may not be correct or optimal for your situation. However, they have been proven to work and have been tested in production environments.

This piece was written in response to the lack of a direct, unambiguous guide to installing WebSphere on Unix. However, while it's designed to get you up and running quickly, it can't cover every aspect. You should expect to consult other sources where necessary.

The WebSphere Application Server Information Center is the complete, definitive guide to the product. It is supplied by IBM in HTML and PDF formats either online (see section 13) or as part of the product media set.

IBM's International Technical Support Organization (ITSO) also publishes a number of "Redbooks" that cover specific aspects of implementation. Of these, the most useful for WebSphere administrators is "IBM WebSphere Application Server V5.1 System Management and Configuration." It includes a platform-specific chapter for AIX installation; this information is also relevant to other Unix-based systems. (There is a related Redbook dealing specifically with Linux, "Installing WebSphere Application Server Version 5.1 on Red Hat Enterprise Linux Advanced Server Version 3.0.") The Redbooks can be downloaded from IBM's web site.

Of these two main sources, the Redbook is easier to follow and more immediately helpful than the Infocenter, which makes heavy use of hyperlinking that makes following a logical sequence harder. The Infocenter is, however, more detailed and comprehensive, fully searchable and useful as a reference.

IBM Press now publishes several books about WebSphere that cover administration and deployment topics more approachably than the official documentation. These are listed in the bibliography.

Getting Started

You will need at least one Unix server with an installed operating system listed by IBM in the "Prerequisites for WebSphere."

Ensure that your platform is up-to-date with all recently recommended patches and fixes before beginning. The server must be connected to a suit-

able TCP/IP-based network. If your WebSphere system will be part of a larger site that includes Web servers and database servers, these other systems should be installed ready for use in advance. The examples we use are from a Sun Enterprise 280R server running Solaris 8.

You will need the IBM-supplied CD-ROM (or download) labelled "WebSphere Application Server and IBM HTTP Server" for your platform. It's up to you to make sure you have the appropriate licensing and support for WebSphere. WebSphere Application Server doesn't require a license key or registration.

Download Fix Pack 1 (FP1) for WebSphere Application Server V5.1 Base from IBM's WebSphere support site. This will update the installed product to release V5.1.1.

If a later fix pack is available, download it instead. You're unlikely to need any of the cumulative or individual fixes initially, unless your development team has a specific requirement.

You'll also need a build of the application(s) you'll be running under WebSphere. WebSphere applications are normally deployed via Enterprise Application Repository (EAR) files, which are archives containing all the required files. EAR files are built and supplied by application developers. Along with the application, they should supply a list of additional requirements such as:

- Databases, including schemas and initial contents;
- Messaging services and queues;
- Key URIs, such as home pages, entry points, and administration pages;
- Minimum CPU, memory, and disk requirements;
- Any additional required setup (e.g., logging configurations, static content files, supporting utilities, and URI aliases).

If this is not available, you may need to ask for more information during the deployment. Generally, deploying WebSphere applications is easier if you have access to the developer(s).

Other Requirements

The WebSphere graphical installer requires X Windows. You must be able to view X11-based programs running on the host(s) on which you will be installing WAS. Either a local graphics monitor or a remote X11 server/emulator such as Cygwin/X will suffice.

A supported Web browser (generally Internet Explorer or a Mozilla variant) is required to administer WebSphere.

It's possible to install supporting products on the same system as WAS and reduce the hardware requirements in test and development environments. However, it's usually undesirable on production sites for performance, scalability. and security reasons.

Platform Resources

Ensure that each WebSphere server meets the minimum CPU, memory and disk space requirements given by IBM for running WebSphere V5.1. If possible, dedicate a separate large disk partition to the application log data.

Web Servers

Most J2EE applications are web-based, so you're likely to need a Web server or Web-serving infrastructure (such as a load balancer farm). While you can access applications on WebSphere directly using HTTP or HTTPS, this approach isn't recommended for production use since WebSphere contains no facilities for request logging or access control, nor will it perform as well as a dedicated Web server.

Read IBM's list of supported Web servers for WebSphere before selecting a suitable product. The Web server communicates with WAS using a component called the **HTTP Plugin** supplied with WebSphere. The plugin is available in a number of forms to support different Web servers. It can support:

- EAPI-compatible products (based on Apache, including IBM HTTP Server)
- NSAPI-compatible products (SunONE and others)
- ISAPI-compatible products on Windows platforms (e.g., Microsoft IIS).
- Lotus Domino Web server.

If your Web server supports an API compatible with at least one of these products, it should work.

Before proceeding, install and configure your Web server, including IP addresses, names, DNS entries, and secure certificates where required. Make sure that it has network access to your WAS nodes.

Database Server

Many J2EE applications use a database at the backend to store application and session data, and audit information. Databases are typically accessed over TCP using JDBC-compliant drivers, although WebSphere supports a number of other methods.

Some databases require client software on the WebSphere node to access a remote server. For example, for DB2 V7 servers, you have to install the DB2 client on the WebSphere node and catalog the remote DB2 nodes and databases. WebSphere uses JDBC to communicate with the *local* DB2 client, which in turn forwards the queries to a remote DB2 server via the network.

Before proceeding, install and configure any required database management system, and create and initialize the databases required by your applications. Make sure that remote access to the DBMS is possible from the WAS system.

Operating System

If you are not using a naming service such as DNS or LDAP:

1. Add the hostnames and IP addresses of any supporting servers to the /etc/hosts file on the WebSphere node(s).
2. Add the hostnames and IP addresses of the WebSphere

node(s) to the /etc/hosts files on the supporting servers.

If you're using a naming service, add entries for all the hostnames and IP addresses.

Installation

This section covers the installation procedures for WAS base edition and the fix pack.

MESSAGING SERVICE (MQ)

The J2EE 1.3 standard specifies that a **messaging service** must be available to applications. IBM meets this requirement in WebSphere by bundling a version of its MQ Series software with the product, also known as WebSphere Embedded Messaging or the WebSphere JMS provider. The applications you're installing may not need this feature; confirm this with the developers before installation.

IBM's installation instructions and examples assume that you'll install the MQ components. These require the creation of dedicated user and group accounts and some kernel configuration. However, this is an option and is unnecessary if your applications don't use a messaging service. Although the MQ components are selected for installation by default, it's safe to de-select them if they won't be used. It will save you effort and overhead.

You can install the MQ components separately after installing WebSphere, should they be required at a later stage. There doesn't appear to be a supported method for *uninstalling* the MQ components separately from the rest of WebSphere.

Here we assume that you won't install the MQ components. If this is not the case, read the relevant Infocenter sections carefully before proceeding.

Overview

We'll do a custom install of the WAS Base product. During installation, we'll ensure that only the required components are selected. After installation, we'll also install the most recent fix pack to update the product with later fixes from IBM.

Installing WebSphere Application Server

1. Insert the WebSphere CD-ROM into the drive on the server and mount it. (If this isn't possible, either mount the CD-ROM via NFS from a remote system or recursively copy the contents over the network to a directory on the server using rsync(1) or a similar utility.)
2. In the mounted CD-ROM directory, change to the subdirectory for your platform (e.g., sun/).
3. Run the install script as root:
   # **./install**
4. The installer first checks the current OS patch levels. It may warn about missing or obsolete patches; however, they may not be essential to the WAS operation (e.g., window manager and X11 patches aren't required if you don't run a desktop environment on the system). If in doubt, cancel the install and update the patches.
5. Select **Custom** setup.
6. At a minimum, select the following components:
   - Application Server;
   - Administration & all subcategories;
   - Performance and Analysis Tools & all subcategories.

   Note that WebSphere will only run under the Java SDK releases bundled or supplied by IBM.

   If you have a Web server installed on this system, also select 'Web server plug-ins' and indicate the type of server. If you'll be using the IBM HTTP Server version 1.3 on this server, select that component and ensure that the Plug-in for IHS 1.3 is also selected.

   If you want to test WebSphere with a demo application, select 'Samples.' Otherwise, don't install them. *Deselect* all other features, including subcategories not listed above.
7. Accept the default path for the installation directory unless your requirements are different (/opt/WebSphere/AppServer on Solaris).
8. Accept the default node name and host name.
9. Review your choices and proceed with the installation.
10. Registering the product is optional.

"You can access apps on WebSphere directly using HTTP or HTTPS, but it isn't recommended for production use since WebSphere has no facilities to request logging or access control, nor will it perform as well as a dedicated Web server"

Parasoft SOAPtest®

Keep Your Skills Ahead of the Crowd

Keeping your IT skills ahead of the crowd is not as difficult as most people fear. Staying on top of the trends may seem like a daunting task if, like most people, you assume that each new technology is a completely new invention that you must learn from the ground up. Fortunately, nothing is really all that new. Inventors typically create new technologies by studying existing technologies, then building upon them in ways that extend and improve them. 100% new technological advancements are very rare.

Inventors almost always leverage legacy technologies as they invent new ones. Why not leverage your own knowledge of those legacy technologies as you try to learn about the new inventions? To learn about new technologies as painlessly as possible, consider how each new advancement is similar to what you already know.

For example, consider Web services. Web services are a new trend, but — at a technological level — the parts of a Web service are not all that unique. Web services are based on remote procedural calls — messages sent to a server, which calls the requested function. RPCs were developed years ago, and are hardly a new concept. Really, the only "new" thing in Web services is the standard that is being used to write the application. If you break down Web services in this way, it's easy to learn about them. To continue with this process, you might next explore the payload requirements, the process for determining what function to call, and how the call works. As you can imagine, it's a lot more efficient — and interesting — to learn about a new technology based on its relation to familiar technologies than to learn about it by reading the specification cover to cover.

As always, the devil is in the details. But most details are critical only if you want to specialize in a given technology. For instance, if you want to specialize in Web services, you need to familiarize yourself with the details of Web service development. In that case, your next step would be to learn how to format the messages, how to expose Web services, and so on.

*– **Adam Kolawa**, Ph.D.*
Chairman/CEO of Parasoft

It's automated. It's fast. And it's the most versatile Web Services testing tool.

SOAPtest® enables you to deliver better Web Services in less time.

The simple fact is, no other Web service testing product can do what SOAPtest can do. From verifying functionality and performance to ensuring security and interoperability, SOAPtest automates all critical testing processes across the entire lifecycle of your Web service.

But don't take our word for it.... Go to www.parasoft.com/soaptest_WSD and try it for free—no commitment, no obligation. If you like it (and we suspect you will) just let us know. We'll be more than happy to help you and your development team get up and running.

For Downloads go to www.parasoft.com/soaptest_WSD PARASOFT®

Email: soaptest@parasoft.com Call: 888-305-0041 x1209

Features

- WSDL schema verification and compliance to standards
- Automatic test creation using WSDL and HTTP Traffic
- Data-driven testing through data sources (Excel, CSV, Database Queries, etc)
- Scenario-based testing through XML Data Bank and Test Suite Logic
- Flexible scripting with Java, JavaScript, Python
- WS-I Conformance: Basic Profile 1.0
- WS-Security, SAML, Username Token, X.509, XML Encryption, and XML Signature support
- WS-Security Interop testing emulator
- MIME Attachment support
- Asynchronous Testing: JMS, Parlay (X), SCP, and WS-Addressing support
- Windows Perfmon, SNMP, and JMX monitors
- Detailed Report generation in HTML, XML and Text formats
- Real-Time graphs and charts

Benefits

- Uniform test suites can be rolled over from unit testing to functional testing to load testing
- Prevent errors, pinpoint weaknesses, and stress test long before deployment
- Ensure the reliability, quality, security and interoperability of your Web service
- Verify data integrity and server/client functionality
- Identify server capabilities under stress and load
- Accelerate time to market

Protocol Support

- HTTP 1.0
- HTTP 1.1 w/Keep-Alive Connection
- HTTPS
- TCP/IP
- JMS

Platforms

- Windows 2000/XP
- Linux
- Solaris

Contact Info:

Parasoft Corporation
101 E. Huntington Dr., 2nd Flr.
Monrovia, CA 91016

www.parasoft.com

11. After installation, use the launch pad tool to start the server and verify its operation. Don't launch the administration console from this window.

    If you're installing IHS 1.x on a separate system, run the installer again there and select the IHS and plugin components only.

Installing WebSphere Application Server Fix Pack

1. If the default server is running (e.g., because you started it using the launch pad during the installation), stop it using the command:

```
# /opt/WebSphere/AppServer/bin/stopServer.sh
    server1
```

   Wait for the command to complete.
2. Unpack the fix pack in an empty directory using tar(1):

```
$ gzip -dc was51_fp1_solaris.tar.gz | tar -xvf -
```

3. Become the superuser and source the WAS setup script:

```
# . /opt/WebSphere/AppServer/bin/setupCmdLine.sh
```

4. If your WAS installation is already configured, back up its current state as a precaution using the command:

```
# /opt/WebSphere/AppServer/bin/backupConfig.sh
```

5. Run the update wizard:

```
# ./updateWizard.sh
```

   An X window will appear asking you to select the language for the installation. (Note that the installer may be sluggish on older hardware with long delays between screens.)
6. Continue past the initial information dialogue.
7. Select IBM WebSphere Application Server v5.1 from the product list and hit **Next**.
8. Select Install fix packs and hit **Next**.
9. The correct fix pack directory path should be displayed already. Hit **Next**.
10. Select the v5.1.1 fix pack (was51_fp1).
11. Proceed with the fix pack install. Don't interrupt the update.
12. Once install is complete, exit the update wizard. The unpacked archive files can now be removed.

The WAS command script versionInfo.sh (in the bin/ directory) can be used to verify the current release and fix level of the product.

Installing the HTTP Plugin

The HTTP plugin is a shared object that links to the Web server to handle browser requests for WAS applications. If the Web server is running on the same host as the application server then you can install it while installing WAS. If the Web server is separate, then you have to run the WAS installer on it too as above.

Different versions of the plugin are supplied for different Web servers. Use a version that's compatible with your Web server software. Some less common Web servers support third-party APIs such as the Netscape one (NSAPI) or the Microsoft IIS one (ISAPI); in these cases, use the plugin version most compatible with your server (e.g., use the iPlanet plugin for the Zeus Web server since it supports NSAPI).

This procedure assumes you're installing the plugin on a separate Web server. However, the dialogues described are identical to those that appear when installing the plugin as part of WAS base.

1. Select custom installation.
2. Select the appropriate HTTP Plugin and any required components.
3. You'll be prompted for the path to the Web server configuration file (e.g., httpd.conf for Apache/IHS). If this file doesn't exist yet, create an empty file with the correct name and give the path to that (e.g., /tmp/obj.conf for iPlanet/NSAPI). You can configure the Web server manually later, using the temporary file as an example.

After installation, also apply the WebSphere Application Server fix pack to update the plugin. Keep the plugin fix level synchronised with the application server fix level.

Initial WebSphere State

By default, WAS is installed with a single application server instance called **server1**. This server runs the administration console application ('adminconsole').

WAS topology and configuration is hierarchical. At the highest level is the *cell*. (The administration console can only manage a single cell.) A cell contains one or more *nodes* representing physical servers. Each node may contain one or more application *servers*. In the base edition, the cell can only contain a single node and both are named after the system hostname. However, you can create multiple servers under the node. (WebSphere Application Server Network Deployment is used to manage multiple nodes in a distributed cell.)

Certain aspects of WebSphere can be configured at any one of these levels; this is called configuration *scoping*. Duplicate entries in lower levels *override* those in higher levels, letting you create cell-wide 'default' settings that can be overridden for individual nodes and servers.

Warning: On installation, WAS contains several default settings in the *node* scope. Some of them have empty values. Nevertheless, they'll override any similar settings created manually in the cell scope unless they're deleted first. To avoid confusion, we'll make all

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

our initial configuration changes in the node scope, which is the default.

WAS also initially defines two *virtual hosts* that map the destination IP/port number pairs (*host aliases*) of incoming web requests to applications in the cell. These virtual hosts are called default_host and admin_host. The latter is used by the adminconsole application and maps requests for the host alias *:9090 (i.e.. any IP address on TCP port 9090) to the console. The former can be used by your own applications. Assuming WAS was installed on a fresh system, it defines host aliases for *:9080 and *:9443, which are the default ports for **server1**.

Starting and Stopping WAS

Application servers in the base edition are started and stopped individually using the startServer.sh and stopServer.sh scripts in the WAS bin/ directory. For example (assuming the WAS bin directory is in your PATH):

```
# startServer.sh server1
ADMU0116I: Tool information is being logged in
  file
     /opt/WebSphere/AppServer/logs/server1/
       startServer.log
ADMU3100I: Reading configuration for server:
  server1
ADMU3200I: Server launched. Waiting for
  initialization status.
ADMU3000I: Server server1 open for e-business;
  process id is 12995
```

The **server1** instance must be running for you to manage the WAS base edition since it contains the adminconsole application.

You may want to write a single wrapper script for the various WAS command scripts. Such a script is useful for controlling WAS during system boot and shutdown (See Listing 1).

LISTING 1

```
#!/bin/sh
#
# script to start and stop WebSphere Application Server
5.x
#

WAS_HOME="/opt/WebSphere/AppServer" # WAS install dir
SERVERS="server1 MyAppServer" # list of app servers

if [ ! -d "${WAS_HOME}" ]; then
 echo "$0: ${WAS_HOME} does not exist, aborting" >&2
 exit 1
fi

case "$1" in
'start')
 # increase resource limits
 ulimit -n 1024
 ulimit -s 16384
 for s in ${SERVERS}; do
  ${WAS_HOME}/bin/startServer.sh $s
 done
 ;;
'stop')
 for s in ${SERVERS}; do
  ${WAS_HOME}/bin/stopServer.sh $s
 done
 ;;
'status')
 ${WAS_HOME}/bin/serverStatus.sh -all
 ;;
*)
 echo "Usage: $0 <start|stop|status>"
 exit 1
 ;;
esac
```

Online Resources

- IBM WebSphere Information Center: http://www-306.ibm.com/software/webservers/appserv/was/library/
- IBM developerWorks WebSphere Application Server Zone: http://www-106.ibm.com/developerworks/websphere/zones/was/
- IBM Redbooks:www.redbooks.ibm.com/
- Sample configuration scripts: www-128.ibm.com/developerworks/websphere/library/samples/SampleScripts.html
- WSADMIN Primer: www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100421(written for WebSphere on z/OS but largely relevant to all platforms)
- WebSphere-World (a good product news site): www.websphere-world.com/index.php
- Cygwin/X: http://x.cygwin.com/
- Curl: http://curl.haxx.se/
- Lsof: ftp://lsof.itap.purdue.edu/pub/tools/Unix/lsof/
- rlwrap: http://utopia.knoware.nl/~hlub/uck/rlwrap/
- Nagios: http://www.nagios.org/
- Precompiled open source utilities in pkg format for Solaris can be obtained from www.sunfreeware.com/

References

- Roland Barcia, Bill Hines, Tom Alcott, and Keys Botzum. *IBM WebSphere Deployment and Advanced Configuration*. IBM Press/Prentice Hall PTR, 2004.
- Lavena Chan, Roger Cundiff, Shaun Lauzon, Christopher Mitchell, and Leigh Williamson. *IBM WebSphere System Administation*. IBM Press/Prentice Hall PTR, 2004.
- Kovari et al. IBM WebSphere V5.0 Security. Technical report, IBM Corp., 2002. sg246573.
- Roehm et al. IBM WebSphere V5.1 Performance, Scalability, and High Availability. Technical report, IBM Corp., 2004. sg246198.
- Sadtler et al. IBM WebSphere Application Server V5.1 System Management and Configuration. Technical report, IBM Corp., 2004. sg246195.
- Sadtler et al. Installing WebSphere Application Server Version 5.1 on Red Hat Enterprise Linux Advanced Server Version 3.0. Technical report, IBM Corp., 20

(Note: Part 2 of this article, to be published in the June 2005 issue, will cover server administration and advanced topics.)

NEW!

CodePro AnalytiX™

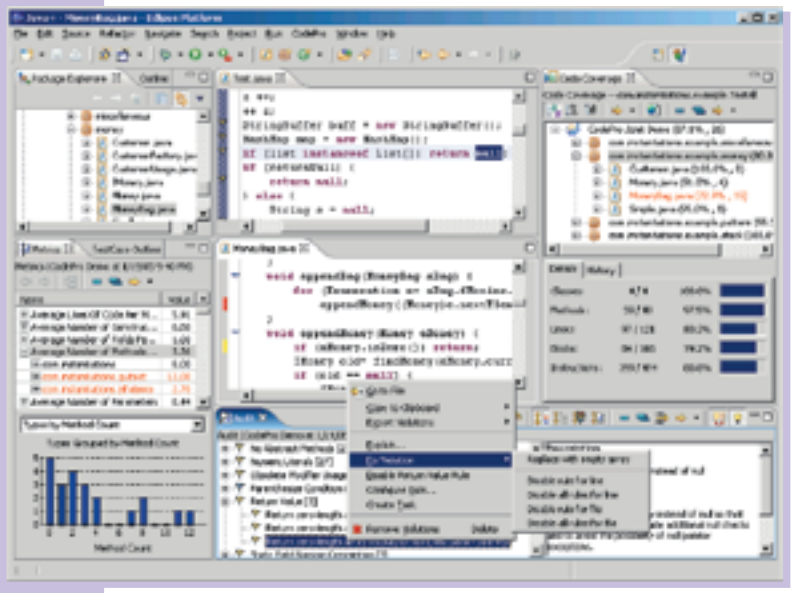
Continuous Collaborative Code Analysis for Rational®, WebSphere® & Eclipse

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

Configure the system via the command line

The Portal Scripting Interface

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

One of the great advantages of the WebSphere software platform is that it's been built with a great deal of flexibility. A product simply wouldn't bear the WebSphere name if there weren't several different ways to do things. WebSphere Portal Server is no exception. With the release of version 5.1 IBM has added another way to administer the configuration of the Portal. This is sure to delight the poor, overworked Portal administrator who doesn't want to learn the art of XMLAccess and wants to avoid the use of a Web-based administration interface all costs.

This new feature, named the Portal Scripting Interface, lets the portal administrator configure the system via the command line. It's an extension of the wsadmin command-line interface for WebSphere Application Server and so it uses a similar syntax including the ability to take JACL script files as input (hooray!). Believe me, this is an advance that many Portal professionals have been waiting for ever since the introduction of the wsadmin tool for AppServer.

The end result is an interface that automates portal admin tasks and eases the burden of making minute changes to nested Portal objects or transferring new portal configurations from a developer workstation.

The Portal Scripting Interface, which I'll call PSI from now on (not to be confused with pounds per square inch or some reference to psionic powers), is invoked from within the *WPS_HOME\bin* directory. The syntax would normally look like this:

```
WPS_HOME\bin\wpscript.bat
```

But of course it really isn't that simple. There are some implied parameters in that command. The first one is the connection type, or conntype. By default this value is SOAP, which indicates that the interface should connect to the Portal via the SOAP protocol. Another possible value could be RMI indicating that the interface should connect over IIOP. A third possible value could be NONE indicating that only the command shell should be launched and not explicitly connected to any running instance of the Portal (and not very useful for administering the portal).

The second implied parameter is the port on which to connect to the Portal. If you're using the default SOAP connection type, then the default value for the port parameter is 8882. In a Network Deployment configuration, you'd want to use the default port 8879 to make this connection. The SOAP connection port value of the server you're attempting to connect to can be viewed in the WebSphere Administration Console under Application Servers>WebSphere_Portal>End Points>SOAP Connector Address.

So an explicit string for launching the tool would look like this:

```
WPS_HOME\bin\wpscript.bat –
  conntype SOAP –port 8882
```

As you might expect, when WebSphere Security is enabled for the Portal, proper security credentials have to be supplied:

```
WPS_HOME\bin\wpscript.bat –conntype
SOAP -port 8882 –user wpsadmin -
password password
```

Once the PSI has been launched, you must actually log into the Portal you're attempting to administer. This command, executed in the PSI, uses the syntax of the underlying wsadmin interface for AppServer. Familiarity with JACL or wsadmin would help at this point but it isn't necessary. Suffice it to say that commands are entered in a hierarchical format. They simply represent underlying Beans that are being invoked to do particular tasks. For our Portal login command, we have to invoke the Portal bean. After invoking PSI log into the Portal with:

```
$Portal login wpsadmin password
```

Congratulations! You're now connected to the Portal and ready to issue administration commands.

If you're using the new virtual portal feature of WPS 5.1, you can log into your virtual portal using a subcommand of the Portal bean. Let's say your virtual portal URI is /wps/myportal/blueportal (where the "blueportal" part of the URI indicates the name of the virtual portal), then the following commands

```
$Portal setvp blueportal
$Portal login wpsadmin password
```

will get you logged into the desired virtual portal.

Get Help Fast

Each of the beans available in this tool have help options. If I wanted to get a list of all of the available help options for the Portal bean, I could simply type

```
$Portal help
```

This would return the top-level list of help for the Portal bean. If I was more curious about just the login method of the Portal bean (which we used to log into the portal), I could type

```
$Portal help login
```

This would return help information specific to that method. The available beans in the PSI are:

$Portal
$Content
$Layout
$Portlet
$Look
$Access
$PacList

Experiment with the help function on each of them to gain a better idea of the hierarchical structure of this interface.

Work Those Index Paths!

Let's say I have a portal page hierarchy that looks like this:

```
Content Root
  My Portal (label, uniquename:
   wps.myportal)
    Home (page, uniquename:
     wps.myportal.home)
    Corporate Directory (page,
     uniquename: wps.myportal.
     CorpDir)
    WorkPlace (label, unique
     name: wps.myportal.
     WorkPlace)
      Email (page, uniquename:
       wps.myportal.WorkPlace.
       Email)
      Docs (page, uniquename:
       wps.myportal.WorkPlace.
       Docs)
```

Let's think about this content-node hierarchy. If you were to think about the page structure in our example, you could assign some hierarchical values to the objects. For instance, the Content-Root we could say is at the root location of the tree, or simply /. If this were the case, then we could say that the My Portal content-node is one level down in the tree. Sort of like a directory off the root filesystem in Unix, this would be /0/1. The Home content-node, as a child of /1 would be at /0/1/2. This is called an index path.

Some examples of these index paths are as follows:

```
/     The root content node.
/0    The first child of the root
      content node.
/1    The second child of the root
      content node.
/0/0  The first child of the
       first child of the root
       content node.
/0/1  The second child of the
       first child of the root
       content node.
/0/2  The third child of the first
       child of the root content
       node.
0     The first child of the cur-
rent content node.
1 The second child of the current
   content node.
```

The Content Bean

Content-nodes in the PSI are referenced by the Content bean. This bean lets you search for a particular content-node, view its settings, update the settings, create a new content-node or delete a content-node.

If I was curious to know some details about the Corporate Directory content-node, I could invoke the Content bean to tell me about it. The Content bean uses the following syntax for a search:

```
  $Content find <type> <by>
<value>
```

So to find and display some info about the Corporate Directory node with uniquename wps.myportal.CorpDir, I would use:

```
$Content find any uniquename "wps.
  myportal.CorpDir"
```

More specifically, since I know the CorpDir is of type 'page,' I could execute a more concise search:

```
$Content find page uniquename
  "wps.myportal.CorpDir"
```

There are several different kinds of searches you can do. Executing

```
$Content help search-types
```

will show you these different searches. Keep in mind the help function for the Content bean is always available using:

```
  $Content help
```

Okay, so big deal, you found the content-node. It's far more interesting if you were to display some other info about this node. Before you can display other attributes of the node, you have to "select" the node. Finding it isn't enough, you have to "select" it before running an additional command against it:

```
$Content find any uniquename "wps.
```

```
myportal.CorpDir" select
```

Now that you've got it selected, try any of the following:

```
$Content get type
$Content get "wps.myportal.
  CorpDir" id
$Content current
```

The ID of the content-node (or any portal object for that matter) is the UID or Unique ID of the object. With this interface, most of the actions (get or set) are invoked against the UID of an object. Once you have this id, some of the operations become easier.

There are a few special id values that we can use. The most useful is called 'the root.'

```
$Content select the root
```

This command will select the root content node. Second in usefulness is the 'the parent' special id:

```
$Content select the parent
```

This command will select the immediate parent of the currently selected content node.

Let's say for example that the unique ID of our CorpDir node (the UID that was auto-generated by the Portal when we created the page) was found using one of the commands above. Let's say for argument's sake that this id is *_6_00KJL57F9D04J770_D*. We could then issue some other commands such as:

```
  $Content get "wps.myportal.
   CorpDir" id
$Content get _6_00KJL57F9D04J770_D
  themename
$Content set _6_00KJL57F9D04J770_D
  theme "Finance"
$Content get _6_00KJL57F9D04J770_D
  position
```

If we were interested in what was underneath our WorkPlace content-node, we could select it and then search on that content-node for objects called compositions (or pages to you and me):

```
  $Content find label uniquename
    wps.myportal.WorkPlace select
  $Content search composition
```

This will return a list of all the pages (or compositions) contained under the WorkPlace label.

But surely this isn't the exciting part. No, we're much more interested in creating some content-nodes instead of merely displaying information about them. Luckily for us, the Content bean has a method called 'create.' Let's use our CorpDir content-node as an example (we retrieved its uid in one of the above examples):

```
  $Content select _6_
    00KJL57F9D04J770_D
  $Content create composition
    "SubPage" html
```

This will select the CorpDir content-node and then create a page underneath it called SubPage with html as the supported markup.

```
  $Content select the root
  $Content create label "NewLabel"
    html select
  $Content create composition
    "NewSubPage" html
```

This sequence first selects the content root, then creates a new label underneath it and selects this new label, then creates a page underneath it. By contrast, we could also delete the content-nodes:

```
  $Content delete <id>
```

This command would delete the content-node with the id specified. For safety, the system doesn't let you delete a content-node with children. Gotta make sure those kids have parents!

Lens on Portlets

Even though we can't affect portlets directly using the PSI, we can definitely get some good information about the portlets defined in our portal repository. To do so we use the Portlet bean.

```
$Portlet search webmodule namehas
  "News"
```

Seems pretty straightforward. This command will execute the Portlet bean with the search method against objects of the webmodule type whose names contain the word "News." Once we have those objects returned, we could take the uid of one of them and gather more info:

```
$Portlet get webmodule <id>
  contextroot
$Portlet get webmodule <id> name
```

Of course, the Portlet bean also has a help function (as do its methods) in case you get stuck.

Sadly, there's no "set" capability with the Portlet bean. Perhaps we should start a campaign to get that functionality added…

Layout and Hierarchy

A bean that's useful for manipulating the layout of the content-nodes is the Layout bean (rather well named, I

"PSI will probably supplant XMLAccess as the primary automated administration interface"

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

think). With the Layout bean, we can use the index path to manipulate our content-nodes. The Layout bean has several methods (that can be seen by executing $Layout help). One such method is 'move.'

```
$Layout move to 0
```

This command would move the currently selected content-node to the root position.

```
$Layout move by 1
```

This command would move the currently selected content-node 1 level down the tree. By extension, moving by -1 would move the node up the tree one notch. This could be useful if you didn't have a lot of nested pages and whatnot under the currently selected content-node. In the event that you had an incredibly complex content-node tree, you could use another of the Layout bean's methods to transfer the currently selected content-node and all of its children to another parent:

```
$Layout transfer <node id> to
  <new parent node id>
```

In our page structure example, let's pretend that we got our uid values for the WorkPlace content-node *(_6_00KJL57F9D04K630_A)* and the Home content-node *(_6_00KJL57F9D04K219_C)*

```
$Layout transfer _6_
  00KJL57F9D04K630_A to _6_
  00KJL57F9D04K219_C
```

Once we completed this command, our new page structure would look like this:

```
Content Root
   My Portal
      Home
      WorkPlace
         Email
         Docs
      Corporate Directory
```

We also could have executed the "adopt" method as follows:

```
$Layout select _6_
  00KJL57F9D04K219_C
$Layout adopt _6_
  00KJL57F9D04K630_A
```

By first "selecting" the node we wish to have perform the adoption, we can then instruct it to do so. Very cool.

And now for something about the composition of these content-nodes that we're adopting and transferring all over the place. These content-nodes are comprised of rows and columns that together are called containers. Inside these containers we find our portlets that are known as controls.

The Layout bean can be used on both the containers and controls of a content-node.

Issuing the command below will give you the index paths of the potentially very complex layout of containers and controls on a page:

```
$Layout index
```

Appending a uid will give you the absolute index path of that object:

```
$Layout index <id>
```

If I start off with a blank page that I've created using the Content bean, I would definitely want to use the Layout bean to create horizontal or vertical containers (rows or columns) and controls (portlets) in those containers. If we were to use the CorpDir page as an example:

```
$Content select _6_
  00KJL57F9D04J770_D
$Layout create container
  horizontal select
$Layout create control <uid of
  portlet to add>
```

This sequence selects the content-node we want to add the portlet to. It then creates a new row on the page and selects it. The last step is to add the portlet to that row.

If we decided later to add some other portlet to this page, we wouldn't want to have to delete the whole thing and re-create it. So in that case we could simply add to our page:

```
$Content select _6_
   00KJL57F9D04J770_D
$Layout index /0 select
$Layout create control
    <some_portlet_id>
```

This sequence selects the CorpDir page by its ID. Next we select the /0 index path of the selected object. This will return us the parent container of the page (row or column we don't much care, as long as it is the top level object on the page). Lastly we add a new portlet by using its ID.

And Onward...

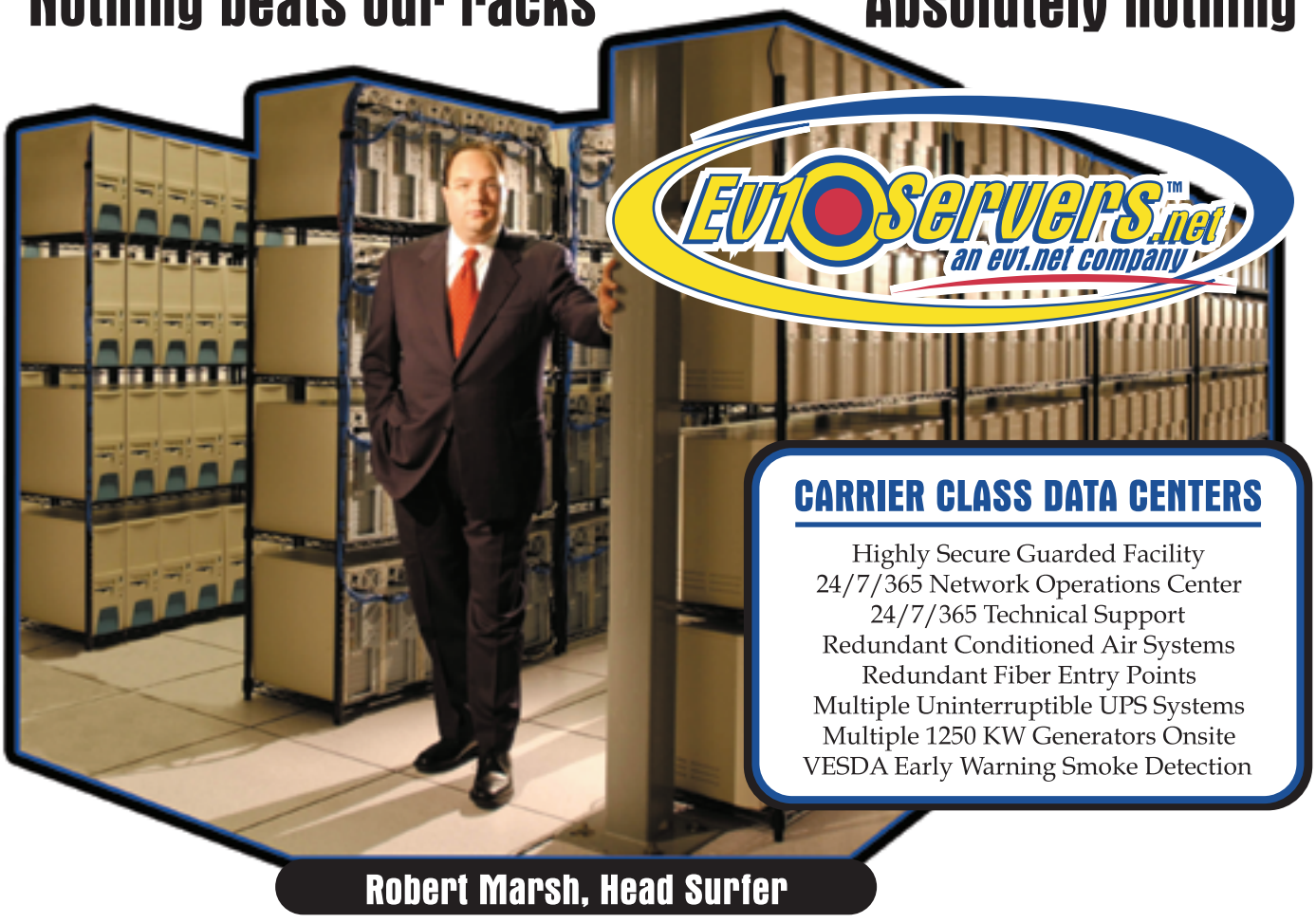
There are other beans that can be used to impact our portal configuration. We can use the Access bean to read in permissions or the PacList bean to assign them to various objects. And while I reviewed the major beans that one would use to administer the configuration of the portal, my comments are by no means exhaustive.

As with all things WebSphere-related, the proof is really in the hands-on tinkering. Go install WPS5.1 and play around with the Portal Scripting Interface. Before long you'll be doing more advanced tasks such as creating entire JACL scripts composed of many actions and reading them all in at once.

This interface is new. It currently has some limitations. But the direction is pretty clear. In future releases expect the PSI to increase in importance as new functions and beans are added and expanded. In fact, I would expect PSI to supplant XMLAccess as the primary automated administration interface.

There's some information in the WPS 5.1 InfoCenter regarding this new tool. It's currently located at: http://publib.boulder.ibm.com/infocenter/wp51help/index.jsp?topic=/com.ibm.wp.ent.doc/wps/ad_psi.html.

Nothing beats our racks Absolutely nothing



START YOUR OWN WEB HOSTING BUSINESS TODAY!

from **$299*** *Instant Activation!*

Dedicated Server

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

Over 20,000 Servers!

1-800-504-SURF | ev1servers.net

PLESK7 RELOADED
Preferred Control Panel

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

Flux is good for SOA, but it can also be an enemy

Don't Wait to Test SOA Applications

BY SERGE LUCIO

Serge Lucio is a product strategist at IBM Rational who focuses on service-oriented architectures. He has more than 10 years of experience primarily in the areas of model driven development and software verification. He is one of the creators of the Hyades project, now known as the Eclipse Test and Performance Tools Platform.

A lot has been said about the opportunities presented by service-oriented architectures (SOAs), especially their ability to enable business flexibility in an interoperable, technology-agnostic manner. But little has been said about verifying the functional quality of these SOA applications. As many organizations start delivering their first SOA applications, they realize that the flexibility they're gaining dramatically impacts the way software quality has to be addressed. The days of well-known test configurations are gone. A successful SOA is always in flux, and flux is the enemy of quality assurance engineers!

SOA is about making the IT fabric adapt quickly to business needs. This agility impacts the way organizations tackle how they verify their applications' quality. We will discuss the dynamic nature of service-oriented development and its implications for ensuring SOA applications' quality. We will highlight and describe the key activities needed to verify and test SOA applications. And we will stress the importance of addressing quality early in the lifecycle and explain how test automation plays a key role in this new operating environment as well as the need to do on-going tests once applications are deployed.

Testing SOA applications is not a trivial variation of traditional testing. SOA is blurring the line between development time and runtime, changing the very nature of what we used to call an application. An SOA application is constructed by composing and orchestrating loosely coupled services. A key SOA architectural principle is that service consumers have no knowledge of their service providers' implementation, making it virtually impossible to identify the closure of an application. In other words, trying to "box" an application is like pulling strings in the dark: You don't know when you'll get to the end of it.

Figure 1 shows a simplified view of an SOA application with its different layers. At the bottom, application interfaces representing business functions are exposed as services. These services get composed into higher-level business services that get choreographed to deliver the business value of an SOA application right at the top.

Of course, the reality of an SOA is not a clean, layered stack of services, but rather a network of services. Compared to traditional component-based apps, this makes incremental integration testing more challenging, and calls for thorough unit testing of individual services. As for any component-based application, an SOA application quality is as good as its weakest link. Figure 2 exemplifies the potential impact of poor-quality services as they get integrated into higher-level services. The higher in the stack, the more likely services will fail due to the poor quality of an underlying service; the more difficult it becomes to analyze the problem. Testing services as units is not an option.

Thoroughly Testing Services Is Not an Option

Unit testing services is obviously a critical activity in ensuring the functional integrity of an SOA application. To a great extent the unit testing of services is similar to testing regular software components. But one needs to address some specifics of an SOA application when it comes to testing services.

First, interoperability testing is an important step in the unit testing process. As SOA applications comprise a number of services developed in an organization or provided by third-party vendors, a number of integration issues may arise from incompatible interpretation of services messages. Taking the example of SOAP-based

Web Services, it's highly recommended to verify services implementations against pre-defined profiles.

Tools exist to facilitate this task. For instance, the Web Services Validation Tooling is an Eclipse technology project (see http://www.eclipse.org/wsvt/index.html), which delivers capabilities to verify Web Services against profiles defined by the WS-I Organization (http://www.ws-i.org/). Specifically, service specifications captured in WSDL, as well as SOAP messages, can be verified, preventing a number of interoperability issues that could arise during integration.

Another important difference between services and regular software components lies in the various usage models services can expose for their consumers: from the simple Remote Procedure Call usage model to Request-Response to Fire and Forget, etc.

When dealing with Remote Procedure Calls, the testing approach can be very similar to regular unit testing. Proxies generated by development tools such as Rational Application Developer can be leveraged to test a service much like a regular class, using simple testing frameworks such as JUnit.

But even with RPC services, the nature of the unit testing needs to be a little different because services and regular Java classes are different:

1. Basic services are often stateless.
2. Services are designed to promote loose coupling, hence they expose a limited number of operations (verbs) with fairly granular data

This translates into unit tests that need to be very data-driven as opposed to state-driven. In other words, what's important is the variability of the data rather than varying the sequences in which operations are invoked.

Equivalence partitioning is a suitable technique for creating data-driven tests. Glenford Myers described it for the first time in the "The Art of Software Testing." The technique is simple enough to be applied manually, but Rational Application Developer simplifies it by providing dedicated tools to edit and verify the test data. Figure 4 shows a screenshot of a test data table.

Besides RPCs, request response is a very common usage model for services. It enables simple short-lived asynchronous execution of services. The challenge with such model is that the service response comes in the form of an invocation of a service operation with some meaningful data. Meaning that the test driver needs to stub this operation to do the appropriate data verifications. Figure 5 shows a test configuration for a service using a Request-Response model.

Similarly, fire and forget presents unique challenges. The invocation of a service generally won't have any visible effect on the consumer side, but instead will perform some action on the back-end such as a database update or a service invocation. In some cases the test driver could directly verify these actions by accessing a resource that was modified by the service under test. But very often one needs to stub out external services as shown in Figure 6 to verify the interactions between the service under test and some other components or services.

In reality, most usage models require stubbing of services or regular components, but not simply dummy stubs. They need to be smart enough to simulate behaviors, to check data, etc. A number of Open Source frameworks can be leveraged for this kind of stubbing. The most popular ones are EasyMock and MockObjects. Rational Application Developer adds a number of capabilities to these frameworks by enabling a quick definition of stubs behav-

FIG 1: A LAYERED SOA APPLICATION

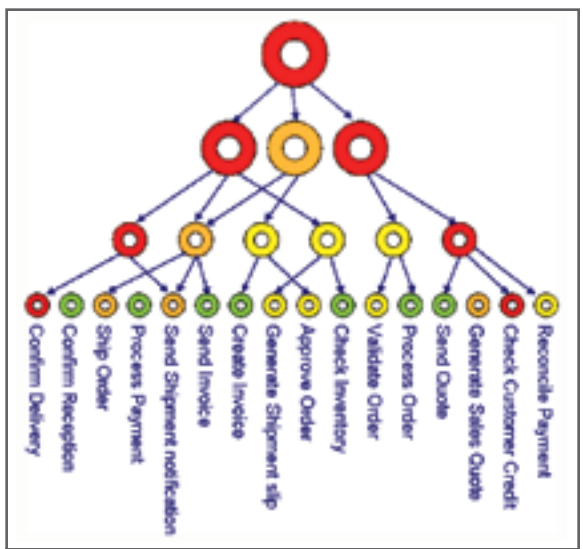


FIG 2: THE IMPACT OF POOR-QUALITY SERVICES ON AN SOA APPLICATION

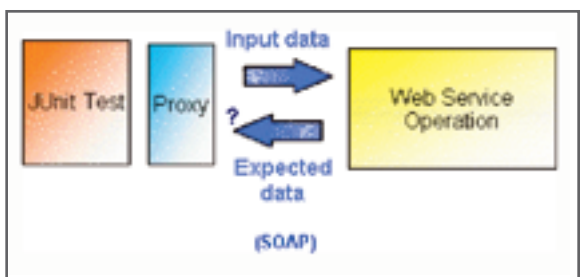


FIG 3: UNIT TESTING OF AN RPC SERVICE

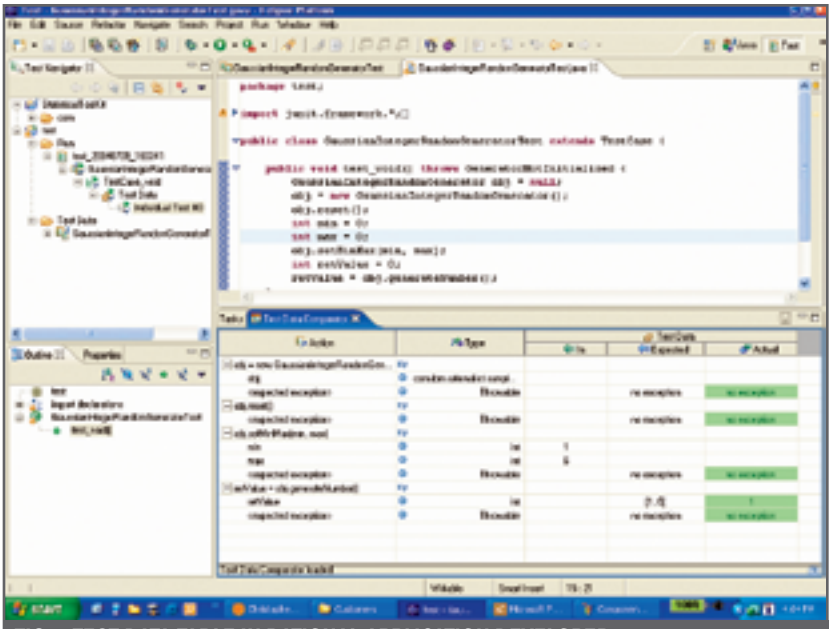


FIG 4: TEST DATA TABLE IN RATIONAL APPLICATION DEVELOPER

ior based on data partitions. Very much like business rules get defined in Websphere Business Integration Modeler, one can simply define the behavior of a stub based on the data values used by the consumer and associate a response to this input.

Independent of these usage models, testing services doesn't stop with testing the operations a service exposes. An SOA application will likely rely on services developed by other development teams as well as third parties. To facilitate the integration between the services developed by a given team with externally provided services, robustness testing must be done on service consumers. Figure 7 shows how stubbing can be leveraged to do service consumer testing.

Stubbing definitely plays an important role in the process of testing services. But its usage doesn't stop there.

Put Some Focus on Test Lab Provisioning

The number of interdependencies in an SOA application makes it virtually impossible to test a "complete" SOA application in a test lab. To mitigate that fact, stubbing techniques can be applied to isolate a portion of this network of services as shown in Figure 8.

There is an obvious issue with this approach. The quality of an SOA application is only as good as its weakest service. Stubbing too-coarse-grain services will yield invalid test results because many services won't be tested in the process. Stubbing too-fine-grain services will make the test lab setup increasingly complex because a number of services (real or stubbed) will have to be deployed and configured to mirror as closely as possible the real environment.

When it comes to the functional validation of an SOA application, the deployment and configuration of the application itself in a test lab becomes a huge burden. Setting up the lab may require dozens to hundreds of applications/services to be installed and configured in a reproducible way, mimicking as closely as possible the real production environment. Given the increased agility of development teams and shorter delivery cycles, the manual setup of such a test lab isn't viable; automation is a must.

One could leverage scripting techniques such as those provided by ANT to achieve such automation, but this kind of setup is likely to require complex actions orchestrated on a number of platforms, resilient to platform re-starts. Some Open Source tools can be leveraged to orchestrate such tasks.

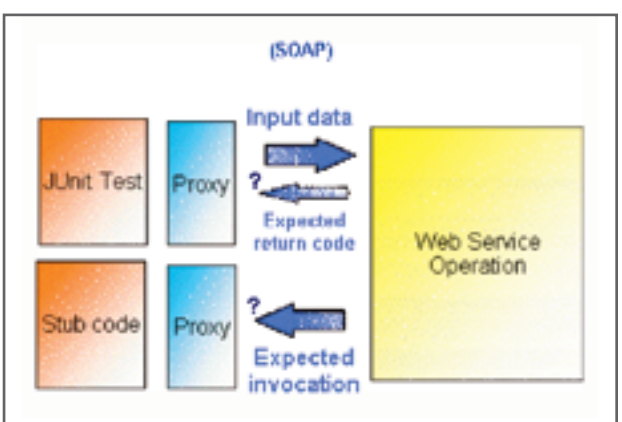


FIG 5: TEST CONFIGURATION FOR REQUEST RESPONSE SERVICE

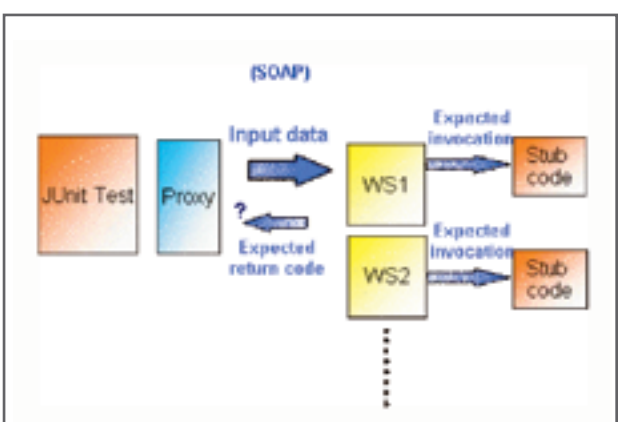


FIG 6: TEST CONFIGURATION FOR FIRE-AND-FORGET SERVICE

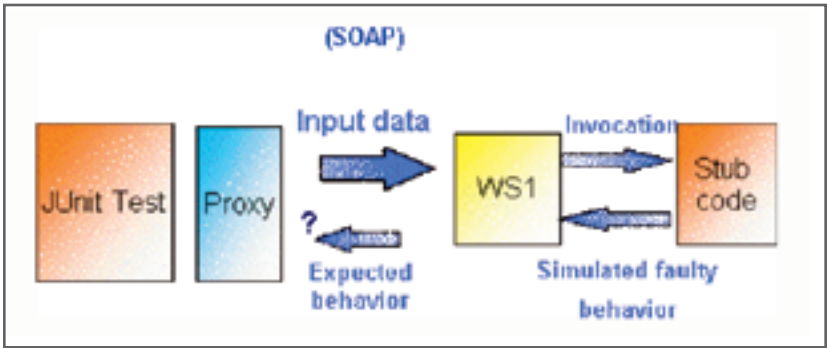


FIG 7: UNIT TESTING OF A SERVICE CONSUMER

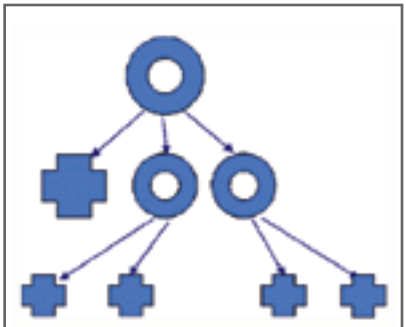


FIG 8: SOA APPLICATION WITH STUBBED EXTERNAL SERVICES

Among others, the Eclipse Test and Performance Tools platform (http://www.eclipse.org/tptp/) provides a framework for managing the choreography of scripts running on multiple platforms. It supports a number of operating systems, but its use is likely to be restricted to the test lab.

A great alternative is simply to leverage the same tools used to deploy the SOA applications in production. Expanding the use of these tools to support automated deployment in a test lab has several benefits. If they're used in production, they can probably get the job done in a test lab. Leveraging the same tool probably means that the test lab will also partially validate the way an SOA application will be deployed in fine.

Tivoli Provisioning Manager is an example of such a solution. As shown in Figure 9, it can orchestrate installations using advanced workflows that can include a number of tasks ranging from network to operating system to middleware to application to services installation and configuration.

But no matter how closely a test lab mirrors the operations environment, it's likely that a small number of stubs will still be required, for instance, to isolate the SOA app from third-party services. This means that the functional test in the test lab is only as good as the lab mirrors the real production environment. In many cases, it's better to expand the functional testing efforts beyond the test lab to verify the as-deployed SOA application. When testing is done in a production environment, test automation becomes a must since organizations need to minimize business downtime.

Invest in Test Automation

In reality, the challenges associated with getting a full-fledged application set up in a test lab are not the only motivator to testing SOA applications in production. The dynamic nature of an SOA app implies that the services it comprises will change over time as the services evolve to meet the needs of other SOA applications. The only way to verify such dynamic systems is to do regular functional verification of the applications in production. This is the best way to guarantee the sanity of applications at all times.

This requires not only that the tests be fully automated, but also that the test scripts encompass end-to-end business processes. As opposed to traditional applications, which tend to address the vertical needs of an organization, SOA applications are very horizontal. Business processes span multiple silos, involving multiple persons over a period of time that

"Testing SOA apps isn't a trivial variation of traditional testing; SOA is blurring the line between development time and runtime, changing the very nature of what we used to call an application"

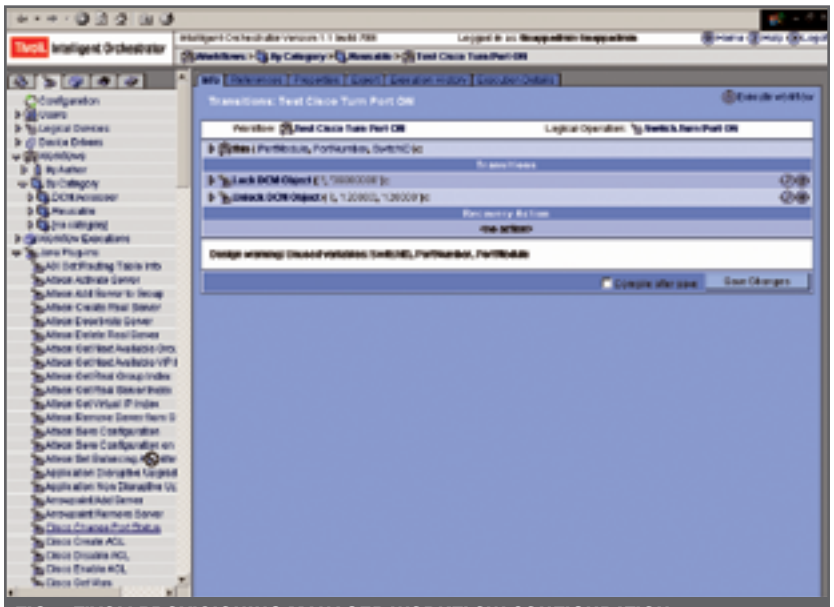


FIG 9: TIVOLI PROVISIONING MANAGER WORKFLOW CONFIGURATION

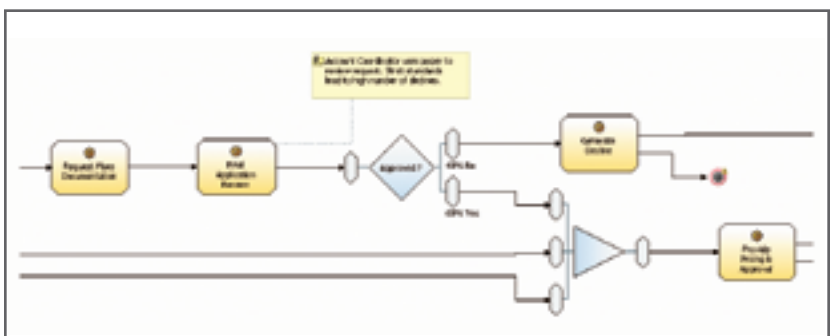


FIG 10: BUSINESS PROCESS SAMPLE

can span days. Testing such applications requires interactions with a number of user interfaces, ranging from corporate applications to customer-facing Web applications to e-mail clients, etc. This is making creating and executing test scenarios increasingly complex and longer. It also requires a careful selection of test scenarios among the multiple business process flows.

Information captured during the business-modeling phase can help testers determine the critical paths in a business process and prioritize the test scenarios that need to be created. As shown in Figure 10, Websphere Business Integration Modeler lets business analysts capture probability information about decisions. Combined with process simulation, this gives testers powerful insight into the various paths that may be critical to test.

In general, test plans should be developed early in the lifecycle. The best approach consists in formalizing detailed sequences of steps and verifications for every manual tasks defined in the business processes. These different sequences are then "orchestrated" into test scenarios targeting multiple objectives. Once test plans are complete, a review by a business analyst is critical to ensure the proper coverage of the business goals.

A number of tools can be leveraged to do such planning. Word processors tend to predominate. Despite their attractiveness, word processors don't foster reuse among multiple test scenarios. A lot of content is generally duplicated making the maintenance of the scripts very tedious. Rational Manual Tester can easily be leveraged to overcome this challenge.

With Rational Manual Tester, each sequence of steps defined for a manual task is captured as a test script. These elementary test scripts are then easily woven into longer test scenarios up to an entire business process without duplicating content. As shown in Figure 11, test scripts maintain references between each other; greatly simplifying the maintenance of test scenarios. The only update needed when a manual task changes is to updats its associated script. All the test scenarios are automatically up-to-date, reflecting the change.

As applications get delivered in the test lab, it's appropriate to automate the test scenarios. Trying to automate them before the applications are delivered is inefficient. Scripts are defined in a very abstract way and need to be bound to the concrete implementation, especially to user-interface controls. Scripts are also very specific and may not work if the actual implementation differs from the original user interface specification.

A number of commercial solutions exist to automate test scripts. A key to the success of such automation rests in how many user interfaces technologies the tool supports. Business processes will likely involve a number of user interfaces ranging from "green screen" applications to different kinds of Web applications (from ASP.NET to JSPs to JSF to portlets) to native applications such as mail clients to pervasive devices. Testing an end-to-end business process will likely require that all these user interfaces be driven by the automation engine.

Rational Functional Tester strikes a good balance between UI

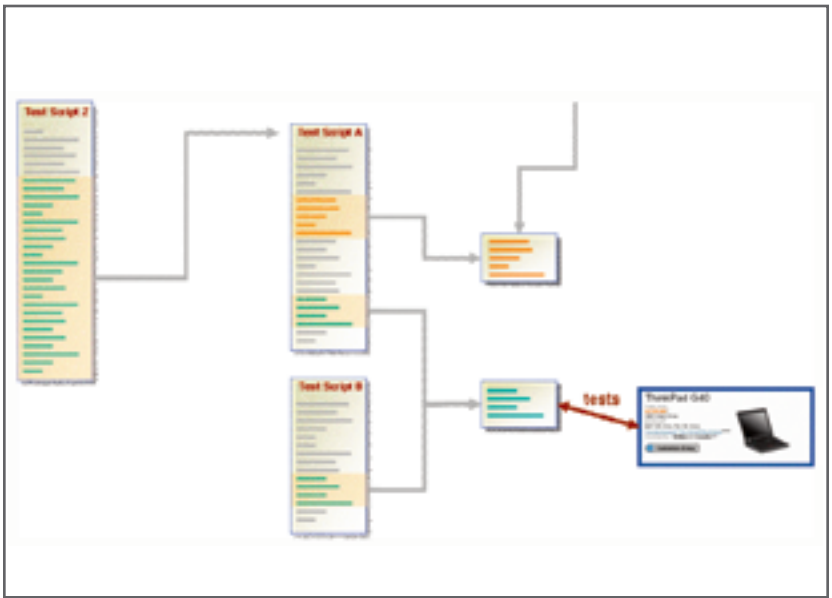


FIG 11: TEST SCRIPTS WITH RATIONAL MANUAL TESTER

technology coverage and openness. In particular, it offers scripting using non-proprietary languages (Java or VB.NET) making it flexible enough to accommodate most conditions. Besides that, test scripts can be easily recorded on a combination of different user interfaces, potentially built with different technologies. This makes the orchestrated verification of an end-to-end business process virtually seamless. Furthermore, test execution can be triggered automatically from the command line, making it easy to integrate as part of an end-to-end build/deploy/test process.

Another key criterion in selecting a test automation tool is its flexibility particularly in conjunction with monitoring solutions. As already discussed, on-going verification of an SOA application is a must. So the ability of a testing tool to integrate with scheduling tools such as Tivoli Monitoring for Transaction Performance or tracing tools such as Wily or Tivoli Monitoring is key to its selection.

Going Beyond Functional Testing

Functional testing is definitely the core of any testing effort, but a number of testing dimensions need to be considered above and beyond that, especially in the context of SOA applications. Performance and security are probably among the most important.

SOA applications are generally based on long-lived business processes, but still need to meet user needs. This can translate into tight response times for business processes feeding user interfaces. Chatty services and large message payloads can greatly impact the performance profiles of an SOA application. So performance verification needs to be dealt with as early as design time and need to be a prime consideration of QA teams.

Another key dimension of SOA quality is definitely security. One of the core values of SOA is in enabling applications to be quickly consumed by other applications either inside or outside an organization. Publishing a service to a third party opens the door to an organization's IT systems. Because services offer a programmable interface, they make an easy target for a hacker. Vulnerability testing is key for applications available "outside the firewall."

A How-to on migrating from Vignette

Making the Move to IBM Workplace Web Content Manager

BY SVETLANA PETROVA

As a senior member of Prolifics, Svetlana Petrova is extremely accomplished in portal design, development, customization, and the creation of portal migration strategies for IBM's latest versions of WebSphere Portal. She is a member of the highly specialized team of WebSphere experts retained by IBM to solve its customers' most complicated technology problems, as well as a lead J2EE developer with in-depth knowledge and experience in developing, implementing, and supporting leading-edge commercially distributed software systems and n-tier Web/GUI applications.
spetrova@prolifics.com

In the last Pro-Spective we detailed the benefits of IBM's Workplace Web Content Management (WebCM) solution and its powerful integration with IBM's WebSphere Portal Server. Many of you may already have another Web content management solution but are interested in migrating to WebCM to benefit from the additional value. This article details the repeatable steps involved in migrating from Vignette to WebCM, and a redeployment strategy.

Since Vignette is a database-driven product, the advantage of its open DB schema is that it gives you an easy way to extract data. After gathering the required information on Vignette database tables and fields, data (such as navigational categories and content of different types) can be analyzed and easily mapped into the site structure and content objects in the WebCM Repository.

By analyzing the Vignette CMA (Content Management Application), you can:

- Identify Vignette content types with their fields and their mapping into corresponding WebCM content templates that will be re-created using the WebCM authoring environment (an easy and quick process).
- Identify content categories – other than navigational – that will be re-created using the WebCM authoring environment as taxonomy(s) with categories.

Analyzing the Vignette workflows provides the information needed to recreate corresponding workflows in WebCM. The corresponding Vignette database tables provide information about each content workflow status.

Repeatable Migration Process

The data migration process is a repeatable process and can be automated by creating a migration tool. One way to implement the tool is as a JSP with an actions menu for the various processes, such as export navigational categories, export content, etc. Keep in mind that the migration tool needs to be deployed on the same WebSphere Portal server instance where WebCM is installed.

With a migration tool, it's possible to create very basic actions and configure them to do particular export tasks. For each action the tool can provide a corresponding JSP that will be displayed to the user to gather some input, if any is required, to initiate an export task, and to display any output (logging/warnings/error message).

The tool basically needs to do three tasks; their implementation can be located in separate modules (JAR files).

EXPORT FROM VIGNETTE

Using SQL queries, access data from corresponding tables in the Vignette database. For this task, a configuration file can be used to specify Vignette content tables and fields per content type, so that SQL can be built dynamically, not hard-coded; or it can specify complete SQL queries.

The tool can provide some basic export tasks (content export, categories export); the list can be extended as needed.

TRANSFORM

Transform extracted data into appropriate intermediate (Transformation) objects. Transformation objects have fields corresponding to a given content type (content template) fields, workflow status, and content profile (categories and keywords), and provide set/get methods to them. These objects may have to be created manually for each content type and provided in a separate JAR file that's shared by the export-import process.

Transformation objects are used by the Export task. While accessing the Vignette database, data get put into transformation objects and kept in a Vector object that's passed into

one of the Import task methods.

A Hashtable object can be used as a Transformation object as well. With this approach Hashtable object keys will correspond to either:

- The WebCM content template fields for content migration
- The site structure parent-child relationships for site structure migration

Using Hashtable objects will allow the migration tool to be configured. Only one set of configurable variables will need to be used for content template fields and Hashtable keys.

IMPORT INTO WEBCM

Finally, using the WebCM published API, create corresponding changes in the WebCM repository. This module will provide different methods to do appropriate actions (site structure, content migration, etc.).

- Based on a corresponding workflow status retrieved from the Vignette database and stored in the transformation object, a content item will be placed into an appropriate workflow stage.
- Based on the content profile (categories and keywords) retrieved from the Vignette database and stored in the transformation object, a content item's corresponding profile properties will be set accordingly.

This module can use a configuration file to specify WebCM object names: workflow with stages, authoring template(s), presentation template(s), content field names that will need to be accessed during the import process.

These objects have to be created manually in WebCM before running the migration process, since the published API doesn't let you create them, only access them.

Figures 1 and 2 show the process flows for site framework and content migration.

The following analysis steps provide additional information for the migration process:

- Analyzing the Vignette Content Delivery Application (CDA) lets you identify content delivery components and their functionality, which will be re-created in WebCM as various generic library components and presentation templates. This is a manual process that can still re-use existing presentation-level HTML to speed up the process, unless the Web site needs a new look-and-feel.

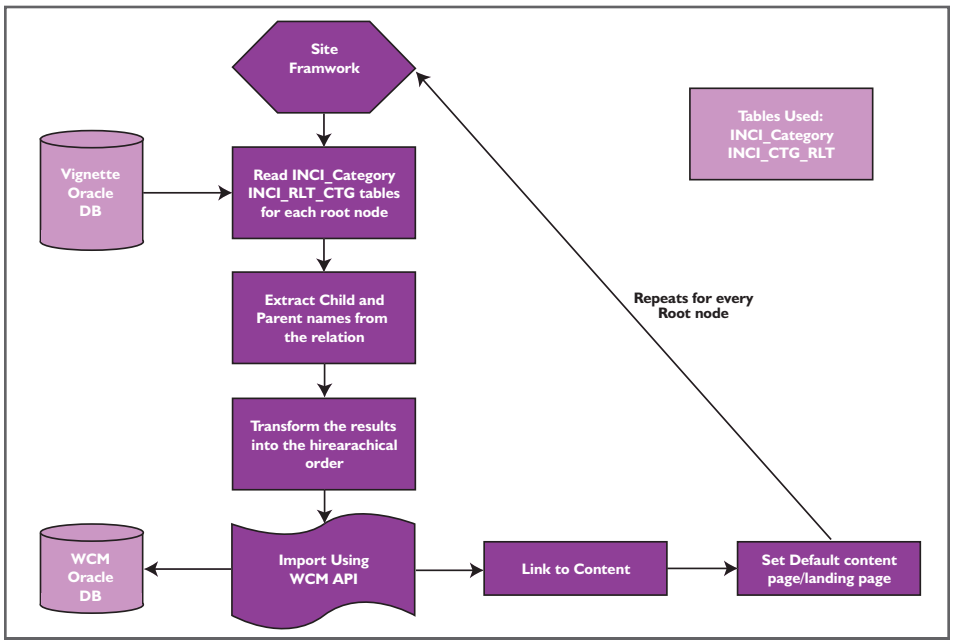


FIG 1: PROCESS FLOW FOR THE SITE FRAMEWORK

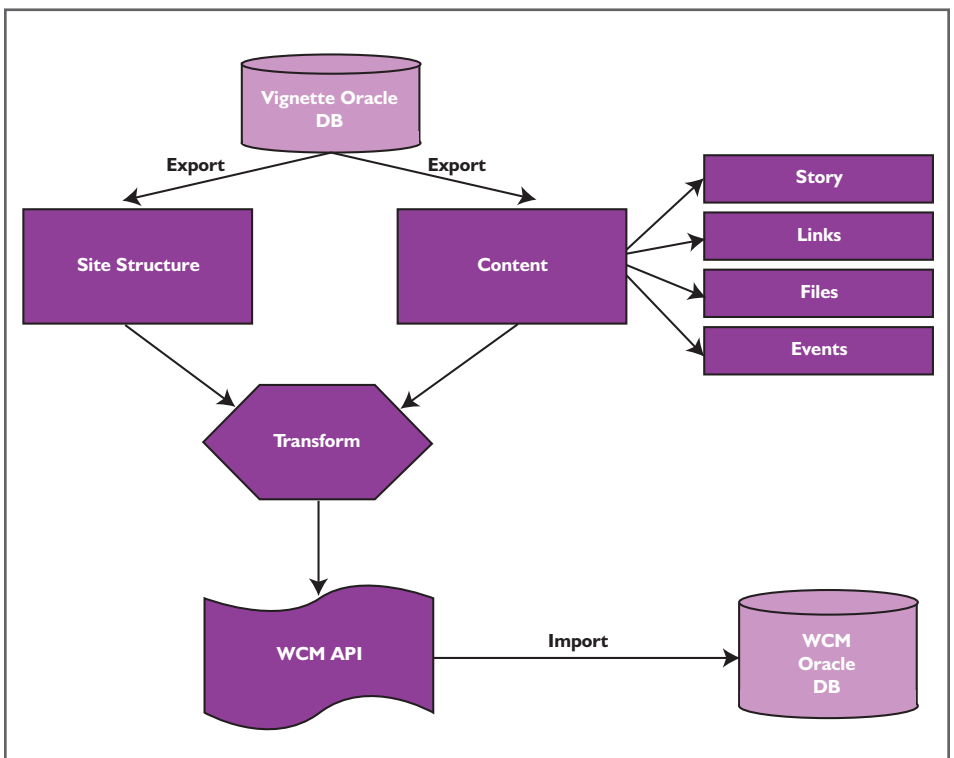


FIG 2: PROCESS FLOW FOR AUTOMATIC CONTENT MIGRATION

- Sometimes as a result of tailoring Vignette applications to the needs of a particular client, custom tags can be introduced; for example, Vignette tags that build links to other content items. Since each custom Vignette tag renders certain markup, which is included in a Web page, the solution for each custom tag can be easily built based on using corresponding WebCM tags, i.e., *AptrixCmpnt* with *start* and *end* attributes.

To build an appropriate WebCM security model, corresponding users and groups have to be created in the WebSphere Portal WMM. If there's an LDAP server in place, the WebSphere Portal WMM can be configured with LDAP.

Integration with another search engine (for example, Autonomy Server V.4) can be easily implemented using the JSP component's references on WebCM presentation templates.

Caching Options

Regardless of the migration process, you'll need a caching strategy that can be tailored to a client's Web site requirements, which may vary widely and sometimes even require additional implementation using published (or even unpublished) functionality in the WebCM API.

To implement a successful caching strategy, the first step is an **analysis** of the site to determine the nature of the content to be cached. Things to consider include:

- Whether the site contains static or dynamic data, or both
- Whether "Connect" tags or URL requests in the Presentation Templates or the Component Designs are used
- Whether data from external sources such as databases will be displayed

Once the site and its content has been analyzed, and it's been determined that caching is a realistic option, then the type or types of caching to be implemented need to be finalized. This includes determining:

- The server's default cache type
- The server's default caching and expiration settings
- What custom caching and expiration methods to use

WebCM provides different caching options as well as pre-rendering feature. For the purpose of this article:

- Pre-rendering refers to a function in WebCM to collate design and content components and then store them on a physical disk as complete HTML files.
- Caching refers to the ability to store content (after it's been rendered from various components) in either memory and/or on the hard disk. The cache is stored when a browser requests a page. Both WebCM and DynaCache (part of WebSphere Application Server) offer both memory/disk caching and expiration management.

WEBCM PRE-RENDERING

The pre-rendering engine renders an entire site (based on the site framework selected) and can't be broken down into partial pre-rendering. It's done by enabling the <Cacher class=….> module in *connect.cfg*. The resulting HTML files can be stored on another device and used by other Web servers.

Pre-rendering doesn't have expiration management and a pre-rendered site is replaced when the pre-rendering engine is triggered. It's suitable for a small to medium-size Web site that contains static information without personalization.

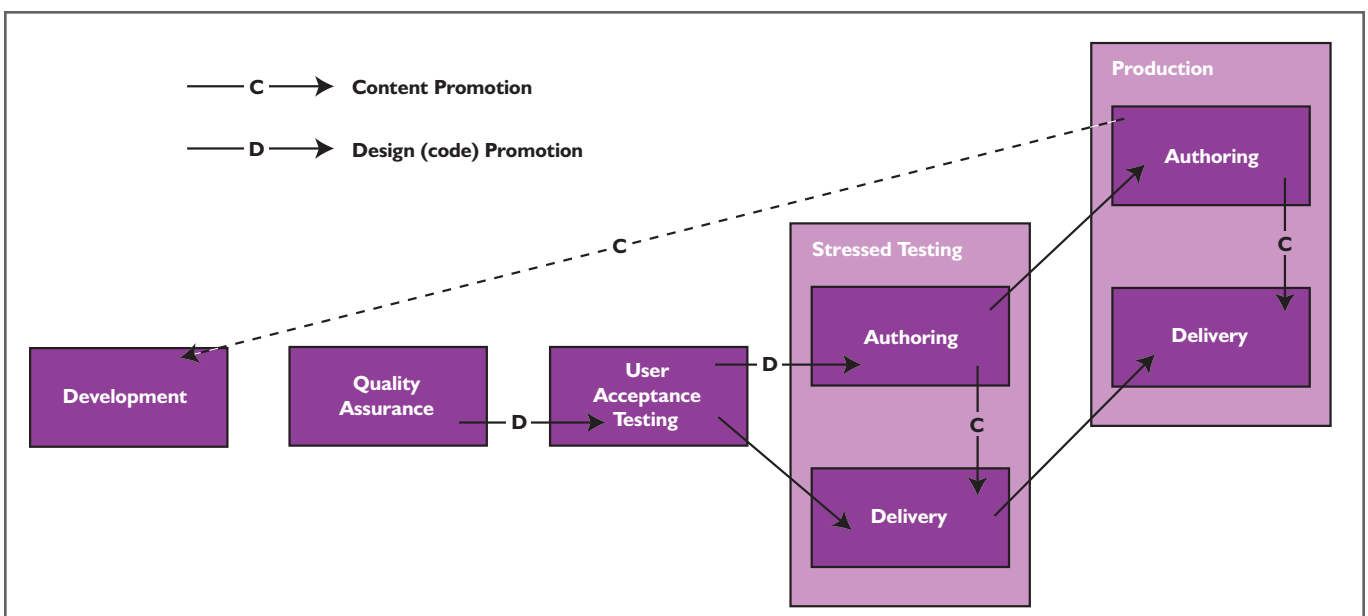


FIG 3: CODE DEPLOYMENT OVERVIEW

www.sys-con.com/edge2005

web services EDGE conference&expo

FALL SERIES

[Engage and Explore] the technologies, solutions and applications that are driving today's **Web services** initiatives and strategies...

CALL FOR PAPERS NOW OPEN!*

Coming to a City Near You ►

Web Services Edge
Fall Conference Series

3 Dynamic Conference Programs
Targeting Major Industry Markets

20+ seminars within 5 tracks will address the hottest topics & issues:

- ► Web Services: The Benefits and Challenges
- ► Web Services Security
- ► SOA (Service-Oriented Architecture) and ESB (Enterprise Service Bus) Strategies
- ► Interoperability, Incremental Integration, & Open Source
- ► The Management Process in Developing a Web Services Strategy

Program Features:

- ► Keynotes
- ► Tutorials
- ► Panel Discussions

Why Attend:

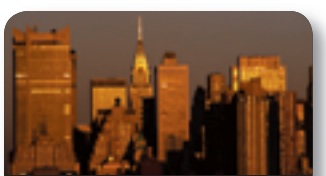
- ► Improve the return on your technology investment
- ► Develop & sharpen your strategy and identify key action steps
- ► Find new ways to reach and impress customers with Web services
- ► Maximize the power of your enterprise
- ► Protect your business from security threats
- ► Assess Web services as a viable option

Attention Exhibitors:

- ► An Exhibit-Forum will display leading Web services products, services, and solutions

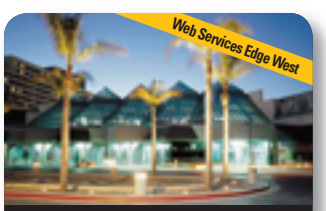
The Westin Washington, D.C.
Washington, D.C.
September 7-8, 2005

1



Helmsley Hotel
New York, NY
September 19-20, 2005

2



The Westin Santa Clara Convention Center
Santa Clara, CA
October 24-25, 2005

3

Register Today! www.SYS-CON.com/Edge2005

Sponsored by

WebServices JOURNAL | XML JOURNAL | .NET DEVELOPER'S JOURNAL | eclipse developer's journal | WebSphere JOURNAL | information STORAGE+SECURITY journal | wldj | JDJ | LINUXWORLD MAGAZINE | MX developer's journal | asp.netPRO | SD Times | CoDe | Software Test & Performance

*Call for Papers email: grisha@sys-con.com

For Exhibit and Sponsorship Information ► Call 201 802-3066

WEBCM CACHING

WebCM caching will also do the pre-rendering function. It also offers a caching functionality in terms of memory/disk caching, basic/component/custom caching, and cache expiration options. These functions are set up through the cache configuration in *connect.cfg*. It must use WebCM as a Web server since the caching file system isn't in true HTML format.

Caching options (basic or custom) can be used in conjunction with memory/hard disk caching to improve performance. However, WebCM memory/hard disk caching can only be used when WebCM is used as the Web server.

There are two caching options:

- ***Basic caching:*** This is a default setting for WebCM and is almost the same as pre-rendering in the sense that the entire site is rendered. The main difference lies in the mechanics of how the physical files are created. Basic caching (via the Cacher module) renders a page at a time. It also provides basic cache expiration options. Basic caching offers a more flexible solution for a larger Web site for more current content because of the expiration management. However, it can't provide caching for dynamic, personalized, or secured content.
- ***Advanced caching:*** Advanced caching that provides options to selectively cache content by session, users, etc. provides the same output as the basic caching option. In addition, 'connect' tags can be used to selectively cache (or un-cache) components in a Web page. This option is useful for a Web site that has largely static content with some dynamic components.

Note: Connect tags can only be processed when the default caching mode is enabled and the WebCM Web server processes the tag.

WEBSPHERE DYNACACHE

DynaCache can be configured to work with the WebCM servlet to provide a cached page when it's first requested. There's a configuration file *cachespec.xml* file to be added to the *[was_root]\installedApps\wcm.ear\ilwwcm.war\WEB-INF*. In this file, parameters have to be specified on site structure, expiration, etc. Pages from DynaCache can be stored on a Web server, but the WAS plug-in has to be configured to work with this setup.

It's also possible to configure DynaCache to cache content at the Web server. This requires that Edge Side Includes (ESI) components (or SunOne's equivalent driver) be installed on the Web server.

Deployment Strategy

During deployment, the design components will be promoted (or transferred) from the development environment to the Quality Assurance (QA), User Acceptance Testing (UAT), and Production environments.

All design components (or code), such as WebCM content templates, WebCM presentational templates, JSPs, and Portal themes are produced in the development environment. Due to the differences in capabilities, a number of code deployments will be used for code promotion.

Figure 3 provides an overview of how the codes (regardless of their location) can be deployed.

For WebCM, syndication can be the vehicle for code deployment. However, the syndication can also be used to promote content as well as design components (codes). However, detailed deployment planning and operational procedures must be set up to ensure content integrity and security.

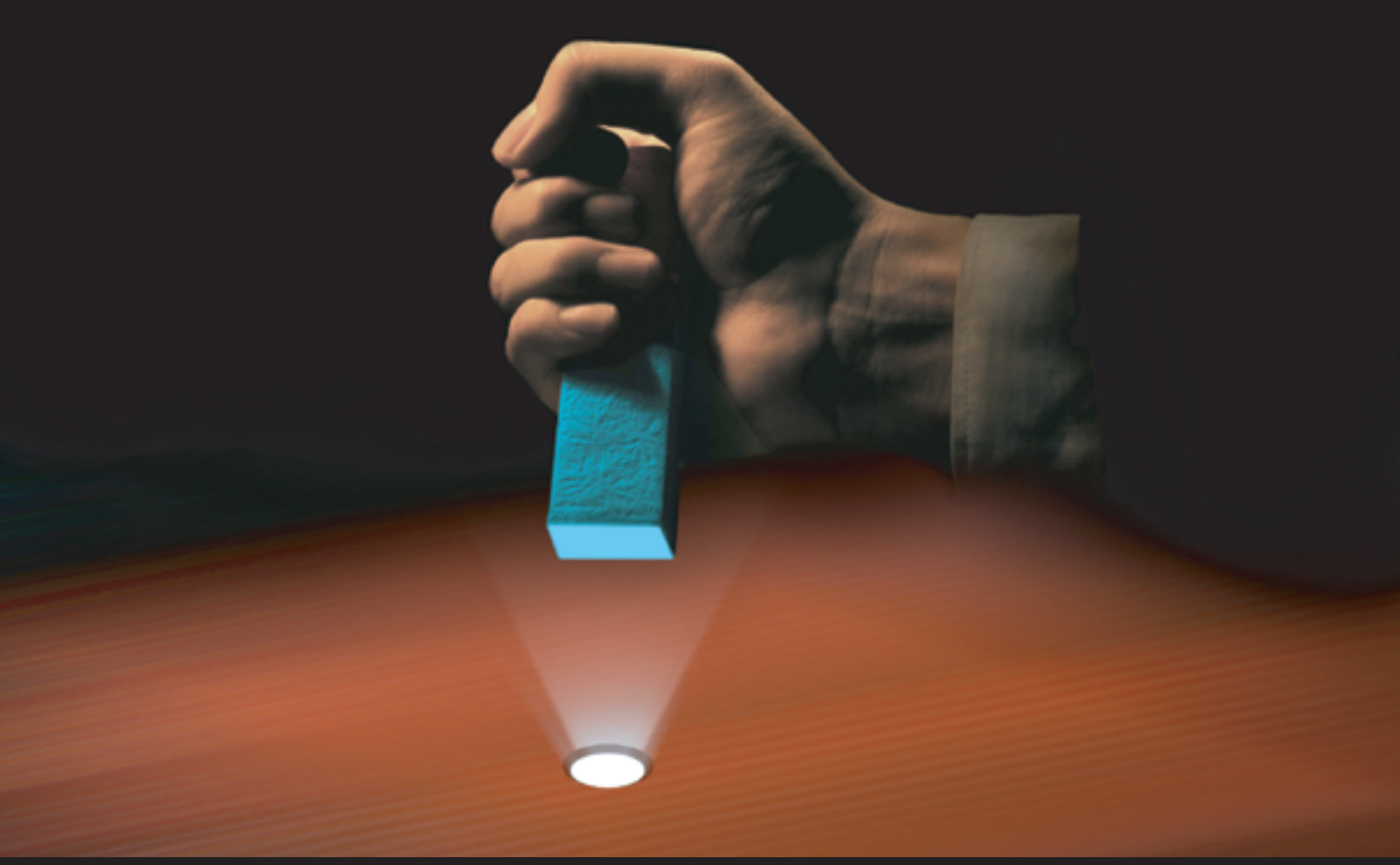
An alternative is to create a custom export/import function for the design components to provide code deployment instead. However, this option uses unpublished features of the API to access the components for both the export and import processes.

Another possible option is the integration with WebSphere Process Choreographer. It also uses part of the unpublished API to create custom workflow actions. The custom workflow action will have a configurable URL to provide content ID, the workflow stage of the content, etc. to a JSP. It's the JSP that will interface with the Process Choreographer using the published API.

"Analyzing the Vignette workflows provides the information needed to recreate corresponding workflows in WebCM"

Conclusion

With a promising future for WebCM including an integrated development environment (IDE), repository syndication, and task-based workflows together with IBM's commitment to delivering world-class portal, workplace, and content management solutions, many organizations are making the transition from Vignette to WebCM. By doing so you not only gain a content solution but you gain a completely robust and scalable architecture.



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

Sophisticated J2EE platform is different and powerful

Understanding WAS for z/OS

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 corss 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.

WebSphere Application Server for z/OS is a comprehensive and sophisticated J2EE application server platform. It integrates with the IBM's zSeries portfolio of hardware and z/OS software assets and exploits the qualities of service of the platform such as close proximity to data, intense scalability, continuous availability, failover support, and stalwart security.

Most developers working with WebSphere for z/OS today don't have IBM zSeries server experience. They're good at J2EE and are more familiar with WebSphere on distributed platforms. This article intends to help folks understand the server architecture and workload management differences between WebSphere for z/OS and distributed platforms. We'll also discuss how to leverage the unique features in WebSphere for z/OS.

WebSphere for z/OS used to be a totally different product from WebSphere for distributed platforms before WebSphere V5. IBM has been trying hard to make them share the same code base since WebSphere V5. The products look quite similar now, but there are major differences in server architecture, installation, security, workload management, performance tuning, and problem determination. Many of the areas where they deviate let the z/OS version take advantage of qualities of service of the zSeries platform.

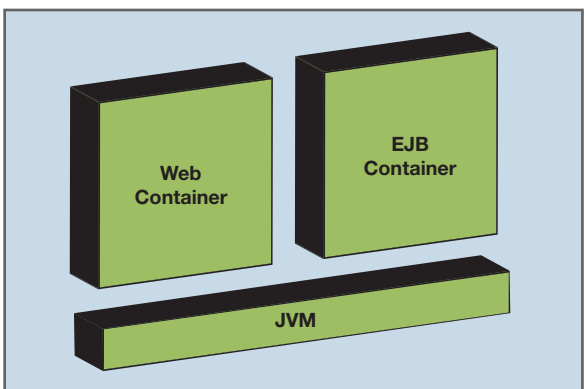


FIG 1: WEBSPHERE FOR DISTRIBUTED PLATFORM SERVER ARCHITECTURE

Defining Terms

Before we start, I would like to introduce some basic z/OS terminology.

Address Space - An address space is the area of contiguous virtual addresses that z/OS assigns to a user (or separately running program) for executing instructions and storing data. It's equivalent to a process on distributed platforms.

TCB - TCB stands for Task Control Block. It represents a task executing in an address space at any one time. A Java thread is implemented as a TCB on z/OS.

WLM - WLM is the z/OS Workload Manager.

Enclave - An enclave is a transaction or unit of work that can span multiple dispatchable units in one or more address spaces. It's equivalent to a unit of work that can cross multiple processes on a distributed platform.

Service Class - Service Class is a z/OS WLM term. It's used to group similar kinds of work with same performance goal.

WebSphere for z/OS Server Architecture

WebSphere for z/OS is equivalent to the WebSphere Network Deployment (ND) on distributed platforms. The ND architectures and concepts are same on all the platforms. The biggest difference is the server architecture. On most distributed platforms, the server is a managed process with a JVM instance. It has a J2EE Web container and an EJB container. Figure 1 depicts a server for distributed platform.

On z/OS the server is split into different address spaces. Figure 2 shows the server architecture.

There are two kinds of address spaces in the WebSphere for z/OS server: the controller and the servant. A controller region runs system-authorized programs and manages tasks such as communications for the server. A servant region is the address space equivalent to the server on a distributed platform

TANGOSOL™

A Cure for Downtime.

Eliminate the single points of failure, add fault tolerance and enable transparent failover for your J2EE applications.

TANGOSOL COHERENCE™ has no single points of failure, ensuring high availability and reliability for your mission-critical J2EE applications. Application state, HTTP sessions and data caches are never at risk. Applications using Coherence keep it up without interruption, even when servers go down.

Try before you buy.
Download a free evaluation license at

www.tangosol.com

Coherence™ is pure Java software that reliably and cost-effectively scales J2EE applications across any number of servers in a cluster. Coherence provides reliable coordinated access to in-memory data that enables safe data caching and user session management for applications in a clustered environment.

Sales	+1.617.623.5782 or sales@tangosol.com
Support	support@tangosol.com or http://www.tangosol.net
Pricing	US$1,995 to US$4,995 per Coherence production CPU Development licenses are offered free of charge.

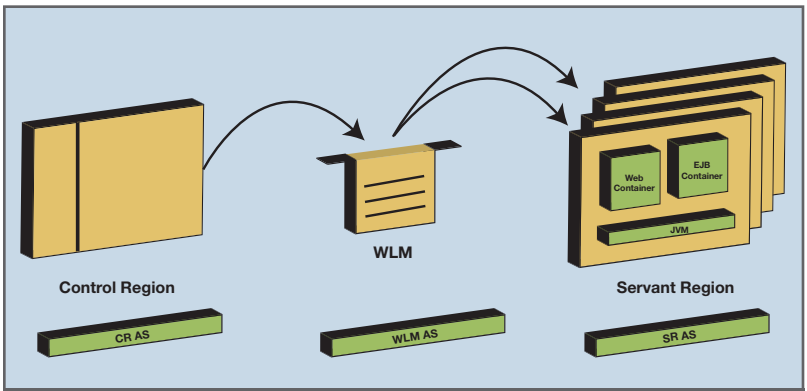


FIG 2: WEBSPHERE FOR Z/OS SERVER ARCHITECTURE

that has a J2EE web container and an EJB container. WebSphere for z/OS server has only one control region. However it can have multiple servant regions. WebSphere for z/OS provides four types of servant regions: Normal, CPUBOUND, IOBOUND, and LONGWAIT. Each type of servant region was designed to handle different types of workloads.

Client requests (including HTTP, MDB, IIOP, etc.) arrive at the control region, which has multiple types of listeners. After receiving the requests, it works with the z/OS workload manager (zWLM) to dispatch the work to the servant regions. All the servant regions in a WebSphere for z/OS server are identical and host the same J2EE application. The granularity of the J2EE application deployment is at the server level.

The WLM Difference

zWLM is totally different from the WLM in WebSphere for distributed platform. WLM on distributed platforms is just a WebSphere service for weighting the priority of cluster members whereas zWLM is part of the z/OS. It ensures that long-running tasks don't monopolize system resources, interactive tasks execute consistent and guaranteed response times, and resources are allocated according to business goals. User requests are classified, prioritized, and allocated to system resources based on operational policies that are bound to business goals.

zWLM lets organizations specify a set of work classification rules that identify all incoming requests. These rules can be established using a variety of business-oriented parameters that include transaction names or types, specific user names or user types and time of day. The user requests are classified by the classification rules. Each work unit is assigned a service class that represents a group of work with similar performance requirements expressed in terms of a service-level goal and relative business importance. For example, a user request is classified to a service class called CBFAST, which has as a performance goal that 80% of the work should be finished in 0.5 second. The importance of the service class is 3 (a number with a value of 1 to 5 specifies the priority of the work). Besides a factor of response time, service level goals can be expressed as execution velocity or discretionary.

zWLM manages resources to meet the performance goals. If all performance goals can't be met, zWLM favors transactions and address spaces with the highest importance level. (If you want to know more about z/OS WLM, please refer to Resource 1.)

How Does the Server Work?

Let's take a look at how the WebSphere for z/OS server and zWLM work together.

When the control region starts, it connects to zWLM as a work manager. It defines the application environment that provides the start procedure name used to start the servant region. Whenever a client request arrives, the control region classifies the work request, associates it with a service class (defined in the zWLM policy), and creates an enclave for the request. Then the control region inserts the work request into the zWLM queue associated with the specified service class. At this point the application environment and the enclave token are sent to zWLM as input.

When the first request is queued to an application environment, if zWLM detects that there's no active servant region exists it will automatically start one. As workloads fluctuate, zWLM adjusts the number of servant region address spaces to reach the defined performance goal.

When a servant region starts up, it connects to the zWLM as well. It selects a work request from the specific zWLM queue and assigns it to a TCB (Java thread) for processing. At this point the TCB joins the enclave created by the control region. The enclave represents the client request

"WebSphere for z/OS is challenging to deal with because of its deep integration with z/OS"

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.

processing start from the control region to the servant region and other subsystems such as DB2. Even the business logic written in a J2EE application only runs in the servant region. Since the enclave has been associated with a specific service class that's tied to a performance goal, zWLM manages the resources cross address spaces to meet the request process performance goal. Once the process finishes, the enclave is deleted.

As you might guess all the client requests sent to a J2EE application are classified before they can be processed. The business logic in the application occurs in an enclave. zWLM doesn't treat workloads in WebSphere in a special way.

WebSphere for z/OS V5.1 and higher can support workload classification by URLs, MDB listener ports, and EJB methods. The WebSphere administrator can set up an XML mapping file to map requests to multiple transaction classes. The transaction class-to-service class mapping rule is defined by the zWLM policy. WebSphere for z/OS and zWLM provide a more flexible way to control your application workload than on distributed platforms.

Application Development Impact

The WebSphere for z/OS server has a unique architecture that's deeply integrated with the z/OS platform. The good news for application development staff is that the architecture doesn't change the J2EE programming model. The application runtime control granularity provided by WebSphere for z/OS can be leveraged to simplify your application design.

Let's walk through an example here.

Image a fictional banking application service provider. It's developing its next-generation Lending process system on WebSphere for z/OS. The Lending process system provides all the functions a loan officer needs to process a loan application and maintain it.

One of the application's functions is to let a loan manager run a campaign report. Since running a campaign report is very time consuming, an architectural decision is made to implement it in an asynchronous process using a Message Driven Bean (MDB). The report requester won't wait for the report online. When the report is ready, the requester will be notified by e-mail. Then he can come back to check and view the report. If the application is to run on a distributed platform, it should have certain kinds of workload management logic built in to prevent the campaign report process from monopolizing application server resources and impacting online transaction processing and system throughput. The alternative is to use a dedicated server with a dedicated application for the campaign report process, but it makes the environment more complicated.

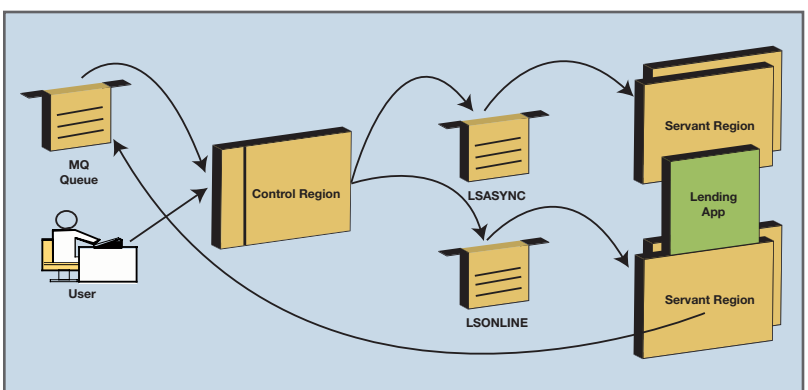


FIG 3: SOLUTION IN WEBSPHERE FOR Z/OS

WebSphere for z/OS provides a better solution for this kind of problem. As shown in Figure 3 the problem is solved without implementing any workload management logic in the application itself.

Two service classes are defined for the Lending system:

- LSONLINE – 95% of the transactions finish in 0.5 seconds with an importance number of 2
- LSASYNC – 80% of the transactions finish in 2 hours with an importance number of 3

The online transactions are classified to service class LSONLINE based on URLs. Campaign report requests are classified to service class LSASYNC based on MDB listener ports. Requests with different service class will be processed by different address space. So there's no chance of the campaign report process fighting the online transaction process for threads. zWLM manages computing resource to meet the performance goals set up for the application. It always favors the online transaction process whenever a resource shortage occurs.

The same technique can also be used to meet different customers' service level requirements. The fictional service provider may need to provide a dedicated server for major customers for isolation purposes. As shown in Figure 4, requests coming from different customers are classified to different service classes (Service Class I and II) with their own performance goals. The system provides different levels of service to different customers. The highest importance number (1) can be given to the most important business customer so that the service level agreement is guaranteed. Further more, you can define multiple zWLM policies and active them in different time periods in your application. So you can run your application in an on-demand way.

Application Performance Impact

I'm going to break the bad news here. The WebSphere for z/OS server architecture makes application performance tuning more complicate than for distributed platforms. Application performance tuning is too big a topic. A separate article is needed to cover it. However, it's helpful to point out some basic zWLM-related concepts.

The WebSphere for z/OS server architecture determines that workloads of the J2EE applications running in WebSphere for z/OS are classified to a service class before they get processed. Unfortunately many application developers ignore this fact. I've frequently seen application development teams do load testing and performance tuning without knowing which service classes the application being tested was running under. And people often ask why an application run very well on a distributed platform but was bad when it runs WebSphere for z/OS. When asked about the service classes, people usually have no idea. If your application is running under an inappropriate service class, you'll never get the performance desired no matter how you optimize your application code. You have to understand that if zWLM can meet the performance goals by allocating a certain amount of service units, it won't try to allocate more resource to make the application always exceed its performance goal. That only happens when the whole machine is dedicated to your application. The basic rule is: if you don't ask for it you'll never get it. So try to figure out how many service classes you need and make the performance goals aggressive enough. Of course they should have enough priority.

With applications running on WebSphere for z/OS share resources with other subsystems on z/OS, zWLM treats workloads in WebSphere for z/OS the same as that of other subsystems such as CICS and DB2. The performance goal and priority for your workload should be reasonable. If the performance goal is too aggressive, there are two possible results. One is that zWLM will try very hard to meet the goal if your service class has the highest priority by sacrificing some other subsystems' performance. The other result is that zWLM will just give up because the goal is too hard to meet. Of course, in the latter case the application's performance gets worse.

At first you may feel that WLM is hard to deal with. Try to understand it better. That's key to making your applications run better on z/OS. You may need to be familiar with RMF reports. It tells you how well the system is doing its work.

WebSphere for z/OS is also a CPU-intensive subsystem on z/OS. If the CPU coefficient is high and the service class is aggressive, performance will be optimal.

Conclusion

WebSphere for z/OS is challenging to deal with because of its deep integration with z/OS. The unique server architecture combined with zWLM does produce a better way to control application workload to meet business goals. It makes it possible to develop enterprise applications with better quality of service (QoS).

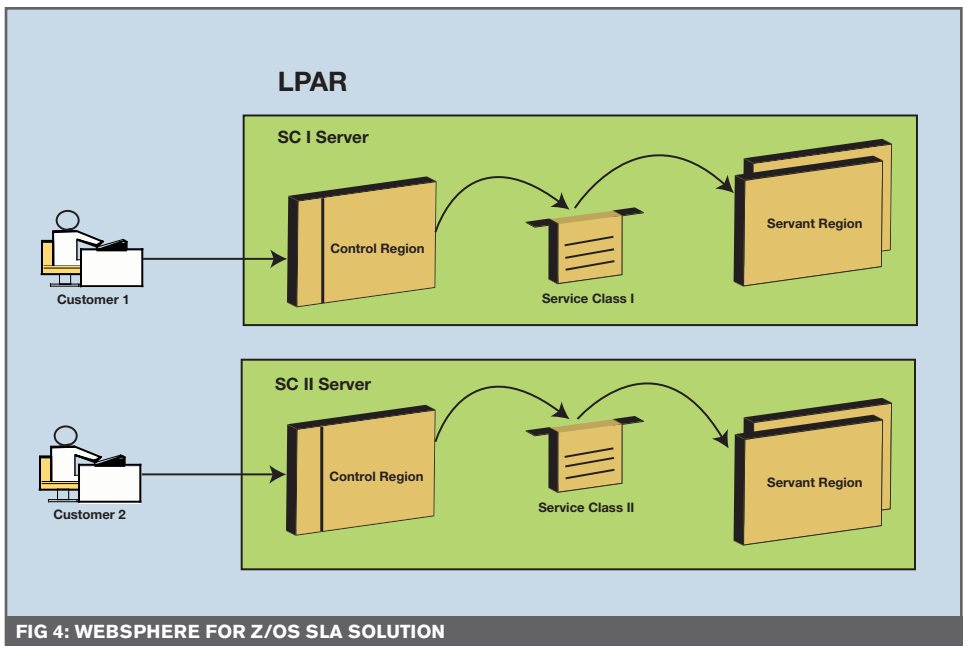


FIG 4: WEBSPHERE FOR Z/OS SLA SOLUTION

Subsequent articles will cover other interesting topics such as topology, scalability and high availability, security, performance tuning, connectivity, and trouble shooting.

Resources

- WebSphere product library : http://www-360.ibm.com/software/webservers/appserv/was/library
- IBM product manual: z/OS MVS planning: Workload Management SA22-7602

For more information on how to set up workload classification rules, control the servant region workload profile, and control the mini and max servant region number you can refer to these articles.
http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100449
http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD101152

From Business Modeling to Web Services Implementation

Part 2 – Implementing a Web Service for a Business Process

BY TILAK MITRA

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

First published on developerWorks WebSphere at www.ibm.com/developerWorks

Part 1 of this article illustrated how to use WBI Modeler 5.1 to model a business process. We used the example of a simple "Web Order" business process and showed how it can be modeled in a BPEL-compliant mode. The artifacts created in Part 1 were exported into the file system for use as the input to the business process implementation – the topic of this article.

In this article we're going to explain how to add business logic to each individual task that comprise the business process and subsequently expose the entire business process as a Web Service. As part of the end-to-end process, we're going create a Web Service proxy that can be used by a service consumer to test the entire business process.

It's strongly advised that you complete the business process modeling exercise in Part 1 (*WebSphere Journal*, Volume 4, issue 4) before going through the hands-on exercise illustrated in this section since it's the prerequisite for this article.

Setting up the Project

This section illustrates the creation of a new *Service Project* in WebSphere Studio Application Developer Integration Edition (WSADIE) and imports the output artifacts of the business modeling exercise (in Part 1).

1. Open WSADIE and create a new workspace. Switch to or open the Business Integration Perspective. We're going to create a new Service Project. A service project contains the files that collectively represent an enterprise service or process. Right click on the Service Project folder and create a new Service Project (***New->Service Project***). Enter the name of the new project as shown in Figure 1 and click ***Finish***.
2. The contents of the process (exported) from the modeler (in Part 1) need to be imported into the newly created *CricketOnline* project. Right click on the newly created project and click the ***Import*** menuitem. In the dialog box that appears, highlight ***File System*** and click ***Next***. Browse to the directory where the artifacts from the business process modeling exercise issue were exported. Check the folder contents as shown in the Figure 2 and click ***Finish***. This imports the business process into our newly created project.
3. The project contents looks like the Figure 3. Notice how each of the business items created during process modeling has a Java class representation in the *Businessitems* folder. The *BusinessItems.xsd* is an XML

FIG 1: NEW SERVICE PROJECT

FIG 2: IMPORT BPEL CONTENTS

representation of the business items.

4. The most important artifacts are *WebOrder.bpel* and *WebOrderInterface.wsdl*. They are the subjects of the next two sections.

Anatomy of WebOrder BPEL

Each process that's modeled during the business process modeling effort manifests itself as a BPEL file. The format of the file is <process_name>.bpel where *process_name* is the name of the process that was modeled. In our scenario, we only modeled one process *WebOrder* and hence we have a file *WebOrder.bpel* in the *ProcessCatalog.WebOrder* folder. Double-clicking the bpel file opens it up in the process editor. (See figure 4.)

The process flow looks like the one we modeled in Part 1 except for the few a extra elements that appear in its representation in WSADIE. *WebOrder Receive* represents the input elements that trigger the process initiation. *WebOrder Reply* represents the output of the process. Between the individual tasks there are *Assign* elements. There's also a list of variables that are used in the business process.

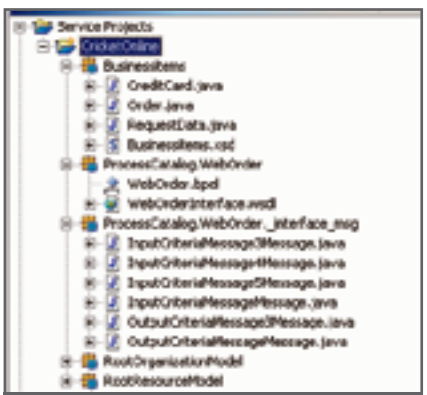


FIG 3: IMPORTED PROJECT CONTENTS

FIG 4: WEBORDER BPEL DIAGRAM

Variables store the messages that are used by a business process. The message within the variable is defined using a WSDL message type that's made up of one or more message parts.

An *Assign* element is used to map the output of a task to the input of a subsequent task. The *Assign* activity is used to copy values from one variable to another. The *Assign* activities in our *WebOrder* business process have already been set and they don't need to be changed unless the process flow has to be altered.

Figure 5 shows how an *Assign* activity copies the output of one task to the input of a subsequent task by copying the message parts of variables. (The *Assign* activity in the Figure 5 copies the output of the Price Order task to the input of the *Save Order* task. Clicking on an *Assign* activity displays its properties in the editor pane).

Notice how the *Order* message part of *PriceOrderOutputCriteriaVariable* is mapped to the *Order* message part of *SaveOrderInputCriteriavariable*. Examining each of the *Assign* activities helps in understanding how the various tasks are linked together in a business process.

Each task in the process is associated with a partner link. Partner links identify the parties that interact with the business process. A partner link defines operations that can be implemented internally or may even map to operations in external Web Services that may be invoked to fulfill the task. Clicking on a partner link shows the task it's associated with.

Our business process has all the necessary *Variables* and *Assign* activities defined and mapped.

Anatomy of WebOrderInterface WSDL

Each business process that can be created in the business modeling step (Part 1) has a corresponding interface definition file (in WSDL format). The WSDL contains a list of *Messages* and *Port Types* among other elements. Figure 6 denotes the contents of *WebOrderInterface.wsdl*.

The *Messages* are used to construct the *Variables* that are used to copy values between inputs and outputs of individual tasks as the process flows from one task to another.

A port type is a named set of abstract operations and messages involved.

Each operation refers to an input message and output messages.

Figure 7 illustrates how the variables of *PriceOrderPT* are composed of *messages* that encapsulate the input and output parameters.

Each port type representing a task in our business process needs to be implemented to satisfy the business logic that it's responsible for carrying out. In the next section we'll take the imported artifacts and generate and create implementation code that provides the overall implementation of our business process.

Process Implementation

There are a few steps that need to be executed in sequence to implement our business process.

Implementing the Map

The first step is to implement the *Map* that links *Submit Order* to *Price Order* and *Transact Payment*.

FIG 5: ASSIGN ACTIVITY BETWEEN PRICEORDER AND SAVEORDER

FIG 6: WEBORDERINTERFACE WSDL CONTENT TREE

There are often situations where the output of a task is a complex datatype. This task may be connected to other tasks that accept as input, data, that can be contained inside the complex datatype that's the output of a previous task. A *Map* is used to map parts of a complex datatype to the input of a subsequent task. In our example, the output of *Submit Order* task is a complex datatype *RequestData* that contains *Order* and *CreditCard* as its contained datatypes. The subsequent tasks *Price Order* and *Transact Payment* requires *Order* and *CreditCard* as their respective inputs. A *Map* in our business process is used just for this purpose.

Clicking on the *Map* element on our process opens the attributes and characteristics of the *Map* in the lower half of the process editor. Clicking on the 'Implementation' link renders a Java scratchpad editor in which the logic (for copying variable values) has to be implemented. (See Figure 8.)

We use a snippet of Java code to implement this logic. It extracts the *Order* and *CreditCard* object instances from the *RequestData* output datatype (of *Submit Order*) and maps it to the output of the *Map* variable. The sample Java snippet is shown in Figure 9.

Note: You'd have to import the *RequestData*, *Order*, and *CreditCard* datatypes to compile that code snippet.

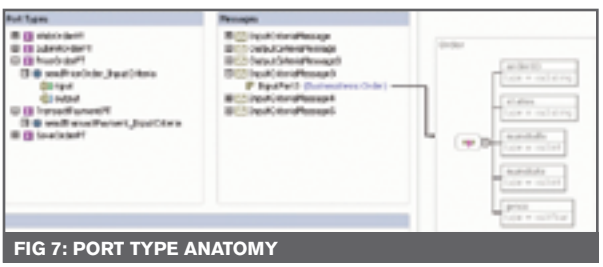


FIG 7: PORT TYPE ANATOMY

FIG 8: MAP PANE

FIG 9: IMPLEMENTATION FOR THE MAP

Creating Java Skeletons & Implementations for the Individual Tasks

Each port type needs to have an implementation for the business logic that the task is supposed to perform. We're going to use Java to implement the business logic. Several steps are involved in this process. A Java skeleton has to be created for a given port type. Business logic then has to be added to the skeleton Java class.

The steps below will illustrate the process for one port type. We'll use *PriceOrderPT* for our example.

1. Right click on the *PriceOrderPT* port type in the *WebOrderInterface.wsdl* file and select 'Build from Service' (***PriceOrderPT->New->Build from Service...***).
2. We'll create a 'Java Service Skeleton.' In the dialog box that comes up, click 'Java Service Skeleton.' Click **Next**. (See Figure 10.)
3. In the next dialog box, (See Figure 11) the attributes for the Java skeleton have to be set properly. The value for the 'WSDL file:' attribute will be the *WebOrderInterface.wsdl* file. Browse in the *WebOrder.ProcessCatalog* folder under the *CricketOnline* project to select the WSDL file. The value of 'Port type:' needs to be *PriceOrderPT*. Select it from the dropdown for the value of the 'Port type name:.' All other values are kept as the default. Click ***Next***.
4. We'd like to create the Java skeleton files in a separate package. Use the values as shown in Figure 12.
5. Click ***Finish.***

This completes the process of creating the Java skeleton for the *PriceOrderPT* port type. The same process (as illustrated above) has to be followed for the other three port types (*SaveOrderPT*, *SubmitOrderPT* and *TransactPaymentPT*). Care should be taken to associate the proper port type names, i.e.:

1. *SaveOrderPT* while creating the Java skeleton for *SaveOrderPT* port type.

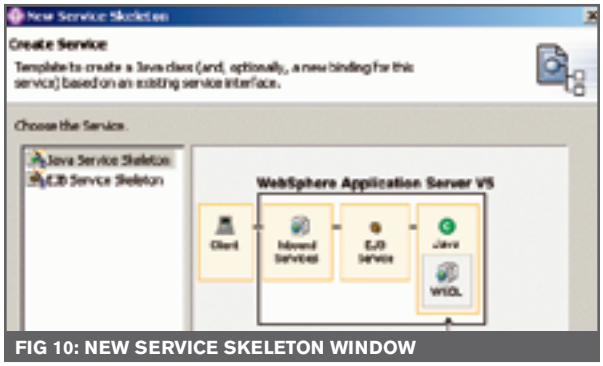


FIG 10: NEW SERVICE SKELETON WINDOW

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

2. *SubmitOrderPT* while creating the Java skeleton for *SubmitOrderPT* port type.
3. *TransactPaymentPT* while creating the Java skeleton for *TransactPaymentPT* port type.

Once finished, the folder structure should look like Figure 13 (with the newly created artifacts).

Note: The skeleton creation process shouldn't be exercised for the *WebOrderPT* port type. This is because we don't need a Java implementation of the process, only its individual tasks need an implementation.


FIG 11: JAVA SERVICE SKELETON DIALOG BOX

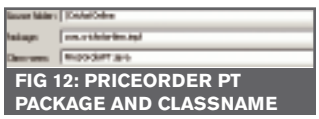


FIG 12: PRICEORDER PT PACKAGE AND CLASSNAME

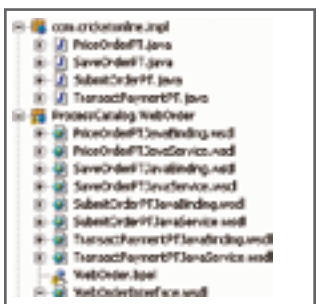


FIG 13: PRODUCT OF JAVA SERVICE SKELETON CREATION


FIG 14: CODE SNIPPET FOR SENDPRICEORDER_INPUT CRITERIA

The exercise that's left is to implement the business logic in the respective Java classes. The business logic for the *sendPriceOrder_InputCriteria* method in *PriceOrderPT.java* is shown in Figure 14.

For a list of all the other Java implementations, see this article on www.websphere.sys-con.

Process Deployment

With the *Map* element defined and the implementation of the individual tasks complete, our process can now be deployed. The deployment of a business process involves a few steps. The deployed code has to be created for the business process as well as for the referenced partners (the port types that represent the individual tasks).

1. Right click on the *WebOrder.bpel* and click 'Generate Deploy Code.' (***WebOrder.bpel->Enterprise Services...->Generate Deploy Code...***).

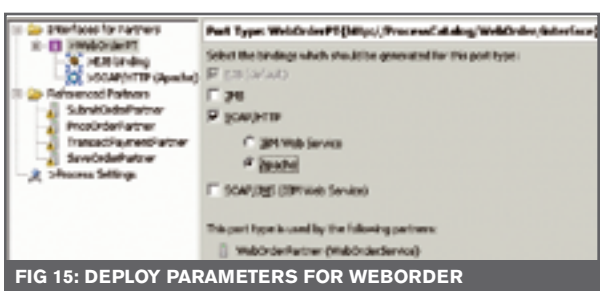


FIG 15: DEPLOY PARAMETERS FOR WEBORDER

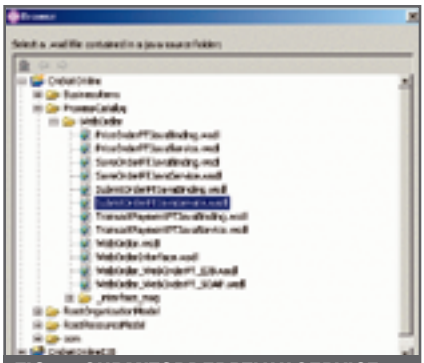


FIG 16: SUBMITORDERPTJAVASERVICE WSDL SELECTION

2. Click on *WebOrderPT* and check the fields as shown below. (We're going to use SOAP over HTTP as the mechanism to invoke the Web Service that we're going to create for our business process). Click ***Ok***. (See Figure 15.)
3. Each reference partner needs to be deployed. The deployment process for *SubmitOrderPT* is:
   - Right click on *WebOrder.bpel* and do 'Generate Deploy Code' again. (***WebOrder.bpel->Enterprise Services...->Generate Deploy Code...***). Click the *SubmitOrderPartner* reference partner. On the right-hand pane, click 'Browse' and select the *SubmitOrderPTJavaService.wsdl* file as shown Figure 16. The attributes should look like Figure 17. Click ***Ok.***
4. The same process has to be done for the remaining Reference Partners. The proper WSDL files have to be selected for each of the partners, namely:
   - *PriceOrderPTJavaService.wsdl* for *PriceOrderPartner*
   - *TransactionPaymentPTJavaService.wsdl* for *TransactionPaymentPartner*
   - *SaveOrderPTJavaService.wsdl* for *SaveOrderPartner*

This completes the deployment of our business process along with its reference partners.

Generate Service Proxy

Our process is now ready to be invoked by any service consumer. We're going to create a simple J2EE Web component to access our business process. To simplify the invocation of a Web Service with SOAP/HTTP, we're going to create a Web Service Proxy class that encapsulates all the details of service invocation and provides a simple-to-use Java API to access the Web Service from the client.

For our business process *WebOrderPT*, the WSDL that's created is named *WebOrder_WebOrderPT_SOAP.wsdl* (in the *ProcessCatalog.WebOrder* package). A Web Service proxy can be created from this WSDL file. The following steps needs to be performed:

1. Right click on *WebOrder_WebOrderPT_SOAP.wsdl* to generate the service proxy (***WebOrder_WebOrderPT_SOAP.wsdl->Enterprise Services...->Generate Service Proxy...***).
2. In the 'Proxy Selection' dialog box that appears choose 'Web Service Invocation Framework (WSIF)' as the type of the service proxy. Click ***Next.***

Subscribe Today!

– INCLUDES –
FREE DIGITAL EDITION!
(WITH PAID SUBSCRIPTION)
GET YOUR ACCESS CODE INSTANTLY!

*The major infosecurity issues of the day... **identity theft, cyber-terrorism, encryption, perimeter defense, and more** come to the forefront in* **ISSJ** *the storage and security magazine targeted at IT professionals, managers, and decision makers*

Editorial Mission

Greater Collaboration and Efficiency Through Education

✓ ***ISSJ*'s editorial mission is to showcase proven solutions** that will guide, motivate, and inspire senior IT and business management leaders in the planning, development, deployment, and management of successful enterprise-wide security and storage solutions.

✓ ***ISSJ*** brings together all key elements of data storage and protection, and presents compelling insight into the benefits, efficiencies, and effectiveness gained by focusing on these two critical areas of IT simultaneously.

✓ ***ISSJ*** is an objective, critical source of information that helps storage and security managers make informed management decisions about what is working today, and what they need to plan for tomorrow, and is the only publication that focuses exclusively on the needs of IT professionals who are driving the enterprise storage architecture/infrastructure while pursuing and incorporating the latest security technologies.

✓ ***ISSJ*** achieves our mission by delivering in-depth features, practical "how-to" solutions, authoritative commentary, hard-hitting product reviews, comprehensive real-world case studies, and successful business models, written by and for professional IT storage and security practitioners.

SAVE 50% OFF!
(REGULAR NEWSSTAND PRICE)

Only $39.99
ONE YEAR
12 ISSUES

www.ISSJournal.com
or
1-888-303-5282

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

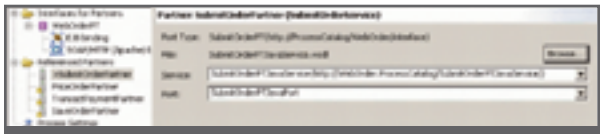


FIG 17: SUBMITORDER PARTNER ATTRIBUTES

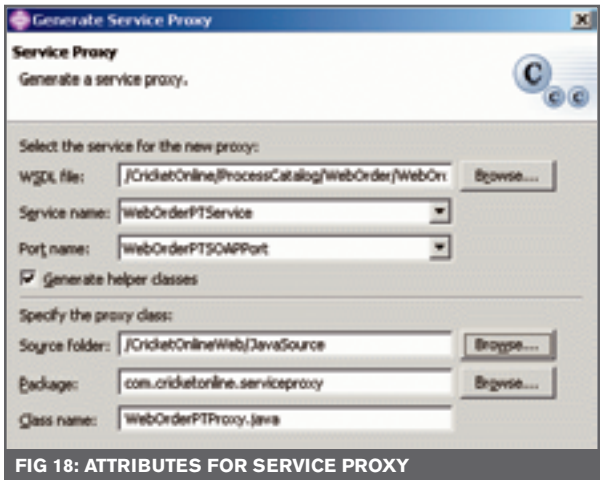


FIG 18: ATTRIBUTES FOR SERVICE PROXY

FIG 19: SERVLET ATTRIBUTES

```
public void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
        WebOrderPTProxy proxy = new WebOrderPTProxy() ;
        RequestData reqData = new RequestData() ;
        Order order = new Order() ;
        order.setNumBalls(30) ;
        order.setNumBats(10) ;
        order.setOrderID("10007") ;
        order.setStatus("new") ;
        reqData.setOrder(order) ;

        CreditCard card = new CreditCard() ;
        card.setCardHolderName("Sachin Tendulkar") ;
        card.setCardNumber("2030848495108574") ;
        card.setExpirationDate("07/2006") ;
        reqData.setCreditCard(card) ;

        String result = proxy.sendWebOrder_InputCriteria(reqData) ;

        resp.getOutputStream().println(result) ;
}
```

FIG 20: CODE SNIPPET FOR DOGET METHOD FOR SERVLET

3. Since our service consumer is a J2EE Web component, we'd like to create the Service Proxy in the Web Project in which our service consumer will reside. The process deployment steps create a Web (*CricketOnlineWeb*) and an EJB (*CricketOnlineEJB*) project as a part of the deployment process. We'll be creating our service proxy in the *CricketOnlineWeb* project. Refer to Figure 18 for the attributes. Note: The 'Source folder' is changed to '/CricketOnlineWeb/JavaSource' by clicking the 'Browse' button to the right.
4. Accepting all other default values, click ***Next.***
5. In the subsequent window, select the *sendWebOrder_InputCriteria* operation and click ***Finish.***

This creates our Web Service Proxy. We're now ready to create a J2EE Web component that accesses our business process through its Web Service proxy. This is the subject of our next section.

Implementing a Service Consumer

Let's keep this one simple. The easiest way to invoke the business process is to use a servlet. We'll create a servlet in the 'JavaSource' folder of the 'CricketOnlineWeb' Web project. Our servlet's attributes are shown in Figure 19.

For our servlet, we just implement the 'doGet(...)' method. The method's implementation creates instances of *Order* and *CreditCard* and sets their attributes. These instances are then added to a *RequestData* object instance and are passed on as a parameter to the *WebOrderPTProxy*'s 'sendWebOrder_InputCriteria' method. The output result is then sent back to the invoker. The snippet for the 'doGet(...)'method is shown in figure 20.

Having created all the required artifacts, we have to create a runtime to test our business process. This is the topic of the next section.

Creating the Runtime

A server needs to be created that will host the business process execution's runtime. The following steps needs to be followed to create the server and invoke the servlet to test our business process:

1. Switch to the *Server* perspective. On the 'Server Configuration' pane in the lower left corner of WSADIE, right click on the 'Servers' folder to create a new 'Server and Server Configuration.' Assign the new server a name and accept the rest as defaults (See Figure 21) and then click ***Finish.***
2. Add the 'CricketOnlineEAR' Enterprise Application project to the newly created 'CricketOnlineServer' server. This is done by right clicking on the 'CricketOnlineServer' project and then using the 'Add and Remove projects…' menuitem.
3. Start the 'CricketOnlineServer.' This is done by right clicking on the 'CricketOnlineServer' in the 'Servers' pane (on the lower half of the pane) and clicking 'Start' ( ). A successful start is denoted by the status line (See Figure 22) in the console.
4. Once the server has started successfully, switch back to the 'Business Integration Perspective.' Right click on *CricketOnlineServlet.java* (our servlet) and select the 'Run on Server...' menuitem. Select the defaults in the dialog box that comes up and click ***Finish***.
5. The servlet's 'doGet(...)' method will get called. This will open up a Web browser in the editor where the output result of the invocation of our business process is displayed. (See Figure 23.)
6. This completes the creation of the runtime and the execution of our business process.

Conclusion

This article took us from where we left in Part 1. The artifacts created in Part 1 were imported into a new Service

project in WSADIE. The BPEL process was modified to add the business logic for the *Map* variable. Each individual task was implemented in Java with its corresponding business logic. The implemented business process was then deployed. For ease of invocation, the tooling-rich WSADIE was used to create a Web Service Proxy for the Web Service definition of our business process. A servlet was also implemented that simulated the invocation of our business process through the service proxy. A runtime environment was created and our business process was invoked through a service consumer (servlet).

This two-part article showed us how to create an end-to-end business process in which the process modeling is done using the WBI Modeler tool; its artifacts are then used readily in a J2EE development environment in WSADIE to implement the business process, create a Web Service definition for the process, and invoke it through a service consumer, testing a total end-to-end scenario.

The main idea that this two-part article brings to its readers is the mechanism of using WBI Modeler and WSADIE in tandem to create, implement, and execute a business process through the use of these two very powerful IBM technologies.

Resources

- The WebSphere Studio Zone in IBM DeveloperWorks is the single most consolidated site for downloads, tutorials and other materials on WebSphere Studio. (http://www-106.ibm.com/developerworks/websphere/zones/studio/)

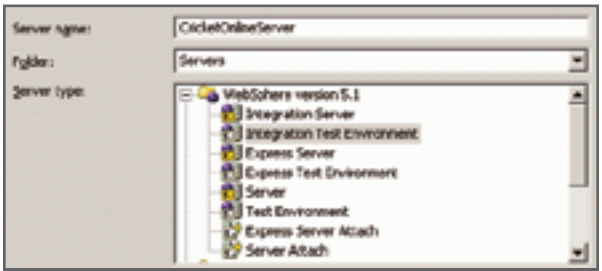


FIG 21: SERVER PARAMETERS


FIG 22: SERVER STATUS LINE


FIG 23: PROCESS INVOCATION RESULT

- You can download the latest version of WebSphere Studio Integration Edition (5.1.1) at www-128.ibm.com/developerworks/downloads/ws/wstudio/support.htmlzz.
- A draft version of an IBM Redbook on "Business Process with WebSphere Business Integration" is available at www.redbooks.ibm.com/redpieces/pdf/sg246381.pdf.

Disclaimer: Since this article was first written, the publicly available version for WebSphere Studio Integration Edition has been changed from 5.1.0 to 5.1.1.

In the last issue, WebSphere Journal (Volume 4, issue 4) *incorrectly titled part 1 of this series* From Business Modeling to Web Implementation. *The correct title should have been,* From Business Modeling to Web Services Implementation. WebSphere Journal *apologizes for the error.*

WebSphere JOURNAL | Coming Next Month...

WebSphere Commerce Design Patterns

Composite Applications, Part III

High-performance Serverside SOAP Processing

WebSphere System Integration Testing

Interview with Doug Wilson, CTO Lotus Software

Part 2 – RFID, Eclipse, etc.

DOUG WILSON

WebSphere Journal recently interviewed Doug Wilson, distinguished engineer and CTO of IBM's Lotus Software Division in Westford, Massachusetts. Part 2 of this interview addresses Eclipse and other topics, with concluding remarks by WebSphere Portal IT architect Richard Gornitsky about future WebSphere Portal enhancements.

WebSphere Journal: Okay, we were talking about vertical markets. One question we have is how is RFID driving things in retail? Are you seeing any thing of note in that area? And are there any other vertical market issues that you're facing that are worth noting?

Doug Wilson: RFID is, of course, a hot area right now. So far, we don't have RFID tags on people, so the interface between that and collaborative systems is still a bit early. But what is interesting about the relationship between the two is that RFID focuses attention in an area that we label sensors and actuators, i.e., the ability to have stations throughout a warehouse or a shipping organization or a point-of-sale that are sensitive to materials coming through them.

They can react to those materials, monitor things as well, and drive an infrastructure for pervasive computing: small, purpose-built devices and appliances that are "sometimes connected" or "often connected" or "fully connected."

We spend a lot of time and energy at the infrastructure level to support those things. That same technology is coming into the collaboration portfolio through Workplace Client Technologies and supports the mobility of users and workers. One of the thrusts of the Workplace initiatives is to make sure that employees can get access to and act on and react on information, whatever their mode of connection is, whether they're sitting in the office, at an airport kiosk, on their handheld or phone device.

WJ: More or less adjusting their needs when and where they need their information.

DW: Absolutely. And again, I stress that it's not just a one-way push of information, but it's the ability of remote employees to take action on that information that separates some of this.

WJ: A virtual office, so to speak.

DW: Yes.

WJ: Earlier you mentioned Lotus supporting new technologies. One of those, of course, is Eclipse. How does everything we've been talking about so far tie in to the underlying Eclipse framework and into the broader focus of moving some of the Lotus product line into the J2EE environment itself?

DW: Well, that's a great question. Building up from this notion about pervasive computing and "sometimes attached" and all the modalities that it introduces, comes the question: How do you manage those devices and how do you manage that platform? And by the way, is the desktop system that we think of today on the laptop really any different from that standpoint than any other device that might be connected to the network?

WJ: There is a merge going on between devices. Each device seems to become pretty much a complete solution....

DW: Or at least capable of operating complete solutions.

WJ: Right.

DW: The problem then becomes: How do you manage all of it? Our philosophy is to drive the management of the capabilities of those devices from an

Doug Wilson and Roger Gornitsky were interviewed by Roger Strukhoff, editor-in-chief of *WebSphere Journal*.

information model at a central location or server location based on the employees' role and identity, and, of course, their role with respect to the business processes that they have to engage in. That management model and the corresponding aggregation of capabilities gives rise to the need for a desktop system capable of being managed remotely, capable of providing high-quality user-interface interactions and also a programming model for applications to be built in that space. For us, that's Eclipse.

We've been working with the Eclipse.org group in a project called Equinox to drive into the Eclipse foundation the capabilities of remote management, the capabilities and extensions of the environment that will use it as a base application platform. That core technology, as well as being the core technology behind IBM's tooling portfolio, is also the core technology behind the Workplace client technology.

WJ: Are security capabilities included in various aspects of the Lotus software and tools?

DW: Absolutely. For example, the local file stores or the local document stores that we provide in Workplace client technology are always encrypted. That encryption is managed with public key certificates or interfaced with the security management system that a customer may have on-premises already.

So, an interesting difference between the engineering that we're doing now in the portfolio, and that perhaps was current at the origin of Notes and over the '90s, is that in those days there were no standards or widely available public capabilities for implementing secure systems. And so by necessity those capabilities were, to use the "P" word, proprietary. That is, while they were based on open and well understood algorithms, the key management infrastructures and the like were necessarily unique to each application.

That world has changed. And now there are many more open standards and systems available for us, and protocols and well-understood norms in the industry for managing the essentials of user identity and security, and the art form in the new portfolio is to take advantage of those without turning back the clock on what the security capabilities are.

WJ: How do you view the competitive landscape with, say, Microsoft Exchange or Novell Groupware, and what sort of market share do you envision in the short- and long-term for IBM and Lotus Notes and Domino products?

DW: Well, I think probably as far as I can go is to say an increasing share. In the space of mail and messaging the market is pretty mature. There isn't likely to be a lot of transition one way or another. Our Notes customers are well serviced in that space and we assure them that we are not going anywhere with the Notes Domino portfolio that they don't want us to go. It's a strong, stable investment, and we are committed to taking them forward without causing them to have to rip and replace the infrastructures that they have invested in over the years, and the programs and customizations and behavior they've invested in.

On Achieving markets share... I think it is important to bring forward a new idea. And our new idea is based in a new notion of collaboration that we call activity-centric computing. In a sort of short sentence, activity-centric computing is about managing the business activities and the thread of events in a business activity, rather than concentrating on managing the artifacts of that business activity.

Systems to date have typically concentrated on managing those artifacts, which is fine and important and we need to do that, but that doesn't bring enough value to the party. In traditional systems, the user is still responsible for the correlation of all of those artifacts with their business problem. They have to know that the contract is in this document and that the financials are in this document and that the chat we had yesterday and the three mail messages you sent me are all relevant to a particular business operation.

WJ: So it's about identifying relationships.

DW: Yes, identifying relationships, helping the users manage those relationships so that as they multi-task through their day the system is assisting them in that multi-task rather than contributing to a flood of information and simultaneously a flood and scatter of information. We view it as important as associating information with business contexts, with business processes that may be simple like a customer number or a customer event or a request for a response

Doug Wilson,

CTO, LOTUS PRODUCT DIVISION, IBM

Doug Wilson is an IBM Distinguished Engineer and chief technology officer for the Lotus Product Division of IBM's Software Group. Prior to this he was responsible for architectural strategy for Domino and WebSphere integration, WebSphere Portal Server and WebSphere content manager, and WebSphere Personalization. He invented and managed the design and development of Lotus InfoBus technology, a data exchange technology for Java components that has been accepted for inclusion in JavaSoft's Java Developer's Kit as a Java standard. Doug holds a degree from M.I.T.'s Department of Mechanical Engineering.

douglas_wilson@us.ibm.com

"RFID is a hot area right now; we don't have RFID tags on people so the interface between that and collaborative systems is still a bit early"

to a proposal or any number of business events that require the correlation of a number of different business activities and a number of different artifacts.

WJ: Is this something we'll be hearing about more often in the near future?

DW: Yes, in fact, you'll see it in the 2.5 version of IBM Workplace. We have a capability called Activity Explorer, which is our first product being released in this space, and we think it provides a lot of great capabilities and is meeting with great reception from customers who have seen early versions of it and presentations about it.

WJ: We understand that you recently judged the IBM Lotus CTO award at the Lotus Sphere 2005 event. Who won and what particular aspect of Lotus technology did they employ?

DW: We gave out three awards, one in each of the major geographies, so an award in Asia-Pacific, an award in EMEA and a North American award. The award speaks to innovation in the building of systems. What we were really looking for with that award was a better idea, an idea that embraces the notions of Workplace that we've been putting forward. We've talked about two of them today.

One of them is this notion of activities and the other is a notion of context; that is how do you apply a business context to the operation of systems. And the winners this year showed great understanding of that problem and really forward thinking in the areas of their application systems, and that's what we were looking for.

Interview with Richard Gornitsky, Part 2

IBM WebSphere Portal IT Architect discusses enhancements

RICHARD GORNITSKY

WebSphere Journal: What would you say we can anticipate in the near future in terms of increased capabilities and enhancements to future releases of WebSphere Portal?

Richard Gornitsky: Well, if you looked at the first versions we really focused on functionality. Then, we looked at improving installation, scalability, and availability.

Now, it's time for even more interesting enhancements, specifically in the usability, deployment, search and content management area. That is, the deployment of WebSphere Portal applications, not the installation. The installation is very easy to do. But when you need to portal applications to hundreds of server, we're providing more and more sophisticated tools to assist you in that.

WJ: Could you elaborate a bit more on availability?

RG: WebSphere Portal has provided availability support through WebSphere clustering. But availability is more than load balancing. Availability is dependent on a well implemented process; development and operational. The process requires tools to support it and WebSphere Portal integrates tightly with the "best of breed" tools; Tivoli's and Rational's.

WJ: Are there any other products that IBM has acquired recently that will also be integrated or provided support through WebSphere Portal?

RG: Yes, definitely and you will start seeing a lot more. An exciting area in which I see tremendous focus on is enterprise system management which includes performance management. I see more tools coming out that will help users drill down to identify problem areas and provide tuning recommendations.

Final Thoughts?

The world is not flat

BY ROGER STRUKHOFF

Each month, as we reach the end of the publication cycle for the print version of *WebSphere Journal,* a conversation ensues about the nature of this column, "Final Thoughts." What should it be, who should write it, should it address daily reality or long-term thinking, how technical should it be, etc.?

Last month, the column's author launched into a pseudo-philosophic discussion about the nature of a sphere and how it shapes our metathoughts as we create our company's IT architecture, infrastructure, and daily procedures. It's interesting to see that the column was published just as a new book called *The Earth is Flat* was being readied for global distribution. This book promises to carry some influence in the coming months and years, as its author is well-known and its premise just the sort of contrarian weltanschauung that media pundits and business leaders love to dissect and discuss.

The only problem is that the earth is round, or rather, spherical. A flat earth doubles the distance between many places, say, Silicon Valley and India, or New York and China. Go ahead and say that the book's author didn't mean his title to be a literal statement. Even taken metaphorically, the flat-earth conception implies an abstraction, i.e., a map, that envisions things quite different from what they are. Mercator and countless others have struggled with this idea for centuries.

Organizations have become flatter, true, but this trend has been in evidence for at least 20 years. There is nothing fundamentally new about eliminating management layers, getting closer to customers, and using IT to do so. The fact that web service development environments and SOAs are the latest way (and most powerful so far) to address this trend only highlights the fact that the concept of flatter organizations is increasingly mature.

But the world is not flattening. The concept is facile, misleading, and just plain wrong. And whether using words to write code or to write books, you have to get it right.

Cultural differences among certain classes of businesspeople, technologists, academicians, and teenagers have been *rounded off,* perhaps. The rapid rise of the blogosphere has *enriched* the volume and *increased* the signal level of our global public conversation. And ever-increasing outsourcing and offshoring has certainly *stretched* supply chains, but at a cost of seriously sleepless nights when an old-fashioned labor slowdown or barge accident at a major port can threaten to bring a serious percentage of the global economy to a grinding halt.

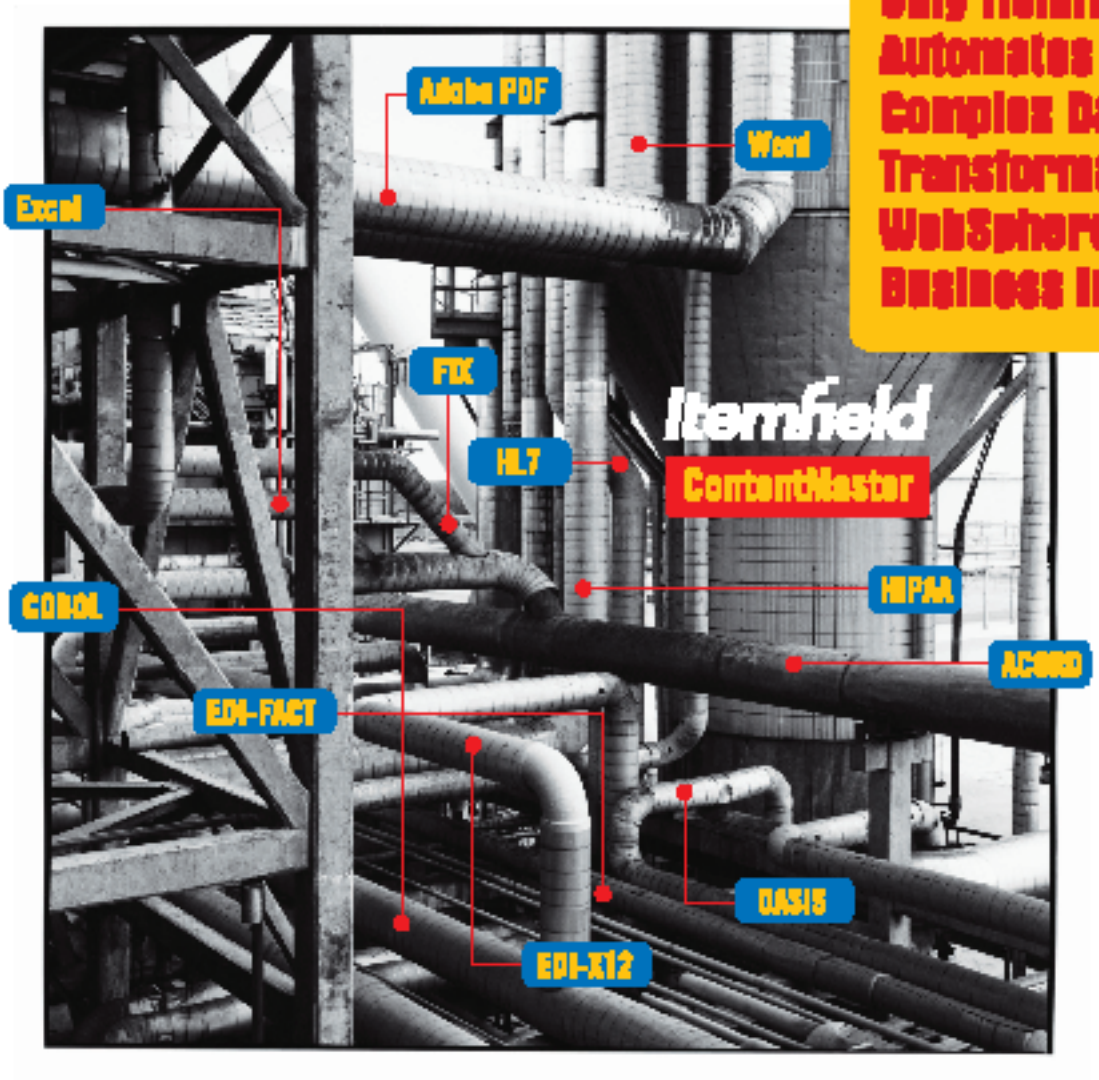
The world is not flat, and it's not getting flatter. It's getting better connected, and it's getting rounder, as we move to a nonstop, 24-hour business cycle that circles back on itself at whatever arbitrary time has been selected for you, depending on your time zone.

In that spirit, this column can never truly provide Final Thoughts. No such thing anymore. As implied at the top of this column, we've reached the end of another production cycle for this magazine. But yet, we haven't. The online version of this and other SYS-CON Media (www.sys-con.com) publications is continuously updated, and the content of our print versions is conceived within this context, not as a separate entity or product line.

Any thoughts we have time to think may be the last thoughts we have in any particular day, but they are hardly final. Our world is multidimensional, hardly flat, and not getting any flatter. How about yours? Send us your ideas for this column. We are looking for our readers' conceptions of the universe, as well as their daily reality. (Send your ideas to roger@sys-con.com). We need to know, how flat is your world, and how final are your thoughts?

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 cofounded 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 www.rssblog.linuxworld.com. Contact him at roger@sys-con.com.

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 to Business Integration"* and to view the ContentMaster Solution for WebSphere Business Integration demo.

Itemfield | 1 800 438 1055 | www.itemfield.com/wbi

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