SYS-CON MEDIA

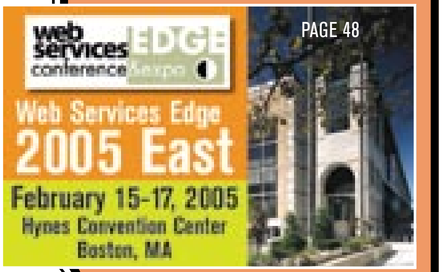
MANAGEMENT • DEPLOYMENT • IMPLEMENTATION

WebSphere® JOURNAL

www.SYS-CON.com/WebSphere

The World's Leading Independent WebSphere Resource

NOVEMBER 2004 VOLUME 3 ISSUE 11



FROM THE EDITOR

Security, Firewalls, and Keeping the Door Locked

BY JACK MARTIN • PAGE 4

BOOK EXCERPT

Building a Combined WebSphere and Domino System

BY JOHN LAMB, MICHAEL LASKEY, & GOPAL INDURKHYA

PAGE 40

FINAL THOUGHTS

Fire Your QA Department and Increase the Quality of Your Software

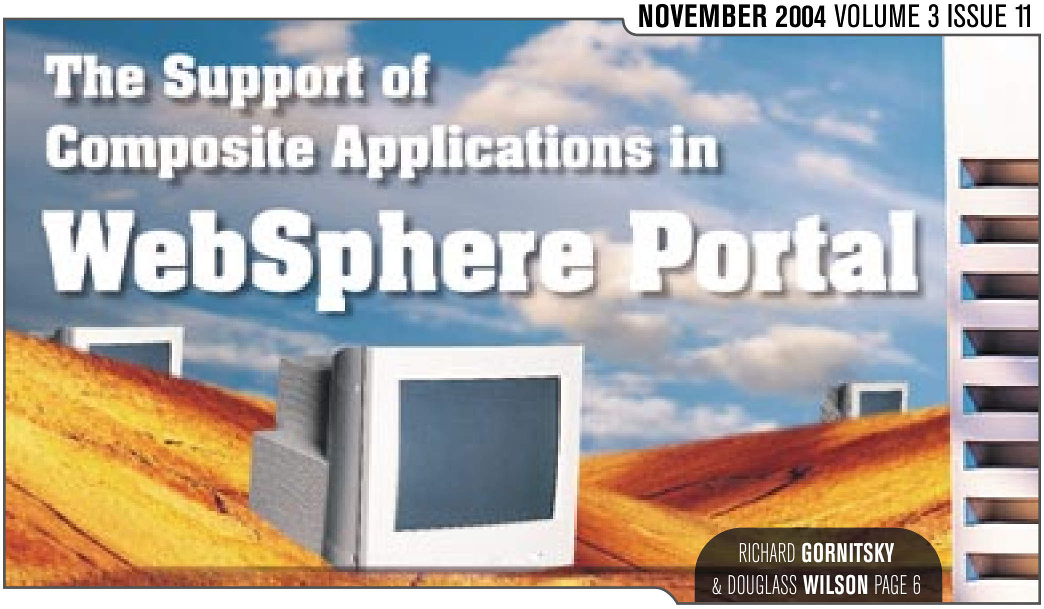
BY MIKE TAYLOR • PAGE 50

DISPLAY UNTIL JANUARY 31, 2004

$8.99US $9.99CAN

11

0 09281 03422 3



The Support of Composite Applications in WebSphere Portal

RICHARD GORNITSKY & DOUGLASS WILSON PAGE 6

How to Simplify IT Infrastructure Management

Using SOAs is key in adapting to change

BY DAVID COX

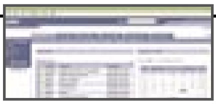
PAGE 10

Exchanging Trade Information Among Mercosul Member Countries

A case study

BY PAULA G. DANTAS & ANDRÉ TOST

PAGE 16



IT PowerPAC WS from Eden

Product Review: More than a help desk

BY JAY JOHNSON

PAGE 20

Deploying WAS Applications Using Maven

A simple and powerful build tool

BY BRENT WORDEN

PAGE 24

Can IT Systems Meet the Challenges Facing Life Sciences Organizations?

IBM's middleware solutions do it all

BY C.J. FROST

PAGE 32

Implementing Sarbanes-Oxley

Using IBM Lotus Workplace for Business Controls and Reporting

BY ASHWINI KUMAR & RAGHAV MATHUR

PAGE 36

Security, Firewalls, and Keeping the Door Locked

BY JACK MARTIN

As the world becomes progressively more comfortable with e-business as a business model for the 21st century, criminals are also gaining a level of comfort in stealing information at a progressively alarming rate. Information theft is a global problem that will continue to grow as criminals become more adept at using technology as a tool to a achieve their goals.

To get a flavor for how widespread this problem could become, you only need to remember that most people have been using the same password for years on many systems. Also, there is no way to control who an employee may knowingly or unknowingly share critical security information with. Last, the entire crime is virtual – the criminals can be located anywhere in the world, strike without warning at anytime, and disappear back to wherever they came from.

You only need to look at what the U.S. Secret Service has been up to with domestic and foreign law enforcement officials, the U.S. Department of Justice, and investigators from the financial services industry. Recently they have announced the arrests of 28 individuals. The suspects were located across eight states and six foreign countries and were involved in an organized global cybercrime network. Charges against the suspects include identity theft, computer fraud, credit card fraud, and conspiracy.

The following excerpt is from a recent announcement of a successful investigation. "Led by the Secret Service Newark Field Office, investigators from nearly thirty domestic and foreign Secret Service offices and their global law enforcement counterparts have prevented potentially hundreds of millions of dollars in loss to the financial and hi-tech communities," said Secret Service Director W. Ralph Basham. "Information is the world's new currency," he continued. "These suspects targeted the personal and financial information of ordinary citizens as well as the confidential and proprietary information of companies engaged in e-commerce."

"Identity theft carries a heavy price, both in the damage to individuals whose identities are stolen and the enormous cost to America's businesses," said Attorney General Ashcroft.

Working in cooperation with the U.S. Attorney's Office for the District of New Jersey, the Computer Crime and Intellectual Property section of the Criminal Division of the Department of Justice, and other U.S. Attorneys' offices and law enforcement agencies, the indictment is a result of a yearlong investigation.

Operation Firewall began in July 2003 as an investigation into access device fraud. The case evolved into a highly technical, transnational investigation involving global credit card fraud and identity theft over the Internet.

The criminal organizations operated Web sites used to traffic counterfeit credit cards and false identification information and documents. These groups are highly organized international criminal enterprises that use Web sites to promote and facilitate a wide variety of criminal activities including electronic theft of personal identifying information, credit card and debit card fraud, and the production and sale of false identification documents. After initial contact via the Internet, the suspects exchanged stolen information and counterfeit documents such as credit cards, driver's licenses, domestic and foreign passports, and birth certificates.

–continued on page 39

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

WebSphere® JOURNAL

Advisory Board

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

Editor-in-Chief	JACK MARTIN
Executive Editor	GAIL SCHULTZ
Contributing Editor	PATTI MARTIN
Product Review Editor	JAY JOHNSON
Editor	NANCY VALENTINE
Associate Editor	JAMIE MATUSOW
Assistant Editor	NATALIE CHARTERS

Writers in This Issue

David Cox, Paula G. Dantas, C.J. Frost, Richard Gornitsky, Gopal Indurkhya, Jay Johnson, Ashwini Kumar, John Lamb, Michael Laskey, Jack Martin, Raghav Mathur, Mike Taylor, André Tost, Doug Wilson, Brent Worden

Subscriptions

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

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

Editorial Offices

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

WORLDWIDE NEWSSTAND DISTRIBUTION
CURTIS CIRCULATION COMPANY, NEW MILFORD, NJ

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

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

SYS-CON MEDIA

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.

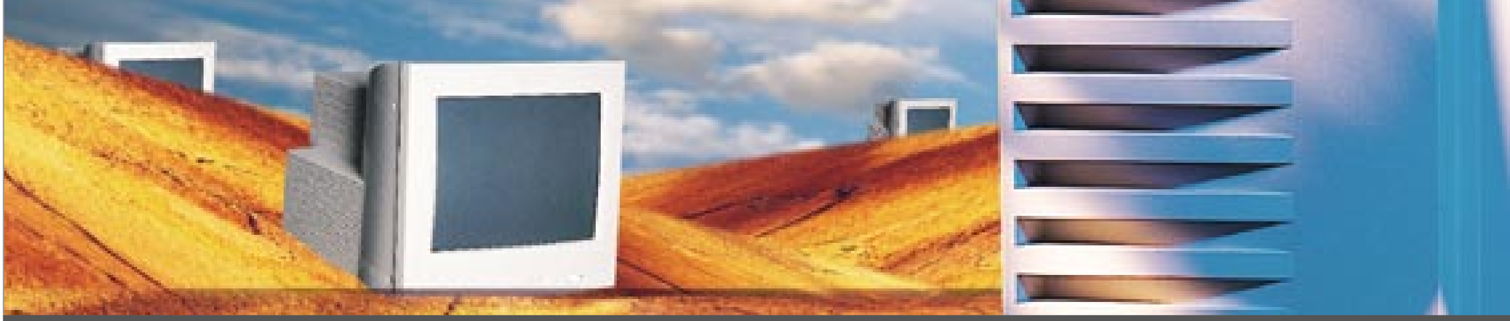
Unless of course, you've discovered the power of Wily. No other software offers our level of insight. Which means you'll be able to monitor transactions and collaborate with colleagues in real-time. Even more important, you'll be able to optimize performance across the entire application—end-to-end.

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



The Support of Composite Applications in WebSphere Portal

The basis for advanced suites of business functions

BY RICHARD GORNITSKY & DOUGLASS WILSON

Richard Gornitsky is a WebSphere Portal architect with IBM's Software Services for Lotus. His expertise is in integrating WebSphere Portal into Fortune 500 firms from concept to production. Richard's 23 years of industry experience includes finance, insurance, telecommunications, pharmaceutical, software manufacturing, and retail/distribution. He has experience in the full life-cycle development of high transaction solutions, which includes simultaneously managing multiple large complex application development projects. Richard is a requested technical speaker and is a coauthor of Wiley Technology Publishing's *Mastering WebSphere Portal.*

rsgornit@us.ibm.com

What are composite applications and how do they help me? In this three-part series we will introduce the concepts behind a new class of applications called composite applications and explain their need and structure. In this article, we will present a high-level overview of composite applications, the benefit they provide, and how WebSphere Portal supports them. In the next two articles we will explore the WebSphere Portal features that support composite applications in more detail.

The term composite application suggests that there is a class of application whose fundamental construction model is the *composition* of parts or components. These applications are formed by choosing components from a catalog or palette, placing those components on some design surface, and interconnecting the components to create the behavior desired.

A simple composite application is shown in Figure 1. This hypothetical application allows the user to choose a major or minor application function from a navigator of choices in the upper left area. Each function is imagined to consist of a record set and a detailed view of that record set. In addition, a context-sensitive area for help and advice is provided for each of the major function sets in the application. A typical application might include a layout like the one shown in Figure 1.

There are, of course, many approaches to building this kind of Web application. Many Web application developers might approach the structure of this application as a frame set where clicks in the function selection area trigger JavaScript actions that select a new URL target for the record summary frame and for the instructions frame. Similarly, clicks in the summary area cause a new target selection for the detail area. Developers familiar with the Struts framework might see this as a set of tiles interconnected by the event handling mechanism of the Struts model.

With composite applications, we offer another alternative for the construction of this kind of application. More and more independent software vendors (ISVs) and corporate developers are searching for better ways to reuse and repurpose computing assets. Developers have used techniques like code libraries, object-oriented class libraries, and other mechanisms to achieve reuse; but the advent of portals and the standardization of a component model in portals (the portlet) bring with it a new and more flexible mechanism for composite applications within WebSphere Portal v5.1.

Composite Applications within WebSphere Portal v5.1

The uses of WebSphere Portal are as diverse as the customer base that has adopted this technology. Portal

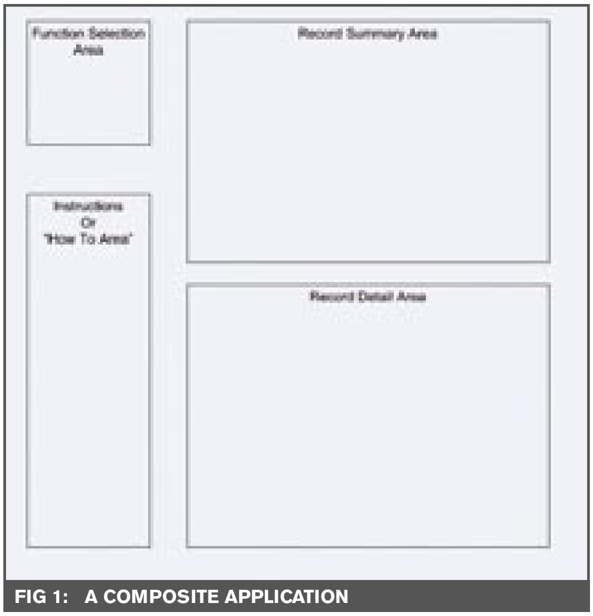


FIG 1: A COMPOSITE APPLICATION

implementations range from simple information sharing to customer relationship portals to corporate performance dashboards, and even advanced applications suites. What has made WebSphere Portal such a flexible and productive platform is its architecture.

WebSphere Portal removes from applications many of the design elements related to infrastructure and GUI support, and allows the developer to focus on creating business functions in his/her application. Support for security, workflow, transcoding, internationalization, navigation, integration, and more are provided by the WebSphere Portal platform while the developer creates portlets or “composite applications” to provide a container of business functionality.

WebSphere Portal also enables segmentation of the development among a team of people that minimizes dependencies on the deployment process. The creation of the page layout, the design of the look and feel, and the creation of the portlet/composite application can all be done independently of the development of the application components. The developer creates the composite application based on business needs. They do not have to worry about look and feel since that is controlled by the “themes” and “skins” that are created independently by the Web designer. Support of navigation, security, and devices is made simpler by WebSphere Portal’s infrastructure features.

More and more, we see WebSphere Portal becoming the basis for advanced suites of business function. Figure 2 shows an example of a composite application built by combining portlet components in WebSphere Portal.

In this example, each of the components is a small, independent application. But when composed together on a portal page, these independent applications can work together as one composite application. The drop-down menu in the portlet on the upper left is used to “send” information from the Orders for October portlet to all of the other portlets on the page; each of those portlets responds by displaying information related to information from the orders portal (e.g., the order details or shipping information).

A further example of an advanced suite of business functionality is IBM Workplace, which is a system of composite applications. Each major function is a composition of advanced “portlets” on a portal “page” using capabilities of the portal framework to interact with other components on the page. It provides a suite of collaboration and communication components that can be customized on a page based on a user’s preference. IBM Workplace provides support for word processing, spreadsheet support, presentation generator, messaging, team collaboration, collaborative learning, Web content management, and collaborative document management.

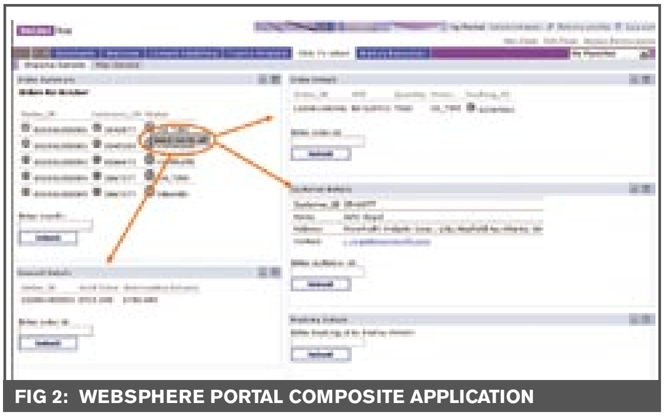


FIG 2: WEBSPHERE PORTAL COMPOSITE APPLICATION

Figure 3 shows the IBM Workplace architecture and the layered design. The Workplace infrastructure services/components are provided by WebSphere Portal server, which in turn is built on the infrastructure/middleware services provided by WebSphere application server.

Portlets: The Modular Building Blocks

As illustrated in Figure 3, composite applications within WebSphere Portal are based on portlets. Portlets can be considered to be modules of the composition model since they are each self-contained applications that support multiple states and view modes.

While simply exposing business data and operations through a portlet offers a lot of value, even greater value can be obtained when portlets work together. To create a composite application they must be able to pass data to each other, react to the data, trigger events, and respond to events. Communication and interaction with other composite applications are through defined interfaces such as messaging, event handling, or services. Each portlet can be removed/added from the model or page without impacting other portlets. The foundation of the composition model is the portal infrastructure which provides access to user profile information and remote content, the ability to communicate and trigger other portlets and to participate in windows and action events, store persistent data, and look up credentials.

Now let’s explore the different methods that portlets provide to support the composite application model.

The simplest and most basic method to implement is portlet messaging. The portlet messaging API enables

“What has made WebSphere Portal such a flexible and productive platform is its architecture”

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.
dwilson@us.ibm.com

“The tool that makes composite applications possible is the portlet wiring tool”

a portlet to send a message to either a specific portlet on the same page or all the portlets. In this model, the message is sent through the portlet action listener and received through the portlet’s message listener. Of course the biggest limitation of this interface is that it is programmatic and the message logic needs to be explicitly coded in each portlet. Further, any portlets that want to communicate using portlet messaging must have prior compile time knowledge of each other.

Messaging only enables you to send and receive data – a very limited functionality. The ability to trigger an event based on the data comes with cooperative portlets. Cooperative portlets are portlets that can declare, publish, and share information with each other using the WebSphere Portal property broker. The WebSphere Portal property brokers enable portlets to publish their typed data items, or properties, and to declare actions that they are willing to execute based on the data types published by other portlets.

The portlet that provides a property is called the “source portlet” while the properties that the source portlet publishes are called “output properties”. The portlet that receives a property is called the “target portlet” and the properties that are received by the target are called “input properties”. With cooperative portlets, the target portlet can provide an action based on the property/data it receives. This greatly enhances and supports our composition model by enabling triggering/workflow models. The whole topic of cooperative portlets as the critical component of WebSphere Portal composition applications will be discussed at length in the next article.

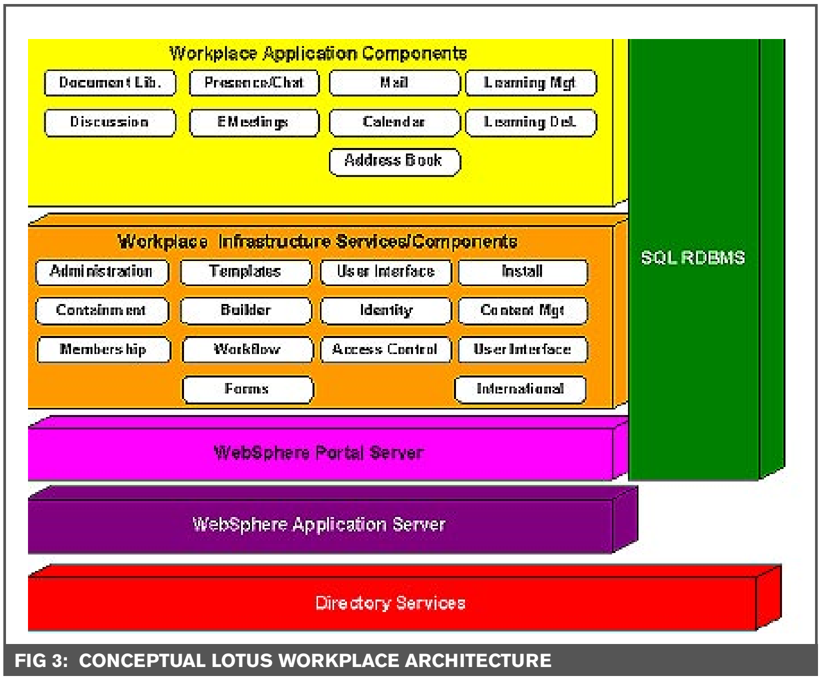


FIG 3: CONCEPTUAL LOTUS WORKPLACE ARCHITECTURE

Creating composite applications with portlets still requires best programming practices. For a portlet to be part of the composition model it must have a well thought out set of capabilities, which it exposes as actions on data types. It must also publish data types that are meaningful to users who will be combining portlets in applications. We will discuss these considerations in more depth in the next articles. Portlets used in composite applications should always adhere to a well-patterned framework such as the Model/View/Controller pattern and ensure that the portlet model is not circumvented by linking portlets by passing information through cookies or URLs (work around to perform page-to-page communication). Portlets have to be designed as session-independent, loosely coupled, encapsulated components that use instance and class variables only for read-only values. Your JSPs should have minimal Java and JavaScript. JavaBeans should be used to encapsulate the interface and pass data to the view JSP for rendering.

Enterprise Extensions Support for Composite Applications

In an enterprise environment, composite applications cannot be limited to inter-module/portlet communication. The components of an application must be able to communicate/interact with other applications on other platforms in local or remote locations.

WebSphere Portal provides this support using multiple vehicles. The most basic support can be done using Web services or J2EE connector architecture (JCA). JCA is the J2EE standard architecture to enable integration with enterprise information systems. WebSphere Portal provides JCA connectors for systems such as SAP, PeopleSoft, Oracle, CICS, IMS, Host-on-Demand, etc.

Integration can also be performed with Web services. Using the wizards provided with WebSphere Studio Application Developer, portlets can be developed that are Web service enabled. The Web service wizard enables you to use the UDDI explorer to:

- Search a UDDI registry for a Web service
- Find the service you need
- Find the WSDL file describing the Web service
- Import the WSDL definition, which creates a Java proxy

object with the same interface as the object implementing the Web service

But a true distributed enterprise composite application requires more than back-end integration. It needs to seamlessly integrate applications over multiple platforms while executing the applications on their native platforms. WebSphere Portal provides this support with support for Web Services Remote Portlets (WSRP). WSRP enables you to publish your JSR 168–compliant portlets to be used by other portals, or you can use remote portlets from within your own WebSphere Portal instance.

Advantages of Using a Portlet/Page-Based Composition Model

You may ask, "Why should I adhere to this model? There are plenty of models out there, why is this better?"

The portlet/page composition model enables rapid implementation of business functionality and reuse by using out-of-the-box portlets from a public catalog such as the IBM portal and Lotus workplace catalog (http://catalog.lotus.com/wps/portal/portalworkplace), or from a private catalog with custom portlets. It also enables ease of customization (which will be discussed in detail in the third article), and built-in authorization and access control to application features since each major feature is a page and each minor feature might be a portlet on the page. It provides built-in support for styles and branding and, most importantly, the ability to seamlessly bring together application functions from a variety of back end systems. Portlets are fundamentally about providing access to back-end information and transaction systems; composition of application function from these differing back-end systems is a natural for portal-based systems.

Tool Support for WebSphere Portal Composite Applications

Now you agree that this is a good model and that it could provide many benefits to your organization. However, at first glance it looks like it requires low-level Java coding, which can be challenging. Wrong! WebSphere Portal provides many tools to rapidly develop components or portlets that adhere to the composite application model.

At the basic level, you can use WebSphere Studio, which provides a sophisticated portlet development wizard that, within a few clicks, can enable you to develop a native or JSR 168 portlet. Within the native portlet, you can make it Struts compliant, support actions, messaging, Web services, cooperative functionality, JCA, JDBC, etc. After you create it and modify the code to support your functionality, you can debug it in the WebSphere Portal test environment local to WebSphere Studio.

At the next level, you can use the WebSphere Portal Application Integrator (WPAI). This portlet-based tool enables users to create portlets that can access and manipulate data in an enterprise application. Using a wizard, users specify what data should be displayed in your portlet by selecting business objects and fields exposed by the enterprise application. No programming is involved. The tool supports PeopleSoft, Siebel, SAP, and JDBC connections to Oracle or DB2 through WebSphere data sources and Lotus Domino.

The Web clipping portlet enables you to quickly take a Web fragment and make a component out of it. You specify how to obtain the external HTML pages and then identify the area of those pages that should be displayed in the portlet. The resulting Web clipper is stored as an instance of the Web clipper runtime portlet.

Finally, the tool that makes composite applications possible is the portlet wiring tool. Programming cooperative portlets can be challenging and also unnecessary for many of the composite applications. The portlet wiring tool allows you to configure connections, or wires, between portlets on a page that register with the property broker to share information with other portlets. Portlets that are wired together will react based on the relationship established in the portlet wiring tool. Once these portlets are wired, they will react to changes to the other portlet(s) relationship that has been established resulting in a simultaneous update to one or more portlets with minimal user interaction.

Web content is a fundamental building block of any portal. It can be organized into components by the tools exposed in the Lotus Workplace Web Content Management content viewer. These viewers allow their content as well as the navigational method to be displayed in a portal page. They are also designed to allow a navigational tree or menu to be displayed in one portlet on the page to control the content that is rendered on another portlet on the page.

These are just some of the tools to assist the implementation of the composite application models within WebSphere Portal. Improved support will come in WebSphere Portal v5.1. This will be discussed in greater detail in the last article of this series.

What's Next?

Clearly this article presented a high-level discussion on how WebSphere Portal supports the composite application model. In the next articles we will go into more detail (with programming examples and architecture diagrams) on the various methods that are provided in WebSphere Portal v5.1 to support this model.

Specifically, we will focus on cooperative portlets and methods to provide end-user customization. The last article will focus on advanced topics for composite applications such as enhancements in WebSphere Portal v5.1 to support composite applications, performing portal navigation from your components, using labels, using portlets as pop-up windows or "dialogs," enabling "Solo Mode," and forming URL commands to open new windows.

By the time we have finished, you will appreciate the composite application model and how WebSphere Portal's support of it will enable rapid development and deployment of your Internet or intranet applications.

Using SOAs is key

How to Simplify IT Infrastructure Management

BY DAVID COX

David Cox is a senior technical staff member in Tivoli's Technical Strategy and Architecture group. He is responsible for creating the Web services and service oriented architecture management strategy for IBM Software Group. David has 20 years of technical experience in system and network management, communications software, and operating systems. He has written numerous technical papers and holds five U.S. patents. David received a BS from North Carolina State University and an MS from the University of North Carolina.

The velocity of business is ever increasing, and business agility is at a premium. Companies are faced with a myriad of challenges such as managing service levels in a complex, heterogeneous environment, increasing resource utilization while maintaining availability and reliability, and reducing IT costs. Infrastructure management provides the capabilities to help simplify and optimize IT infrastructures so that businesses can respond with agility, flexibility, and speed.

Through the implementation of a service-oriented architecture (SOA), companies can simplify the management of their IT infrastructures and more aptly adapt to changing business conditions. SOAs are collections of business processes that use reusable standard interfaces to integrate applications within a company as well as with customers and suppliers. By implementing an SOA, companies can drive down costs, find more opportunities for growth, and help transform their organizations into on-demand businesses.

An SOA is a component model that matches business processes directly to the corresponding technology components needed to execute the business process. Data moves through an SOA through "services", which have well-defined interfaces and contracts. Industry-standard Web services work regardless of the specific hardware platform, operating system, or programming language in the customer's IT infrastructure; this allows them to interact with each other in a uniform and universal manner.

> **"By implementing an SOA, companies can drive down costs, find more opportunities for growth, and help transform their organizations into on-demand businesses"**

The promise of open, standards-based, reusable services is a great opportunity for a business. Services can communicate with each other using standards-based communications capabilities provided by an Enterprise Service Bus (ESB). The ESB provides the connection infrastructure for business transactions to flow from application to application in an SOA.

Web services is an enabling technology for an SOA. Web services enables customers to accelerate the exchange of data. This technology helps customers transform to an on-demand business by easing interoperability of programmed IT services, as well as integration of applications into a company's broader business processes.

The distinction between SOA services and Web services lies in design. The SOA framework does not exactly define specifically how services should interact, just how services can understand each other and how they can interact. It is the difference between defining a strategy of how a process is to be done and the tactics of how it is actually done. Web services on the other hand are simply industry-standard software that define how data is exchanged between applications. Thus, Web services essentially provide the technology to implement an SOA.

Managing an SOA requires many of the same disciplines as the management of any application environment. Some of the high-level management needs are:

- ***Development:*** Services and infrastructure in an SOA should be developed from the start to be self-managing. This saves time and money during the development phase, and improves the time-to-value for getting a manageable application deployed.

- ***Deployment:*** Services need to be deployed into an SOA just like any other application component. The deployment and configuration tools need to understand the relationships between a service and other components in the environment, such as other applications or databases. Deployment tools also need to deploy and configure the other various "touch points" that the service will connect to, including the service implementation, the service client stubs, the service interface definition, and any registry entries for the service (see Figure 1).
- ***Security:*** Security management ensures that services in an SOA communicate in a secure, reliable way. Security management also ensures that users are accurately and consistently identified, and have access only to the appropriate applications. Secure interoperability increases the amount of integration that a business can have with it customers, partners, and suppliers.
- ***Performance and availability:*** Performance and availability management tools ensure that SOA environments are always available to the customer, and are always performing to business expectations and commitments. This reduces the cost of manually monitoring, diagnosing, and correcting problems. It also reduces the cost of missed business commitments and increases customer satisfaction.

SOAs differ from typical distributed application architectures in significant ways. The following are key aspects of an SOA that affect management:

- Services have well-defined interfaces
- Services have well-defined relationships
- SOAs use standards-based communications
- SOAs are more closely aligned with business processes

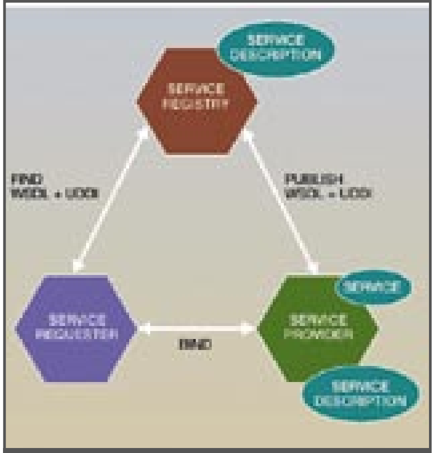


FIG 1: WEB SERVICES ACTORS, OBJECTS AND OPERATIONS

Services Have Well-Defined Interfaces

Web services have strongly defined relationships but are flexible in the way they are implemented, which enables IT systems to take advantage of the capabilities of existing systems and be ready for future changes. A management application typically requires extensive customization in order to configure and monitor a typical application because the interfaces are often inconsistent and proprietary. On the other hand, Web services in an SOA provide a well-defined, standards-based interface. The interface tends to be at a consistent level of granularity and focused on one particular task.

Web services always have an interface defined by the Web Services Description Language (WSDL). By describing interfaces in WSDL, services have moved to a more dynamic and flexible interface system than the older Interface Definition Language (IDL). Management applications can therefore monitor traffic in and out of a service with a WSDL interface, and detect faults or event conditions in a consistent manner. Management applications can also provide policy-based mediation of messages between the client and the provider without extensive customization for proprietary, application-specific interfaces.

SOAs, when implemented correctly, provide loose coupling of services. The benefit of a loosely coupled system lies in its agility and the ability to access data regardless of where it is in the infrastructure. Services can look up other services using a standards-based registry, such as a Universal Description, Discovery and Integration directory (UDDI), rather than using hard-coded locations. The standards-based registries also serve as repositories for the service

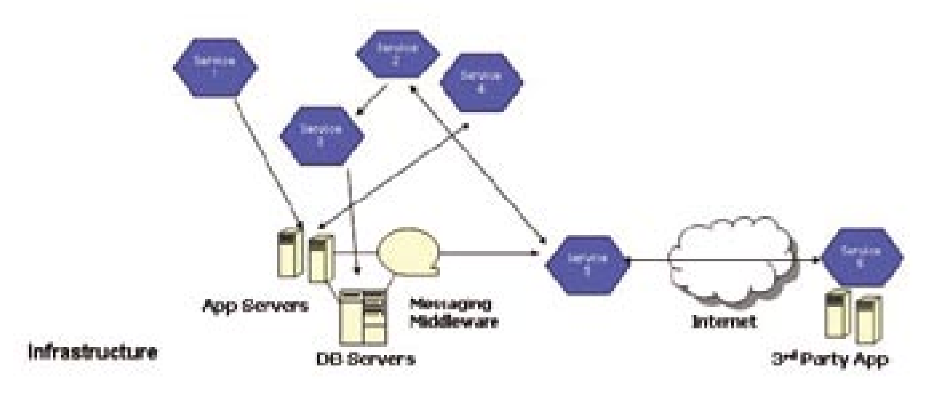


FIG 2: DEPENDENCIES

interface definitions, such as WSDL files. Management tools can leverage these standards-based directories to discover the services and their interfaces, again without extensive customization for the particular proprietary format of specific applications. Provisioning of a service into an SOA is also easier and more consistent because advertising a new service, or a new instance or version of a service, is now standards-based.

> **“The benefit of a loosely coupled system lies in its agility and the ability to access data regardless of where it is in the infrastructure”**

Services Have Well-Defined Relationships

An SOA can be designed in several ways. A customer or vendor may choose to formally model business processes or tasks, and then develop the tasks into flows or compositions of services. Alternatively, the developer may choose to define the service architecture first, and then implement the services.

As the developer creates and implements the service architecture, dependency relationships between the services are usually created. Development tools typically store relationship information in metadata associated with the services, such as deployment descriptors. Since services have well-defined relationships, management applications can use the dependency information to detect where the problem is and how to fix it. For example, in Figure 2, Service 2 has a dependency on Service 3. The management tools can reasonably infer that if Service 2 is having a problem, a problem with Service 3 may be the root cause or vice versa. If both Service 2 and Service 3 are reporting problems, the management tool would recommend that Service 3 be fixed first, since the problem with Service 2 is likely a side effect of the problem with Service 3.

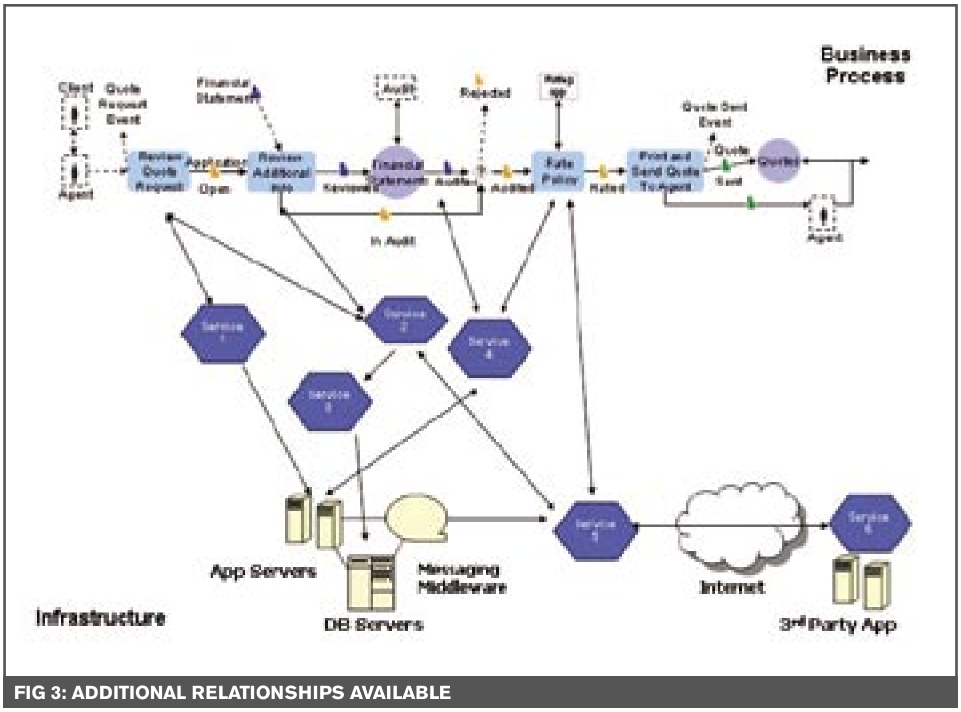


FIG 3: ADDITIONAL RELATIONSHIPS AVAILABLE

The management tools can also use the deployment and configuration information to understand the dependencies between services and the infrastructure, such as application servers and databases, on which they depend. Management tools can use this information to take automatic or manual corrective actions, such as restarting an application server, failing over to a backup service, or provisioning a new instance of a service to add capacity.

If the developer uses a formal process choreography tool, such as a Business Process Execution Language (BPEL) editor, the management tool will have even more information about the service relationships. The management tool can match the observed behavior with the expected behavior as defined by the BPEL flow, and identify and report problems or discrepancies.

If the developer also formally models and documents the business processes using a tool such as WebSphere Business Integrator Modeler, additional relationships may be available (see Figure 3). Management tools can use these relationships to report which business processes will be impacted if a service or infrastructure component reports a problem.

SOA Uses Standards-Based Communication

SOAs provide standards-based communications between the ser-

Ignorance isn't bliss.

Knowing about performance problems before they happen is.

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

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

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

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

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

diagnoSys

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

Visit: www.hwcs.com/wsj19.asp

H&W

www.hwcs.com | 1.800.338.6692

Bridging the Enterprise Computing Gap For 25 Years vices using an ESB. The ESB provides the connection infrastructure for communications between services implemented on different technologies, and provides a common set of on-ramps, off-ramps, and control points. An ESB is not a single product. It is generally implemented using many different messaging technologies, including WebSphere MQ, WebSphere Application Server, and WebSphere Web Services Gateway.

As shown in Figure 4, the ESB provides standards-based control points for many management capabilities. Security capabilities such as authentication, authorization, and encryption are implemented in the infrastructure using standards such as WS-Trust and WS-Security. Standards-based management instrumentation in the ESB allows monitoring and control of the availability and performance of the services, and the infrastructure on which they rely. Standards organizations are defining standard Web services interfaces, such as the Web Services Distributed Management being developed in OASIS for instrumentation of disparate resource types, so that management applications can more quickly manage new resource types without requiring extensive and time-consuming customization.

SOAs Are More Closely Aligned with Business Processes

As previously mentioned, a services architect can use the business process models to guide the service architecture. Since SOAs are more closely aligned with business processes, services can be developed to directly implement part or all of business process steps. Services can be built into a composition or workflow if necessary, to create the proper level of abstraction to implement a business process flow step. These composites or workflows of services can be given service interfaces.

The granularity and focused purpose of services are typically much easier to associate with business process steps than application tasks. Applications often assume context or have interfaces that are proprietary or poorly defined. Often a service integrator can write a small snippet of code to integrate two services, but may need to write extensive, fragile, and non-portable scripting or coding to integrate tasks from two different applications.

Since services are easily composed into business process steps, the relationships shown in Figure 3 are much easier to identify and to monitor. The services can be monitored for availability, or tuned for performance or to meet business needs, whereas a specific task of an application often cannot be managed individually. The management tools can therefore more easily and accurately assess and report the business process impact of a problem in the services layer or the IT infrastructure. Management tools can also use information about the business importance of various business processes to prioritize which services or IT problems should be addressed first.

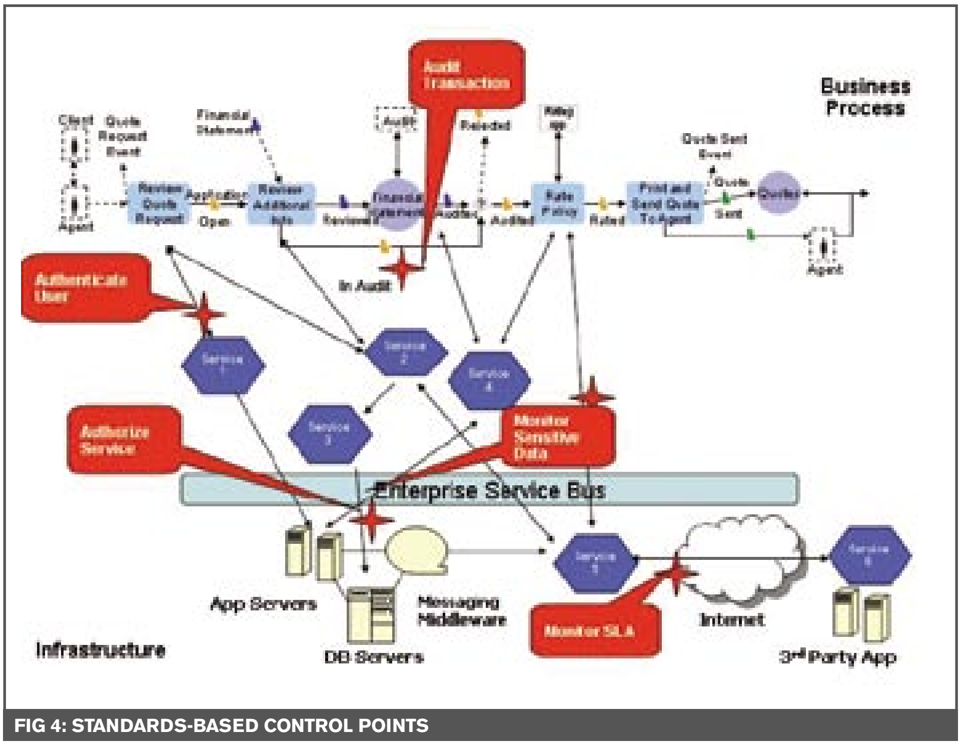
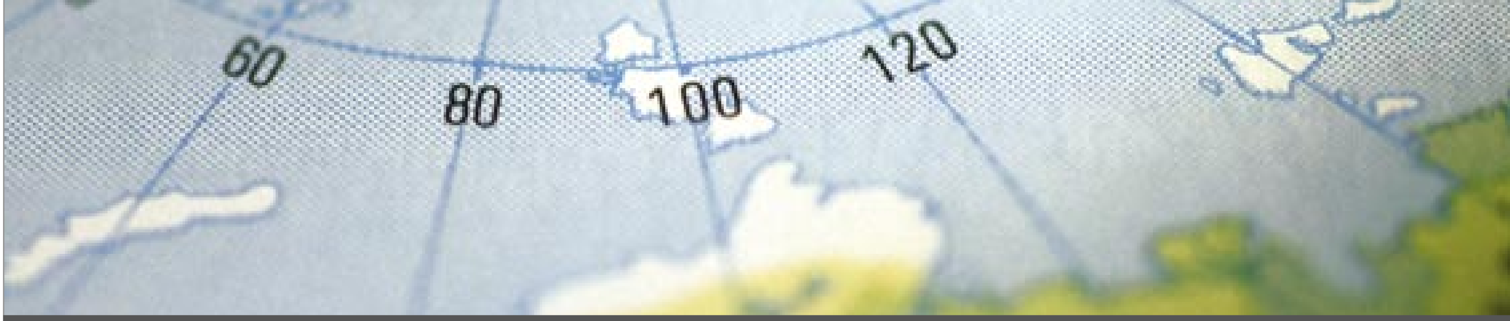


FIG 4: STANDARDS-BASED CONTROL POINTS

Summary

SOAs introduce standards, consistency, and control points into the IT environment that were difficult and time-consuming to create in the traditional distributed application environment. Management tools can leverage these standards and control points to provide better deployment, monitoring and control, and security than ever before. The bottom line is that customers are after flexibility, efficiency, and the ability to respond quickly to their market opportunities and the demands that their customers place upon them, thus allowing them to become more competitive. How well a company manages their IT infrastructure is the key to driving down costs and increasing responsiveness to changes in business. SOAs provide distinct capabilities that ease the management of the IT infrastructure.



Exchanging Trade Information Among Mercosul Member Countries

A case study

BY PAULA G. DANTAS & ANDRÉ TOST

Paula G Dantas is an IT specialist at IBM Brazil working in the Software Group. She has been working with WebSphere products since 2001; her main focus is in the business integration area. Paula provides technical sales support to customers using WebSphere Business Integration tools such as WebSphere Business Integration Server Foundation and WebSphere Studio Application Developer Integration Edition. Before working with WebSphere, she was responsible for developing Lotus Notes applications for the Latin America Software Group team.
paulinha@br.ibm.com.

Mercosul (or Mercosur) is a trading zone among Brazil, Argentina, Uruguay, and Paraguay founded in 1991. Its purpose is to promote free trade and the movement of goods and people, and skills and money between these countries. The four member countries combined represent the fourth largest economy in the world.

As part of their cooperation, the member countries need to exchange data on imported and exported products. In a first attempt, Argentina, Brazil, and Uruguay tried to exchange information using magnetic tapes. It didn't work out, due to the difficulty in matching the operations and the different types of registration of each system in a single, consistent way.

In the second attempt, Brazil and Argentina connected their respective host systems by establishing an active connection to exchange data but this solution had issues by not being reliable and also just supporting batch-mode interactions. Moreover, another big problem was that it was not possible to exchange some detailed information between the systems.

Each country was running a different application on a different platform and was challenged to exchange data between these different systems. As a requirement, the data exchange should be easily accessible with precise or selective searches, avoiding unnecessary interchanges with large amounts of data. It was also important to control access to the system for security reasons.

The Brazilian Ministry of Trade has its data stored in an Adabas database in a mainframe. Most of the applications are developed in Natural, while some Java applications running in WebSphere Application Server 4.0.1 for z/OS exist as well.

Argentina uses Microsoft IIS, Microsoft Visual Basic .NET, and an Oracle database. Paraguay uses Linux with Java and Apache Tomcat with an Oracle database running on AIX. Uruguay runs Linux with Java and Genexus.

Serpro is a federal organization for income tax data processing in Brazil and it was chosen to represent Brazil in the Mercosul project. In 2002, a group of Serpro architects went to an IBM developers' conference and heard about "Web services" and immediately realized the potential for this technology to be used for the Mercosul project. Web services technology promised to allow application interaction based on standardized XML messages across platforms and programming languages, which made it a good fit for the challenge at hand.

The First Iteration

The first step was to connect the Brazilian and Argentinean systems. Both systems would act as a service provider and a service consumer. They used different Web services environments and, back in those days, interoperability problems were common. Hence, the decision was made not to create a common, shared Web Services Definition Language (WSDL) definition of the service, but to let each side define their own service description. The only thing that was formalized was the structure of the data that was sent back and forth.

One of the first questions that had to be answered by the Serpro team was how to expose the functionality that was developed in Adabas as a Web service. Looking at the existing support for Web services in WebSphere, it became clear that a JavaBean proxy was the right answer. As soon as a Java front end to the Adabas application was in place, standard tooling in WebSphere Studio could be used to generate all the required artifacts for the Web service. The JavaBean would run inside the application server on the mainframe and access the back end. The application server would receive the incoming simple object access protocol (SOAP) message, parse it, and invoke the JavaBean.

The JavaBean was developed and WebSphere Studio took care of the rest. At the time, WebSphere used the Apache SOAP engine for its Web services support. The resulting WSDL file was sent to Argentina to be used there to create the respective client code.

The same happened the other way around. The team in Argentina created a Web service based on the .NET framework, and a WSDL file describing that service was sent to Brazil. The Serpro team used WebSphere Studio tooling to generate a proxy for the service and created a Web application around it.

If you have followed the evolution of Web services technology over the past few years, you will have heard about differences in encoding and invocation styles between Web services from different vendors. The Java community preferred "RPC" style services with so-called "SOAP encoding," whereas Microsoft favored "document" style services with "literal" encoding. The respective WSDL files from Brazil and Argentina reflected this, and some changes had to be made to both to make it work on either end. These and other problems related to interoperability are all resolved today between the major Web services vendors, but more about that later.

Listing 1 shows a subset of the WSDL file for the service that is currently offered by Brazil.

Serpro also added some extensions to the code to allow for logging and monitoring of their service. Access to the service was secured by running the connection over a virtual private network (VPN). After some testing, the solution went into production in both countries. The typical transaction rate of the system is very low, so performance and throughput are not yet big concerns. After some time, Paraguay and Uruguay joined the project and developed their own versions of the Web service. They are also in production today, running on Linux, as we had described earlier.

Figure 1 shows the architecture of the overall system.

What's Next?

Currently, Serpro is in the process of migrating towards WebSphere Application Server v5.0.2 on zOS. The Web services engine in this release is a new, high-performance IBM SOAP engine supporting message exchange across both HTTP and JMS. It leverages an XML parser that is optimized for SOAP parsing, giving it significantly better performance than other Web services engines.

The new version allows Serpro to start using standards like JAX-RPC (http://java.sun.com/xml/jaxrpc/), which is the new standard API for programming Web services in Java, and JSR-109 1.0, which is the new J2EE deployment model for Java Web services.

Among other things, JAX-RPC defines a mechanism with which it is possible to manage service invocations by intercepting request and response messages without having to change the actual service consumer or provider. In J2EE, handlers can be configured in a deployment descriptor, without writing any code, providing you with a powerful way of controlling SOAP messages as they pass through your system. For example, using JAX-RPC handlers it is possible to implement logging and other management features. Serpro will utilize this to identify and log client access to the system. Developing JAX-RPC handlers for this allows them to keep management code separate from business logic.

WebSphere Application Server v.5.0.2 also supports the "WS-I Basic Profile" (http://ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html), which will all but eliminate interoperability problems with other Web services engines. WS-I is an open industry organization chartered to promote Web services interoperability across platforms, operating systems, and programming languages. The organization works across the industry and standards organizations to respond to customer needs by providing guidance, best practices, and resources for developing Web Services solutions (www.ws-i.org).

As we had mentioned earlier, the countries participating in the project only share a common data schema describing the exchanged message structure. Given the progress that was made around interoperability, it

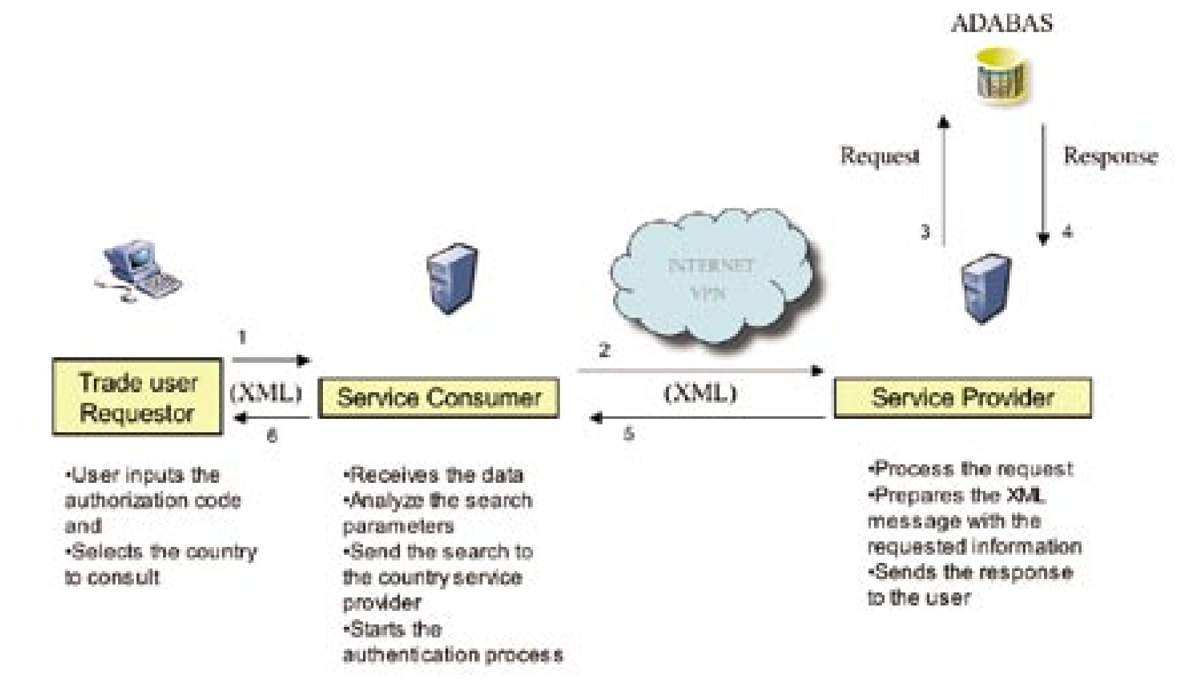

ADABAS
Request
Response
3
4
INTERNET
VPN
1
2
Trade user Requestor
(XML)
Service Consumer
(XML)
Service Provider
6
5
•User inputs the authorization code and
•Selects the country to consult
•Receives the data
•Analyze the search parameters
•Send the search to the country service provider
•Starts the authentication process
•Process the request
•Prepares the XML message with the requested information
•Sends the response to the user

FIG 1: SYSTEM ARCHITECTURE

André Tost is a senior technical staff member in the WebSphere Business Development group where he helps IBM's Strategic Alliance partners integrate their applications with WebSphere. His special focus is on Web services technology throughout the WebSphere product family. Before his current assignment, André spent 10 years in various development and architecture roles in IBM software development, most recently for the WebSphere Business Components product. Originally from Germany, he now lives and works in Rochester, Minnesota. atost@us.ibm.com.

should now be possible to share the entire WSDL definition across systems with only the respective endpoint addresses being different. For example, WebSphere Studio now follows the WS-I recommendation to create "document/literal" Web services by default, thus allowing sharing between the exact same SOAP binding and the .NET system.

Moreover, this release of WebSphere Application Server supports the WS-Security specification, which describes enhancements to SOAP messaging to provide quality of protection through message integrity, message confidentiality, and single message authentication. While there is no immediate plan to deploy the Mercosul services with WS-Security enabled, it becomes an option for offering the service outside the VPN, improving overall accessibility of the system. Note, however, that Apache SOAP does not support WS-Security at this time, so that the systems in Paraguay and Uruguay, both of which utilize Apache SOAP, would have to be upgraded.

Summary

The Mercosul trade system is a great example for the potential of Web services technology. It connects systems running on a variety of platforms, developed in a variety of programming languages. Other countries could join the project at a later time simply by providing and consuming the same services, regardless of implementation. Serpro could utilize their existing mainframe system by developing and running their service implementation with WebSphere Studio and WebSphere Application Server. As new standards evolve, they can start expanding on the quality of service characteristics of the overall system, beginning by adding support for WS-Security.

LISTING 1

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="ServiciosMercosur"
    targetNamespace="http://mercosur.wsdl/
ServiciosMercosur/"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:format="http://schemas.xmlsoap.org/wsdl/formatbind-
ing/"
    xmlns:java="http://schemas.xmlsoap.org/wsdl/java/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://mercosur.wsdl/ServiciosMercosur/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:
xsd1="http://mercosur/">
    <import location="" namespace="http://mercosur/"/>
    <types>
        <schema attributeFormDefault="qualified"
            elementFormDefault="unqualified"
            targetNamespace="http://mercosur/"
            xmlns="http://www.w3.org/2001/XMLSchema"
            xmlns:soapenc="http://schemas.xmlsoap.org/
soap/encoding/"
            xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd1="http://mercosur/">
            <import namespace="http://schemas.xmlsoap.
org/wsdl/" schemaLocation="http://schemas.xmlsoap.org/
wsdl/"/>
            <import
                namespace="http://schemas.xmlsoap.org/
soap/encoding/" schemaLocation="http://schemas.xmlsoap.
org/soap/encoding/"/>
            <complexType name="Subregimen">
                <all>
                    <element name="codigoSubregimen"
nillable="true" type="string"/>
                    <element name="codigoSubregimenDesc
ripcion"
                        nillable="true" type="string"/>
                </all>
            </complexType>
            <complexType name="ArrayOfSubregimen">
                <complexContent>
                    <restriction base="soapenc:Array">
                        <sequence/>
                        <attribute ref="soapenc:array-
Type" wsdl:arrayType="xsd1:Subregimen[]"/>
                    </restriction>
                </complexContent>
            </complexType>
            </schema>
    </types>
    <message name="ws_ConsultaDestinacionRequest">
        <part name="numeroDeclaracion" type="xsd:
string"/>
    </message>
    <message name="ws_ConsultaDestinacionResponse">
        <part name="result" type="xsd1:DeclaracionDeExpo
rtacion"/>
    </message>
    <portType name="ServiciosMercosur">
        <operation name="ws_ConsultaDestinacion" paramete
rOrder="numeroDeclaracion">
            <input message="tns:ws_ConsultaDestinacionReq
uest" name="ws_ConsultaDestinacionRequest"/>
            <output message="tns:ws_ConsultaDestinacionRe
sponse" name="ws_ConsultaDestinacionResponse"/>
        </operation>
    </portType>
    <binding name="ServiciosMercosurBinding" type="tns:
ServiciosMercosur">
        <soap:binding style="rpc" transport="http://sche-
mas.xmlsoap.org/soap/http"/>
        <operation name="ws_ConsultaDestinacion">
            <soap:operation soapAction="" style="rpc"/>
            <input name="ws_ConsultaDestinacionRequest">
                <soap:body
                    encodingStyle="http://schemas.xml-
soap.org/soap/encoding/"
                    namespace="mercosur" parts="numeroDe
claracion" use="encoded"/>
            </input>
            <output name="ws_ConsultaDestinacionRespons
e">
                <soap:body
                    encodingStyle="http://schemas.xml-
soap.org/soap/encoding/"
                    namespace="mercosur" use="encoded"/>
            </output>
        </operation>
    </binding>
    <service name="ServiciosMercosurService">
        <port binding="tns:ServiciosMercosurJavaBinding"
name="ServiciosMercosurJavaPort">
            <java:address className="mercosur.
ServiciosMercosur"/>
        </port>
    </service>
    <service name="http://mercosur.wsdl/
ServiciosMercosur/">
        <port binding="tns:ServiciosMercosurBinding" name
="ServiciosMercosurPort">
            <soap:address location="…"/>
        </port>
    </service>
</definitions>
```

More than a help desk

IT PowerPAC WS from Eden

BY JAY JOHNSON

Jay Johnson is a consultant for the WebSphere software group at IBM.
jay2ee@hotmail.com

If you've read my column before, you've mostly seen reviews of software development tools for WebSphere. This column is a bit of a departure from the usual. All of us have experienced calling a support desk for help, and afterward feeling more frustrated than before the call. Support people struggle with limited, uncorrelated information and slow, unreliable tools.

The main thing missing is the ability for the responder to efficiently correlate existing information pertaining to the user's problem. In almost all cases, customers must call in to report the problem and then sit through additional calls to find a solution. Major angst can build up on both ends of the phone line, and once led me to do something kind of radical.

I joined forces with three other software developers about 10 years ago and started a company. Our goal was to create and sell a Web-based help desk. Of course it consisted mostly of static pages with batch-processed content. Our Java app generated new pages overnight from a defect tracking database and allowed CSRs to enter all the information that managers required. Unfortunately, our Web application never caught on.

We thought we failed to capture the market because we were ahead of our time. A short time later, we concluded that the Internet was not a suitable medium to host our sort of application since a thin-client solution could never compete against client/server apps that did the same things in a much more user-friendly way. Eden Communications, Inc. however, has now proven that we were wrong in deducing our reasons for failure.

IT PowerPAC WS Service Management Suite (ITPP WS), a thin-client help desk by Eden, has an impressive user interface by any standards. Even more impressive to WebSphere fans is that the system was created using WSAD and runs on either WAS 5.1 or WAS Express. It also runs compatibly with all the major database systems including MySQL.

IT PowerPAC WS goes far beyond the rudimentary Web-based help desk we created a decade ago. It is actually five related applications integrated into one. The entire system has a consistent look and feel and information can be shared smoothly between applications.

The first application is an incident manager, otherwise known as a help desk (see Figure 1). From here, users can log and respond to help tickets from customers. Help tickets submitted by phone or e-mail can be routed to an appropriate specialist or team, or automatically solved at first contact.

The incident management page is a gateway to the other functions of the system and facilitates automatic e-mail notification to technicians or teams, routing, reassignment, and escalation of a problem. In the course of these actions, database elements pertaining to resource and asset management, along with problem management, are updated.

The next tab on the main page takes a user to the problem management function. Here users can track defects as well as initiate and track change requests/upgrades. This functionality provides an integrated solution for the Quality Assurance crew since fixes and upgrades can be routed for QA verification.

Other tabs on the incident management page are asset management where users track location and status of software and hardware; resource management where users schedule time and people to work on fixing bugs, making enhancements, or closing out help tickets; and a knowledge base that includes bulletins, SLAs, reference material on products and services. In addition, the system includes role-based administration functions that allow extensive system customization.

The key to making any system comfortable for users is supporting reconfiguration to fit needs and preferences. ITPP WS goes beyond allowing customization – the system encourages it. Each user can customize his or her personal dashboard, as shown in Figure 2. They can choose the information they want to display and where the information is dis-

Eden Communications, Inc.
517 Broadway
Saratoga Springs, NY
12866
sales@eden.com
518-580-0760
800-804-8393
www.eden.com

played on the dashboard.

ITPP WS was built to be flexible, so the initial setup might be considered intensive because there are many choices that have to be made by the administrator before the application is ready for production. The tool addresses this by leading users through the initial process and also providing an Administrator's Guide. In addition, Eden (of course) offers on-site training.

If users are not satisfied with any of the many (150+) canned reports available, they can use the built-in report editor to create new ones, as shown in Figure 3. Users can specify the data items to view along with sorting and grouping.

In the admin view, users with sufficient privilege can load user records from an existing lightweight directory access protocol (LDAP) connection (see Figure 4). An administrator can customize displays and forms, add customers, vendors, companies, technicians, and products supported. All drop-down lists, such as severity, priority, or defect are configurable. Users can also add services and products to be supported. All data entry forms are immediately validated, with the cursor moving automatically to the erroneous field.

Not only does IT PowerPAC WS have a consistent look and behavior, it automatically tracks relationships between the main functions. For example, an incident can be related to a software or hardware element and mesh with asset management. Assets such as hardware loaners, down systems, software patches, and licenses can all be tracked in relation to incidents and resources

The knowledge base tab facilitates searching documents related to software products, versions, patches, and fixes, as well as Service Level Agreements. To help technicians solve and route help tickets, the system presents detailed individual or company profiles including a history of problems solved and correspond-

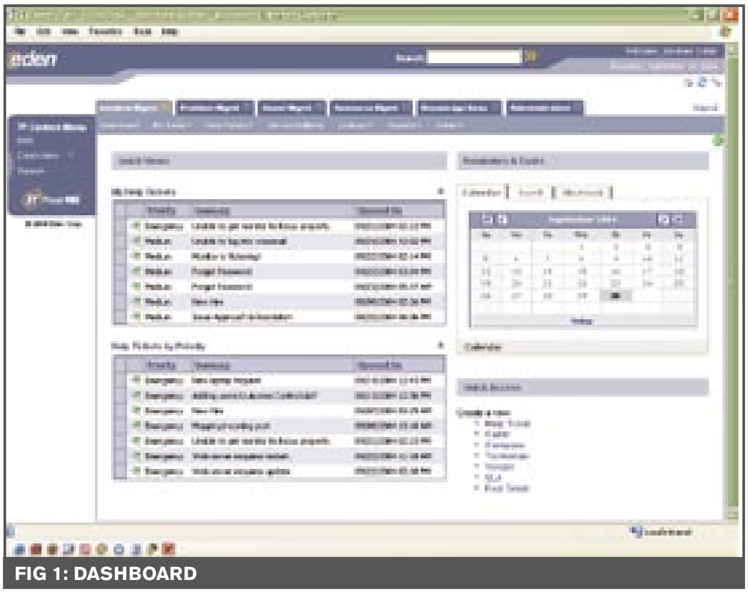
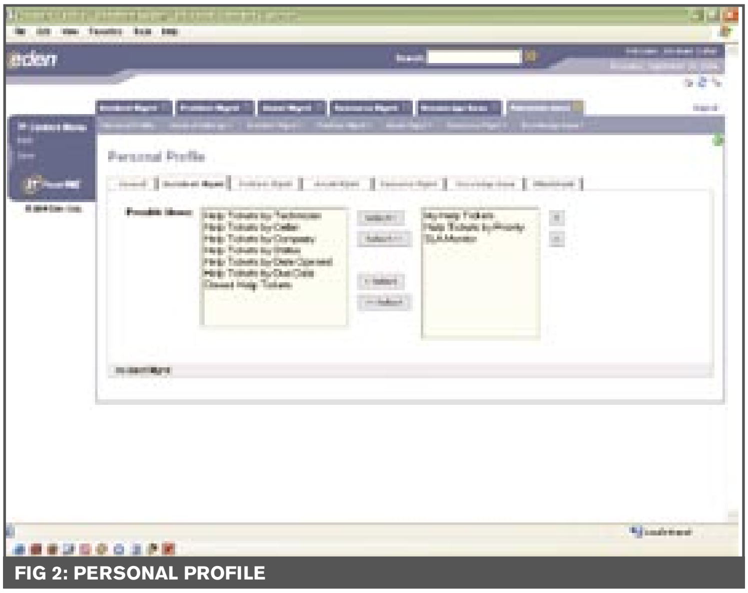


FIG 1: DASHBOARD



Personal Profile

Possible Views:

Help Tickets by Technician
Help Tickets by Caller
Help Tickets by Company
Help Tickets by Status
Help Tickets by Date Opened
Help Tickets by Due Date
Closed Help Tickets

My Help Tickets
Help Tickets by Priority
SLA Monitor

FIG 2: PERSONAL PROFILE

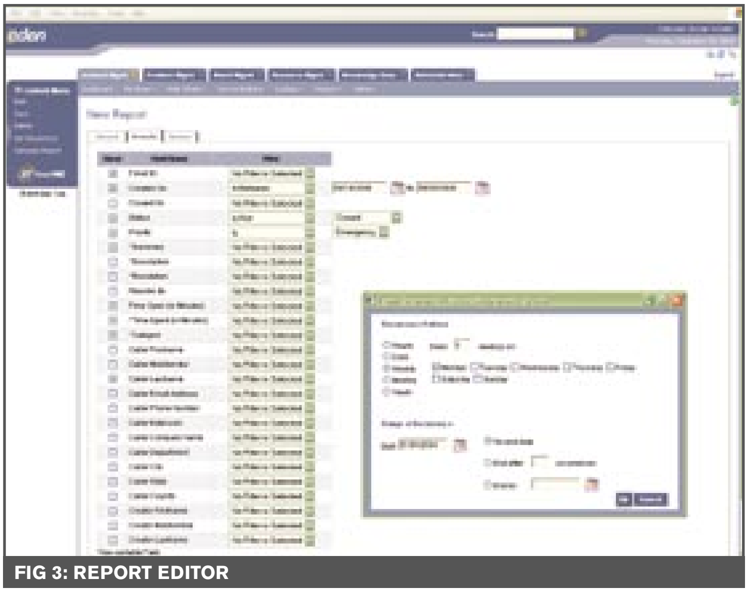


FIG 3: REPORT EDITOR

ing help tickets. Techs can immediately reference all SLAs relating to a ticket and take appropriate action.

Most of us have experienced calling customer support, and after a significant wait, find out that the CSR answering the call cannot solve our problem, but offers to have a more expert CSR call later. The sad thing is, this is often unnecessary. It has been proven that 40 percent of problems can be fixed at triage since fixes have already been formulated for similar problems at least once before, or solutions can be looked up easily in product documentation.

This is where ITPP WS's Auto Solve feature comes in (see Figure 5). Keywords in the problem description are automatically searched to find similar problem descriptions that have already been solved. This makes CSRs look brilliant and customers happier.

It is an interesting fact that 20–40 percent of problems are either so well known or so simple that they are solved without an incident report being opened, yet the statistics for these incidents need to be known for management purposes. To overcome this problem ITPP WS provides Fast Tickets that can be generated immediately via defined templates (see Figure 6).

IT PowerPAC WS's knowledge base is well integrated with the other features in the system. Detailed customer information, including Service Level Agreements, is immediately available for users seeking to solve customer problems. The knowledge base also supports an FAQ and bulletins, along with reference material on products and services. Service bulletins are automatically searched and each can have a distribution list associated with it.

From a management perspective, IT PowerPAC WS tracks progress on a help ticket and provides a non-editable audit trail of tasks and activities. In addition, there are a number of statistical reports available as well as the ability to create custom reports. Managers will also like the system's ability to track the time associated with solving a problem.

Eden plans to build a WebSphere Portal–based version for April 2005. Other features planned are support for IP telephony and additional IT project management. They will also be enhancing the product's current project tasks management by surfacing resource data through Gantt charts and providing the ability to do resource-leveling.

IT PowerPAC WS is a pure J2EE application and can run on top of DB2, MYSQL, Oracle, or SQL Server. Security is handled via JAAS. Beyond being a very nice WebSphere-based IT services desk, IT PowerPAC WS is a fine example of what WebSphere technology can do to create a thin-client application that is as user friendly as a thick-client UI.

This, however, is not the reason service reps and technicians will like the product. They will like it because Eden has done something my partners and I didn't know how to do a decade ago: they made the help desk a problem-solving assistant rather than simply management overhead.

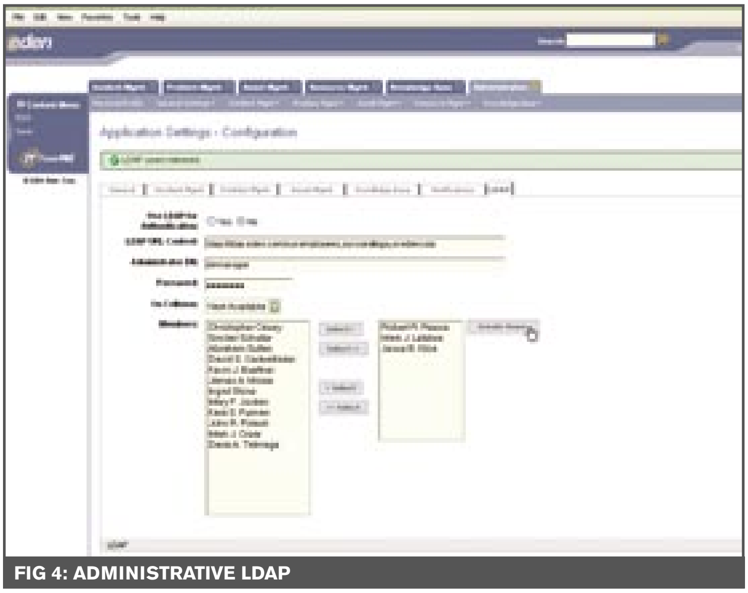


FIG 4: ADMINISTRATIVE LDAP

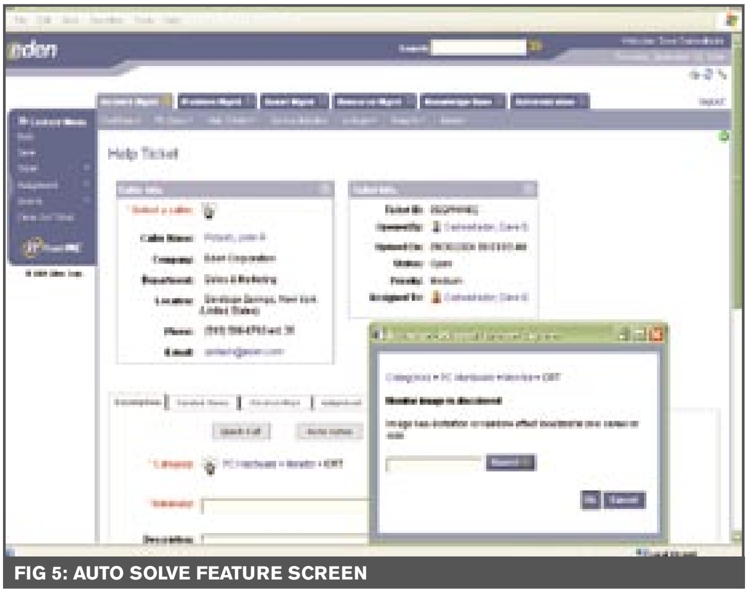


FIG 5: AUTO SOLVE FEATURE SCREEN

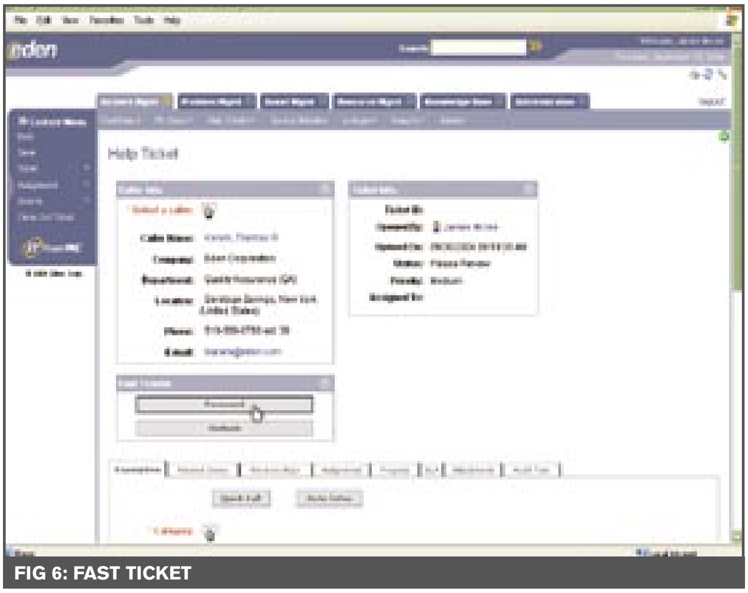
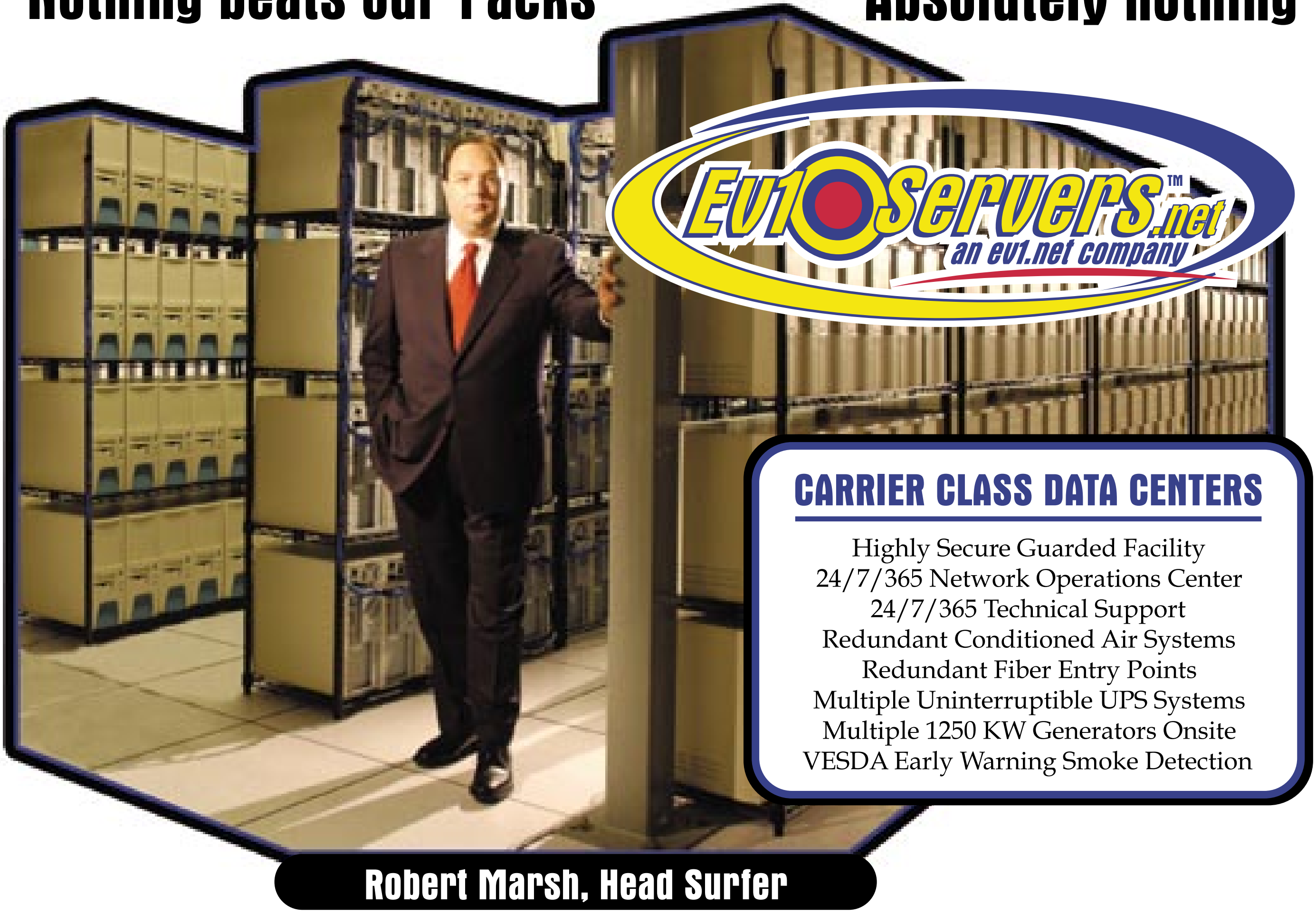


FIG 6: FAST TICKET


Ev1Servers.net™
an ev1.net company

Deploying WAS Applications Using Maven

A simple and powerful build tool

BY BRENT WORDEN

Brent Worden is a senior technical consultant at Perficient, a firm recognized for its expertise around the WebSphere suite of products. He has been developing software for nearly 10 years and developing J2EE applications for more than five years. For more information, please visit www.perficient.com.

brent.worden@perficient.com

When people talk about scalability, they are usually referring to how well an application performs with an ever-increasing load. Another type of scalability that needs attention during an application's life cycle is build scalability, or how well an application's build and deployment scales with ever-increasing complexity and components.

Maven is a build tool from Apache that addresses build scalability. Maven uses simple project descriptors and a highly extensible and open architecture to build, deploy, and release application components.

Through project descriptors, new components can be added to an application's build with little effort from a developer. Once added, Maven determines the component's placement in the application build order through its dependencies, and builds, deploys, and releases the new component and application bundle accordingly.

This article is not intended to be an extensive guide to using Maven and it assumes the reader has a working knowledge of Maven. For more in depth information about Maven's capabilities, visit http://maven.apache.org. Also, a good introduction to Maven's project descriptors and building a generic J2EE application with Maven is found in Charles Chan's article "Project Management: Maven Makes it Easy."

Benefits of Using Maven for WAS Deployment

With the tight integration between WebSphere Application Server (WAS) and WebSphere Studio Application Developer (WSAD), you may question the need to use an external tool to build and deploy J2EE applications to WebSphere. WSAD is a powerful, full-service tool that is capable of building J2EE applications and deploying them on WAS. However, there are some reasons for using Maven to supplement the building and deployment services provided by WSAD:

- Maven makes build independent of a developer's IDE. Developers are creatures of habit and prefer to work with the coding environment in which they are most comfortable and most productive. By using Maven to build and deploy applications, developers are free to use any IDE they choose, which should result in more efficient developers.
- Maven can easily be incorporated into a continuous integration environment. The frequent running of unit and integration tests and timely feedback afforded by continuous integration is crucial for team development. Maven builds are already supported by many continuous integration tools, including CruiseControl and Anthill.
- Maven runs on any environment with a J2SE SDK. Sometimes the target environment hosting WAS is not supported by a developer's chosen IDE. Even with the "write once, run everywhere" promise of Java, portability problems arise when Java applications are written on one environment and installed on another. Since Maven can run on every environment supported by WAS, J2EE applications can be built, tested, and deployed on the same environment hosting WAS, mitigating the chance portability issues go undetected.
- Maven can handle the most complex of build scenarios. As applications and systems become more complex and integrations between disparate enterprise resources become commonplace, it is essential that an application's build can be maintained easily and efficiently. Through a series of plug-ins, Maven is capable of interfacing with many commonly used, development resources. These plug-ins include support for testing through JUnit, source generation using XDoclet, static analysis via JDepend, interaction with Web servers such as Tomcat, plus many more. In addition to the existing Maven plug-ins, if an application's build needs to interface with a resource unsupported by Maven, plug-ins can easily be created using scripting, Java programming, or a combination of the two.

Installing the Needed Software

To illustrate how Maven can be utilized to deploy to

WAS, a simple J2EE application will be deployed. The application will be bundled and deployed as an EAR and will consist of a utility JAR, an EJB JAR, and a WAR. For the purposes of this exercise, the development and deployment environment will consist of:

- WebSphere Application Server 5.1
- IBM JDK 1.4.1 (included with WAS install)
- Maven 1.0
- WebSphere 5.0/5.1 Plug-in 1.2

INSTALLING WAS

It is assumed that WAS is installed and working correctly. If WAS is not installed, run the WAS install wizard choosing the full installation.

Once WAS is installed, set the JAVA_HOME environment variable to $WAS_HOME/java, where WAS_HOME is the root directory of the WAS installation. This setting is required. The subsequent WAS deployment only works using IBM's JDK.

INSTALLING MAVEN

To install Maven, it's best to follow the Installing Maven instructions.

Once Maven is installed, create or edit the $HOME/build.properties file, where HOME is the user's home directory. In this file, add the following line:

```
maven.was5.home = $WAS_HOME
```

again, where WAS_HOME is the root directory of the WAS installation.

Of note, an Internet connection is required by Maven in order to download any needed dependency JARs. If the connection is through a proxy, additional proxy settings must be added to the $HOME/build.properties files.

INSTALLING WEBSPHERE PLUG-IN

The WebSphere 5.0/5.1 plug-in is available from the Maven Plug-ins project. Using this plug-in makes deploying J2EE applications to WAS extremely easy and provides a highly reusable approach to deploying many applications efficiently. To install the plug-in, use Maven itself. Simply enter the following command on a single line at a command prompt:

```
> maven plugin:download
  -Dmaven.repo.remote=http://maven-plugins.source-
forge.net/maven/
  -DgroupId=maven-plugins
  -DartifactId=maven-was5-plugin
  -Dversion=1.2
```

INSTALLING THE SAMPLE J2EE APPLICATION

The entire application is contained in the archive. To install the application, simply unarchive the files into any directory. This directory will be referred to as ROOT in the remainder of this article.

The Sample Application

PROJECT LAYOUT

In the situation of a multicomponent project, such as this J2EE application, it is easiest to divide the project into many modules with each one responsible for building a single component. Then have a top-level project controlling the building of the modules.

Each of the modules, as well as the top-level project, is a Maven project. Each of these Maven projects has the same general setup:

- A required Maven project descriptor (a project.xml file) that defines project dependencies, source files, and other metadata
- An optional Maven build script (a maven.xml file) that customizes the build for a project
- Optional project-specific properties (a project.properties file) used to customize the build setting for a project.

UTILITY JAR MODULE

The sample-util module (located at ROOT/modules/sample-util) creates a simple, utility JAR that might be shared across many application components. Setting up the sample-util JAR module is straightforward and involves defining the build information in the project descriptor as well as defining the target goal in the Maven build script.

Of note in the project descriptor is the sourceDirectory element. It defines the directory (relative to the sample-util directory) containing all the Java source files to be compiled into class files. These class files will be bundled into the sample-util JAR.

The Maven build script is as simple as it can get. It contains a single build goal, which is also defined as the default goal:

```
<!-- Install jar into local repository -->
<goal name="build" prereqs="jar:install"/>
```

The jar:install goal is the prerequisite goal because it is necessary for the JAR to be in Maven's local repository in order to build other subprojects depending on sample-util.

EJB JAR MODULE

The sample-ejb module (located at ROOT/modules/sample-ejb) creates an EJB JAR replete with all Java classes, deployed code, and EJB bindings needed for WAS deployment. Setting up this EJB JAR involves using a combination of out-of-the-box Maven plug-ins and customizing the build for WAS-specific details.

The build information found in this module's project descriptor is similar to that found in the sample-util project. The only change is the addition of dependencies. The most important dependency is the sample-util entry:

```
<dependency>
  <groupId>maven-was</groupId>
  <artifactId>sample-util</artifactId>
  <version>1.0</version>
  <properties>
    <ejb.manifest.classpath>true</ejb.manifest.
classpath>
  </properties>
</dependency>
```

This dependency is crucial for a few reasons. First, it allows this project's source files to be compiled because all dependency JAR files are added to the compilation classpath. Second, it assures that Maven always builds the sample-util module before building the sample-ejb module. Last, the ejb.manifest.classpath element informs Maven to add the sample-util JAR to the classpath found in the EJB JAR's manifest, which is created by Maven.

The build script for this project is more involved than that found in the sample-util project. This is the result of the additional WAS-specific tasks that need to be completed in order to create an EJB JAR deployable on WAS.

Like the sample-util project, this project contains a build goal:

```
<!-- Install jar into local repository -->
<goal name="build" prereqs="ejb:install"/>
```

The ejb:install prerequisite goal is used to install the EJB JAR into the local repository so it too is available for building subsequent subprojects.

Prior to installing the EJB JAR to the local repository, WAS-specific deployed code must be added to the EJB JAR. This can be accomplished by attaining the was5:ejbDeploy goal provided by the WebSphere plug-in. To hook into Maven's build process, a custom post goal is defined:

```
<!-- generate deployment and rmic code for an ejb
jar -->
<postGoal name="ejb:ejb">
  <attainGoal name="was5:ejbDeploy"/>

  <!-- copy jar with deployed code over original -->
  <copy file="${maven.was5.ejbDeploy.file.out}"
        tofile="${maven.build.dir}/${maven.final.
name}.jar"/>
</postGoal>
```

This post goal is defined to run after the ejb:ejb goal, which creates the raw EJB JAR, is complete. The post goal in turn attains the was5:ejbDeploy goal and then copies the EJB JAR with deployed code over the original, raw EJB JAR.

Another aspect of creating WAS deployable EJB JARs is ensuring the EJB deployment descriptor and WAS EJB bindings are bundled with the EJB JAR. Bundling resource files inside an EJB JAR is accomplished by placing them in the src/ejb directory of this module. The src/ejb directory is significant as it is the default setting for the maven.ejb.src property. This property defines the root directory containing files to be included in the EJB JAR. When the resource files found in this directory are copied into the EJB JAR, the directory structure is preserved. Therefore, by placing both the ejb-jar.xml and ibm-ejb-jar-bnd.xmi files in the src/ejb/META-INF directory, these files are easily bundled in the EJB JAR in their correct location.

WAR MODULE

The sample-war module (located at ROOT/modules/sample-war) creates a WAR containing JSPs, servlets, and other WAR resources. Setting up the WAR module is, again, straightforward and involves correctly defining dependencies in the project descriptor as well as customizing the build script.

The build information found in this module's project descriptor is similar to that found in the sample-util module. Again, some dependencies are added to ensure correct compilation and bundling:

```
<dependency>
  <groupId>maven-was</groupId>
  <artifactId>sample-ejb</artifactId>
  <version>1.0</version>
  <type>ejb</type>
</dependency>
<dependency>
  <id>commons-codec</id>
  <version>1.3</version>
  <properties>
    <war.bundle>true</war.bundle>
  </properties>
</dependency>
```

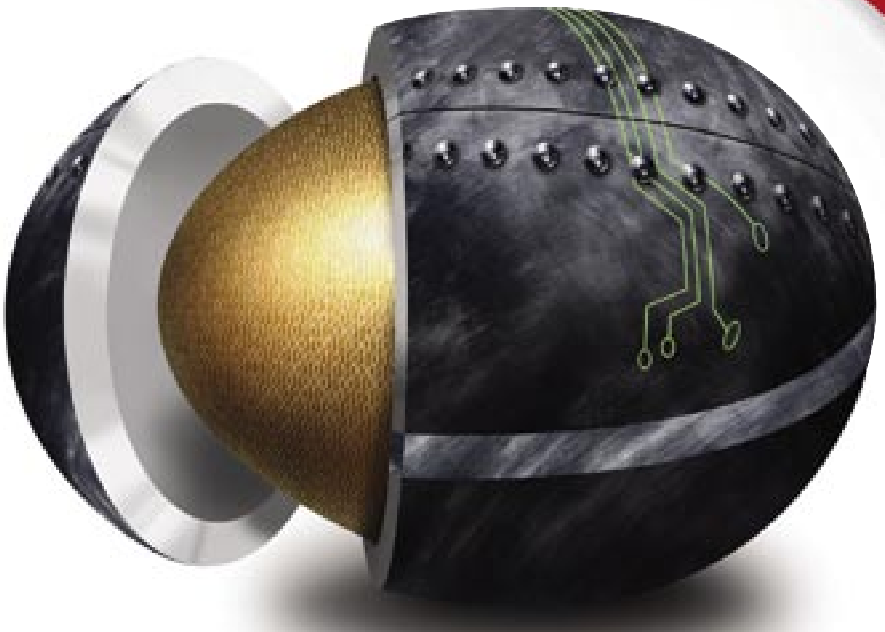
By adding the sample-ejb dependency, Maven always builds the sample-ejb module before building the sample-war module. Also, note the war.bundle element for the commons-codec dependency. This informs Maven to bundle the commons-codec JAR in the WEB-INF/lib directory of the final WAR.

Again, the build script is very simple, like that for the sample-util module. It contains the single build goal with the single prerequisite goal. This time the prerequisite is war:install:

```
<!-- Install war into local repository -->
<goal name="build" prereqs="war:install"/>
```

The war:install prerequisite is called to install the WAR into the local repository so it too is available for building subsequent modules.

As for the WAR resources, they are automatically bundled into the final WAR provided they reside in the src/webapp directory of this module. This src/webapp directory is special as it is the default setting for the maven.war.src property. This property defines the root directory housing the files to be included in the WAR. So, to include JSPs, images, deployment descriptors, and the like, simply place those resources in that directory.



Get FREE Exhibition Hall Admission

Protect your assets.

Infosecurity delivers information, education and solutions for a higher level of security.
Infosecurity heightens the sharing of information critical to a more secure and compliant information infrastructure through the delivery of a highly balanced, quality educational program for security professionals, practitioners and business leaders. Infosecurity is committed to advancing the knowledge of emerging threats and vulnerabilities, solutions to mitigate these dangers and best practices in the effort to protect both private and public networks.

Earn up to 10 CPE Credits!
(ISC)$^{2®}$ has partnered with Infosecurity to bring you the “Gold Standard” in continuing professional education. Visit us online to see the entire conference program.

Don’t miss the Storage Zone.
STORAGE ZONE
If your prime concern is the efficient management, retrieval and availability of your organization’s information, Storage Zone will provide you with access to the latest products, technologies and advice from industry leaders.

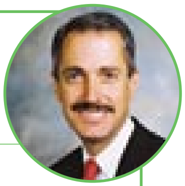
Real Risks – Real Solutions.
Information security technology is complex and ever-changing to meet continued threats. Remain apprised of what suppliers are offering. Evaluate competing and complementary products and services by visiting with leading vendors.

Sampling of participating companies: ADVA Optical Networking, Inc., Alta Associates, Astaro Corporation, Avaya Inc., Barracuda Networks, Black Dragon Software, Borderware Technologies, Canaudit Inc., Cisco Systems - UPN & Security Systems, Citadel Security Software, Credant Technologies, CyberGuard Corp., DLTtape, Endforce, Ernst & Young, Exabyte Corporation, FortiNet, GlassLock, IBM, Internet Security Systems, Inc., Juniper Networks, Lancope, Network Engines, PatchLink, PKware Inc, Protocom Development Systems, Qualys, Radware, Inc., Rapid 7, Securit-E-Doc, SonicWALL, Sophos Inc., SPI Dynamics, Spirent, SSH Communications Security, Sygate Technologies, Symantec Corporation, Tablus, Teros, Top Layer Networks, Ubizen Inc. For a complete listing of vendors visit us online at www.infosecurityevent.com.

Wednesday
Dec. 8 | 11:30am
Opening Keynote: Rudy Giuliani

New York City Former Mayor.

Thursday
Dec. 9 | 11:30am
Keynote Speaker: William Pelgrin



Director of the NYS Office of Cyber Security and Critical Infrastructure Coordination.

Register today!

For early-bird conference discounts and free exhibition admission, register online at www.infosecurityevent.com. Questions? Call 888-251-0566, or 203-840-5690. To Exhibit, please call 203-840-5387.

Conference: December 7-9, 2004 **Exhibits:** December 8-9, 2004
Jacob K. Javits Convention Center, NY www.infosecurityevent.com/websecure

infosecurity NEW YORK

Global Media Sponsor: SC
Premier Media Sponsors: CSO The Resource for Security Executives INFOSTOR
Premier Education Partner: (ISC)²
Premier Education Sponsors: LexisNexis Mealey Publications & Conferences Group
Annual Conference: ISSA
Annual Chapter Conference: ISSA New York Metro
Official Media Sponsors: Information Storage + Security Journal
Global Sponsor: symantec.
Gold Sponsors: NOKIA AVAYA
Silver Sponsors: Juniper Ernst & Young Protocom Development Systems
Produced and Managed by:

EAR MODULE

The final module is the sample-ear module (located at ROOT/modules/sample-ear). This module is responsible for bundling all other modules into an EAR and deploying it to WAS. As before, this is accomplished through project dependencies and a custom build script.

In addition to the normal setup, this module requires a couple of property settings to aid with the deployment. These properties are added to the project.properties file and are as follows:

```
maven.ear.appxml.generate = true
maven.ear.displayname     = MavenWasSample
```

The maven.ear.appxml.generate property informs Maven to generate the deployment descriptor for the application. The maven.ear.displayname property sets the application's display name. This change in name is needed because the default display name contains illegal characters and is an invalid WAS application name.

This module's project descriptor lists all the other modules as dependencies to ensure correct compilation and bundling (see Listing 1).

By adding the other module dependencies, Maven always builds the other projects before building the sample-ear module. Also, note the ear.bundle and ear.appxml.war.context-root elements for the dependencies. The ear.bundle element informs Maven to bundle each dependency's final artifact in the EAR. The ear.appxml.war.context-root is unique to WAR dependencies and specifies which context root to use for the Web application. This value is added to the EAR deployment descriptor that is generated by Maven.

Like the sample-ejb module, the build script for this module involves some WAS-specific tasks used to actually deploy and start the J2EE application.

Again, like the other modules, this module contains a build goal:

```
<!-- Install ear into local repository -->
<goal name="build" prereqs="ear:install"/>
```

Prior to installing the EAR to the local repository, the WAS-specific goals can be invoked to actually deploy the EAR. This is accomplished via a post goal set to run after the ear:ear goal, which creates the EAR, is completed. This post goal in turn attains the WAS goals to actually perform the application deployment (see Listing 2).

The post goal provides the means to either deploy the EAR (using the was5:installApp and was5:startApp goals) or reinstall the EAR (using the was5:reinstallApp) over an existing deployment. How that choice is determined will be illustrated shortly in the Rebuild the Application section.

PROPERTY	DESCRIPTION
maven.was5.host	the host to connect to
maven.was5.port	the port on the host to connect to
maven.was5.username	user ID to authenticate with
maven.was5.password	password to authenticate with
maven.was5.server	the name of the server hosting the application

TABLE 1: COMMONLY USED PROPERTIES

TOP-LEVEL PROJECT

The top-level project (located at ROOT) is responsible for building all the modules and acts as a controller over the entire build process.

The project descriptor is quite simple and contains only rudimentary metadata about the project. It is simple because the top-level project contains no source code so all the build information is omitted.

The build script is also very simple, yet very crucial to building the entire application. The default goal of the build script is build. This goal uses the Maven reactor plug-in to propagate the build goal to all of the modules contained in the application. The reactor is responsible for determining the module build order based on their individual dependencies and is responsible for attaining the build goal for each of the modules. The relevant goals are shown in Listing 3.

Deploying the J2EE Application

BUILD THE APPLICATION

With all the project descriptors and build scripts in place for the module, building and deploying the EAR is a simple exercise:

1. Ensure the WAS server is running.
2. Execute Maven from ROOT:

```
> maven
```

3. Sit back and witness the magic.

What is actually witnessed from the command line are the steps Maven takes to build and deploy the application.

First, the Maven reactor determines the build order of the modules based on their interdependencies. This is indicated by the following output:

```
Starting the reactor...
Our processing order:
sample-util
sample-ejb
sample-war
sample-ear
```

Next, the first module, sample-util, is built. This is indicated by the following output:

```
+----------------------------------------
| Executing (build): sample-util
| Memory: 9M/10M
+----------------------------------------
```

The subsequent output details all the goals attained in order to create the sample-util JAR. These goals include:

- ***java:compile:*** compiles all the source files
- ***test:test:*** runs any JUnit tests
- ***jar:jar:*** actually bundles everything into the final sample-util JAR

After sample-util is built, sample-ejb is built as illustrated by the output:

```
+----------------------------------------
| Executing (build): sample-ejb
| Memory: 10M/19M
+----------------------------------------
```

The proceeding output details the step taken to create the sample-ejb EJB JAR. These steps for the most part mirror those taken to build the sample-util JAR.

Of special note are the many lines of output involving the was5:ejbDeploy goal whose successful completion is indicated by:

```
[wasEjbDeploy] EJBDeploy complete.
[wasEjbDeploy] 0 Errors, 0 Warnings, 0 Informational
Messages
```

Next, sample-war is built, which is indicated by the output:

```
+----------------------------------------
| Executing (build): sample-war
| Memory: 17M/19M
+----------------------------------------
```

Again, a series of goal output follows detailing the steps required to build the WAR.

Finally, sample-ear is built starting with the output:

```
+----------------------------------------
| Executing (build): sample-ear
| Memory: 17M/19M
+----------------------------------------
```

Of special note in the subsequent output are the many lines of output involving the was5:installApp and was: startApp goals whose successful completion are respectfully indicated by:

```
[wasInstallApp] ADMA5013I: Application
MavenWasSample installed successfully.
```

and

```
[wasStartApp] Started Application [MavenWasSample]
```

VERIFY THE DEPLOYMENT

After Maven is finished deploying the EAR, the appli-

cation can be tested to ensure the EAR was successfully deployed. The Web application contains a simple JSP page accessible at http://localhost:9080/sample/digest.jsp. If the EAR is working properly, the page should display a simple form that can be used to perform MD5 hashing of entered text.

REBUILD THE APPLICATION

Once the application is working correctly, it may become necessary to reinstall the application if it is ever modified. This can also be accomplished through Maven by adding a property to the command line:

```
> maven —Dwas.reinstall.app=true
```

Customize the Deployment

Throughout the article a WAS default installation on a local machine was assumed. To deploy the application to a WAS instance with a special configuration, the build needs to be modified to accommodate those unique environments. Luckily, the WebSphere plug-in provides many configurable properties that can be set to alter the deployment behavior. Some of the most commonly used properties are shown in Table 1.

To use the SE properties, simply specify new property values on the command line. For example, to deploy the application to a different server needing user authentication, use a command like the following:

```
> maven —Dmaven.was5.server=myserver —Dmaven.was5.
username=deployer —Dmaven.was5.password=pw4deployer
```

The plug-in offers a wider range of configuration properties than those listed above and used in this article. Be sure to visit the WebSphere plug-in Web site to learn all of the plug-in's options

Promise of Build Scalability

In the beginning of this article, it was stated that Maven could help address build scalability by making it easy to add and maintain new components to a J2EE application.

How easy is it to add another EJB JAR or WAR to this application? To add another utility JAR, EJB JAR, or WAR to the application, all that needs to be done is to replicate the subproject structure for the type of component to be added, modify the project descriptor with the new project metadata and dependencies, and finally add the new component to the EAR's list of dependencies. After that is complete, when Maven is used next to deploy the EAR, the new component will be built with all the other modules and it will be bundled in the final J2EE application.

Summary

The ease of building and deploying J2EE applications and its ability to interface with an ever-increasing number of development tools and resources make Maven a viable build solution for WAS deployment.

LISTING 1

```
<dependency>
  <groupId>maven-was</groupId>
  <artifactId>sample-util</artifactId>
  <version>1.0</version>
  <properties>
    <ear.bundle>true</ear.bundle>
  </properties>
</dependency>
<dependency>
  <groupId>maven-was</groupId>
  <artifactId>sample-ejb</artifactId>
  <version>1.0</version>
  <type>ejb</type>
  <properties>
    <ear.bundle>true</ear.bundle>
  </properties>
</dependency>
<dependency>
  <groupId>maven-was</groupId>
  <artifactId>sample-war</artifactId>
  <version>1.0</version>
  <type>war</type>
  <properties>
    <ear.bundle>true</ear.bundle>
    <ear.appxml.war.context-root>sample</ear.appxml.war.
context-root>
  </properties>
</dependency>
```

LISTING 2

```
<postGoal name="ear:ear">
  <j:choose>
<j:when test="${was.reinstall.app}">
      <!-- reinstall the application -->
      <attainGoal name="was5:reinstallApp"/>
    </j:when>
    <j:otherwise>
      <!-- deploy EAR -->
      <attainGoal name="was5:installApp"/>

      <!-- start the application -->
      <attainGoal name="was5:startApp"/>
    </j:otherwise>
  </j:choose>
</postGoal>
```

LISTING 3

```
<!-- build all subprojects -->
<goal name="build">
  <j:set var="goals" value="build"/>
  <attainGoal name="projects:reactor"/>
</goal>

<!-- reactor goal -->
<goal name="projects:reactor">
  <maven:reactor
    basedir="${basedir}"
    includes="**/project.xml"
    excludes="project.xml"
    banner="Executing (${goals}):"
    ignoreFailures="false"
    goals="${goals}"
    postProcessing="true"
  />
</goal>
```

IBM's middleware solutions do it all

Can IT Systems Meet the Challenges Facing Life Sciences Organizations?

BY C.J. FROST

Time is money in any industry, but for life sciences companies a single day of having a new drug on the market can be worth hundreds of thousands, sometimes millions, of dollars. And that money, in addition to its obvious effect on the balance sheet, funds research into new treatments that can literally mean life and death.

So the need for efficient, integrated systems, and the need to conduct research and get to market ever more quickly is acute. Meanwhile, pharmaceutical, biotechnology, and research entities must meet more stringent regulatory requirements than ever before. In addition, the clinical trial process (a critical part of research), has become increasingly complex, costly, and risky for huge multinational organizations as well as small, specialized subcontractors.

Life sciences organizations also need to respond to societal, business, and technology trends. On a societal level, the population is aging yet is more informed and demanding than ever. Core business dynamics have changed: succeeding by producing single "blockbuster" treatments is risky and difficult, especially with the need to reduce errors and costs while improving quality; generics have changed the cost-to-profit equation; and there is increased focus on research and development. New technologies in imaging, genomics, proteomics, systems biology, and more have great promise but take time and expertise to implement.

"Every day that a pharmaceutical company has a treatment in the market is worth a lot of money and can make critical differences in people's lives. So even cutting one year off of an average 15-year research process is huge," said C.J. Frost, market segment manager, Life Sciences, IBM Software Group.

In addition, said Frost, "These companies are in the middle of a huge change in the way drugs are created. While it has the potential to radically change the practice of medicine, it's very challenging for companies right now."

In the past, drugs were created to treat symptoms. More recently, treatments were designed to eliminate infections and diseases. Now, pharmaceutical companies are starting to develop "targeted treatments," in which researchers study groups of patients with like genetic makeup, environment, and health issues to determine who is most vulnerable to a given condition. Eventually, targeted treatment will mean creating treatments that are designed specifically to be most effective for certain genetic groups, said Frost.

IT systems are critical in how life sciences companies will grapple with these medical, research, societal, and business issues to improve lives and be successful. They need to integrate their business and IT processes within and beyond their own organizations, allowing them to respond with speed to customers', researchers' and medical professionals' demands, competitive threats, and regulatory requirements.

IBM is working with life sciences groups to create this capability, helping develop life sciences discovery and clinical genomics standards, and teaming with organizations and universities to build a worldwide repository for all protein structure information.

This experience has been key in the creation of IBM's Industry Middleware Solutions for Life Sciences, which are designed to address the industry's most pressing challenges. The solutions are part of IBM's effort to deliver middleware solutions based on customer preferences for buying solutions designed for their industry. Each solution draws on the appropriate functions from IBM's WebSphere, Lotus, Tivoli, DB2, and Rational middleware brands and IBM's industry-specific middleware, combined with applications from independent software vendors (ISVs) and industry-expert services.

There are five IBM middleware solutions for life sciences entities. They are modular, which allows companies to implement pieces at their own pace, and are based on open standards that support platform independence.

"We wanted our overall architecture for the solutions to ensure that all the components work well together and efficiently so that customers can move out from addressing one challenge to the others based on their needs," said David A. Epstein, director of Solution Development, Public Sector, IBM.

Other important design principles, he said, were to have reusable components, use industry standards including business and data models, adhere to open standards, and build a services-oriented approach. "It was also absolutely essential that we build them to integrate with existing applications and other technologies in use in our clients' environments," he said.

The solutions specifically address the most pressing challenges facing the life sciences industry: complying with government requirements throughout their business and development processes, efficiently managing clinical development and trial processes, and the need to undertake development of targeted treatments.

The need to meet increasingly stringent regulatory requirements for the safe creation of drugs is one of the industry's biggest challenges because "in the past if a problem was discovered in manufacturing, the government could shut down the production line where it occurred," said Epstein. "Now, they view it as a process problem and can stop related processes, which can shut down an entire company."

Adds Frost, "In years past, drug companies had to demonstrate compliance at the end of the development process. Today, they must do so throughout the process while adhering to business-oriented requirements as well."

Challenges to do so include the fact that there are very large amounts of disparate data with little or no information security at many life sciences organizations. Often they also have IT environments that include old systems that have to be maintained because they contain critical research data that is needed in audit processes and can have future value.

The IBM Middleware Solution for Life Sciences Corporate Information Asset Management is designed to provide secure storage and processing of life sciences documentation and data records to help companies efficiently meet regulatory requirements. The solution provides a records management environment for the storage; search/retrieval; long-term archiving and disaster recovery of data, documents, analytical results; and e-mail messages generated from pharmaceutical discovery through manufacturing and delivery. It improves workflow, streamlines access to information,

"The ability to conduct research as quickly as possible, and get to market fast, will be the difference between success and failure for many of these companies"

and improves security and risk management.

Life sciences companies also have issues in managing the increasingly complex and costly clinical trial process, including recruitment of investigators. "They have to recruit the right doctors/investigators with the right patients among decreasing numbers of willing participants. They have to mitigate the potential liability in the use of experimental drugs, and they have to deal with subcontractors and a lack of information and access that have made the logistics increasingly complicated," said Frost.

Furthermore, he said, "they have to do all this while meeting government requirements precisely for trials, cutting costs, and completing the process as quickly as possible."

The IBM Middleware Solution for Life Sciences Investigator Recruitment and Trials Management is designed to increase the effectiveness and productivity of clinical trials by bringing together learning modules, investigator resources, collaboration capabilities, and access to applications. The solution provides an integrated, portal-based environment for investigator access which can result in more educated investigator teams, automated recruitment, increased investigator collaboration, and more rapid trials.

IBM also has a solution designed for small- and medium-sized businesses participating in trials, usually as subcontractors in the drug development effort. Like large entities, SMB companies must meet stringent regulatory requirements and face intense competition while needing to access and analyze increasing amounts of data and inconsistent processes.

The IBM Middleware Solution for Life Sciences Clinical Trials Management helps life sciences organizations manage time-critical information, documents, budgets, and schedules about investigators, patients, clinical-trial staff – in short, all the components of a clinical trial. This solution can also be used to integrate an organization's clinical trial management process with the FDA submission document process needed to satisfy government regulations. It provides a single interface to the information on a trial and creates a process that can be audited. It is designed to help organizations reduce and manage risk, improve workflow, reduce errors, and increase overall clinical trial efficiency.

There are also challenges in the drug development process before the clinical trial stage. They include the effort to produce targeted treatments that are based on the study of genes and their interactions, or genomics. To do so, healthcare and pharmaceutical organizations must access and analyze increasing amounts of complex data.

The IBM Middleware Solution for Life Sciences Clinical Genomics provides an environment for capturing clinical patient data for reuse as the basis for drug development and other research. By offering data aggregation and streamlined access, the solution helps organizations make use of key data and thus more rapidly identify potentially fruitful development projects. This improved capture and integration of patient data increases process efficiency and improves reuse of clinical data which can cut time needed for research.

For example, IBM and the Mayo Clinic have integrated 4.4 million patient records, which were in a variety of incompatible formats, into a unified system based on a standard technology platform with robust security and privacy features. This will give physicians and researchers access to a comprehensive set of records that can be analyzed with the safeguards needed to protect patient information, ensure confidentiality, and meet government standards.

Another challenge in the R&D process is the need to reduce repetition that results from the loss of key information, including ideas, inferences, and intuition from researchers. Life sciences organizations also need to improve overall knowledge sharing and ensure information exists to bolster patent efforts.

The IBM Middleware Solution for Life Sciences Annotations and Knowledge Sharing helps organizations manage intellectual capital and store annotations. It provides a collaborative environment for sharing knowledge, expertise, and ideas. Insights can be identified and linked with data used for analysis. Key benefits are improved creation of document annotations, increased security of annotation storage, and improved ability to share and search for annotated content. It also helps formalize the decision-making process and makes archived information available for regulatory and patent purposes. The overall result can be more efficient discovery and faster time-to-market.

"Not a small thing to life sciences organizations," said Frost. "The ability to conduct research as quickly as possible, and get to market fast, will be the difference between success and failure for many of these companies."

Using IBM Lotus Workplace for Business Controls and Reporting

Implementing Sarbanes-Oxley

ASHWINI KUMAR &
RAGHAV MATHUR

Ashwini Kumar is a senior software consultant at Prolifics. An IBM certified portal solutions developer, Ashwini has over 8 years of experience in building scalable and distributed J2EE and struts-based Web applications, and is also experienced in process choreography, Web services and business integration solutions. He holds a master of technology degree in computer sciences from Mysore University, India and an MS in engineering management from the Gordon Institute of Tufts University.

akumar@prolifics.com

The passage of the Sarbanes-Oxley Act of 2002 (SOX) marks a new era of accountability for corporate officers. Signed into law by President Bush in response to corporate accounting scandals, it is a major concern and top priority for the CEOs, CFOs, boards of directors, and audit committees of public companies, as well as for auditors, accountants, attorneys and regulatory governing bodies. Starting in 2004 (2005 for smaller companies), the financial reports of publicly traded companies in the United States must begin to comply with the financial disclosure requirements of this Act.

One of the major areas of concern is Section 404, *Management Assessment Of Internal Controls*, which requires companies to include in their annual report "an internal control report, which shall:
(1) State the responsibility of management for establishing and maintaining an adequate internal control structure and procedures for financial reporting; and
(2) Contain an assessment, as of the end of the issuer's fiscal year, of the effectiveness of the internal control structure and procedures of the issuer for financial reporting."

Successful compliance with Section 404 of SOX requires management to make visible and provide documentation of the status of all the various compliance issues associated with the new law. In addition, the company's auditors must review the company's SOX compliance and include an assessment in the auditing report. In order to provide such extensive documentation, companies are turning to solutions offering both content management and reporting components, such as IBM Lotus Workplace for Business Controls and Reporting.

The Act has heralded an age of intense content management at a time when content has morphed into many formats. Scanned paper images from accounting departments, electronic forms from human resources, presentations, e-mail, online messaging logs, and even voicemails are all forms of data made equal under the auspices of the Sarbanes-Oxley Act.

Using information technologies, companies can achieve the following goals needed for SOX compliance:

1. An unambiguous definition of the procedures that a business user, not an information technology professional, could easily use
2. A comprehensive explanation of the business process and the associated roles and responsibilities
3. Automatic reference from a single process step to related, unstructured information and reports
4. Clear identification of the controls within the process
5. The ability to monitor processes and procedures related to content management

IBM's Lotus Workplace products, which are geared towards increasing workplace productivity using advanced content management and diversified collaborative tools, have been further enhanced to address the special requirements of Sarbanes-Oxley with Lotus Workplace for Business Controls and Reporting. This offering is an amalgamation of IBM's expertise in enterprise software and KPMG's expertise in business internal controls and auditing procedures. It helps provide a platform for an organization's business reporting process and a framework for gathering and organizing information about business controls.

The Web-based Workplace software leverages a range of third-party control catalogs and knowledge of industry-specific internal processes to help businesses understand and prepare for Sarbanes-Oxley mandates and jump-start the controls process. An organization can identify, assign, test, and monitor controls. The solution provides role-based access, which directs controls and financial reporting directly to the individuals responsible for execution, and provides real-time access with the "at-a-glance" dashboard. Both allow for quick issue identification, risk mitigation, and responsiveness. The solution also adds minimal impact on day-to-day operations enabling compliance activities to become a fluid part of employees' regular routines.

The content management component of this product allows content to be checked in and out of the system's repository. Library services also provide key features allowing metadata and audit log management, as well as content versioning – safeguards that enable users to roll back to previous versions if necessary. This repository is further augmented by the capability to search using full-text, keyword, or advanced technologies such as pattern recognition, expert recommendation, and semantic searching.

Lotus Workplace for Business Controls and Reporting is built on the industry's leading middleware technologies, IBM WebSphere Portal, and IBM DB2 Content Manager, which provide a single, unified and reliable platform for the entire organization. Lotus Workplace for Business Controls and Reporting also includes a fully integrated Crystal Enterprise Reporting Engine for generating the different reports showing the effectiveness and status of the business controls.

Other features of Lotus Workplace for Business Controls and Reporting include:

- ***A role-based interface.*** When employees log in to the system, they are presented with their individual tasks and responsibilities and the resources and tools they need to complete those tasks.
- ***Role-based reports.*** Role-based reports allow you to report on various combinations of data. You can have executive views showing summaries of control effectiveness and deficiencies in graphical format that are linked to detailed reports about each control. You can generate reports for employees and managers that need to combine information from different data sources.
- ***Robust security features.*** You can protect your sensitive data with features such as single sign-on, role-based access and read-only access.
- ***Scalable framework.*** The product provides a framework allowing you to organize and document your business processes

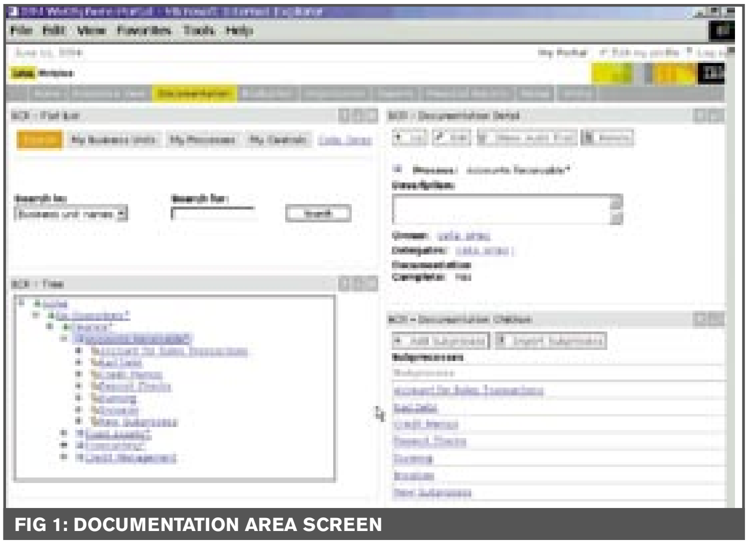


FIG 1: DOCUMENTATION AREA SCREEN

WebSphere Journal Advertiser Index...

COMPANY	URL	PHONE	PG
EV1 SERVERS	WWW.EV1SERVERS.NET	800-504-SURF	23
H&W COMPUTER SYSTEMS	WWW.HWCS.COM/WSJ19.ASP	800-338-6692	13
IBM	WWW.IBM.COM/MIDDLEWARE/AGILITY	800-IBM-4YOU	2
IBM	WWW.IBM.COM/MIDDLEWARE/BUILD	800-IBM-4YOU	15
INFERDATA	WWW.INFERDATA.COM/WDJ	888-211-3421	3
INFOSECURITY	WWW.INFOSECURITYEVENT.COM/WEBSECURE	888-251-0566	27
ITEMFIELD	WWW.ITEMFIELD.COM/WBI	800-838-1055	51
ITKO	WWW.ITKO.COM	877-BUY-ITKO	43
MOTIVE	WWW.MOTIVE.COM/WITHIN1	512-531-2527	52
NASTEL TECHNOLOGIES	WWW.NASTEL.COM/UNCOMPLICATE	800-580-2344	47
PROLIFICS	WWW.PROLIFICS.COM	800-675-5419	19
SYS-CON E-NEWSLETTERS	WWW.SYS-CON.COM	888-303-5282	29
TANGOSOL	WWW.TANGOSOL.COM	617-623-5782	35
WEB SERVICES EDGE 2005	WWW.SYS-CON.COM/EDGE	201-802-3045	48-49
WILY TECHNOLOGY	WWW.WILYTECH.COM	888-GET-WILY	5
WINMILL SOFTWARE	WWW.WINMILL.COM	888-711-6455	31

WebSphere Journal Coming Next Month...

What's New In WebSphere Portal v5.1?

Take enterprise integration to the next level

Looking Beyond MQ

Improve levels of service and avoid costly upgrades

Develop Web Services Clients

Using Macromedia Flash MX 2000

Plug-and-Play Remote Portlets

A vision of Web services for remote portals and expand as needed.

- ***24/7 access.*** Implemented as a Web-based portal application, it gives 24/7 workplace access to employees and users. Even if your employees are offsite, they can retrieve the information and find the tools they need to do their work.
- ***Collaborative environment.*** Electronic workplaces allow employees to collaborate on projects and tasks. Presence awareness and instant messaging allow for real-time resolution of questions and problems.
- ***Ease of navigation.*** Enhanced navigation allows users to quickly complete control tasks for greater focus on core business.
- ***Audit trails and archiving.*** Audit trails and archiving help to ensure process and document integrity.
- ***Third party enhancements.*** You can take advantage of industry insights and knowledge of internal control processes and practices via leading third-party control catalogs from vendors such as KPMG.
- ***Integration of existing systems.*** It doesn't replace existing SAP, PeopleSoft, etc., applications. Instead it accesses information from those applications to present in reports for management so that management can assess risk in their organizations.

Figures 1 and 2 illustrate the implementation of Lotus Workplace for Business Controls and Reporting for the Acme Company. The menu along the top allows easy access into all the different areas, such as documentation, evaluation, organization, and reports. Figure 1 shows the documentation area. The left part of the screen contains a navigation tree illustrating the organizational structure of the business processes or controls. On the right is detailed information about the accounts receivable process including the owner of this process and the list of subprocesses.

Figure 2 illustrates how you can drill down to the controls that make up a process, in this case the controls that are a part of the Bad Debt process in Accounts Receivable.

Technology Components and Servers

Components of Lotus Workplace for Business Controls and Reporting (LWBCR) are installed on three core systems (machines). Distribution of components on these three systems is based on the role performed by each system in the configuration. A typical distribution of these components is listed here:

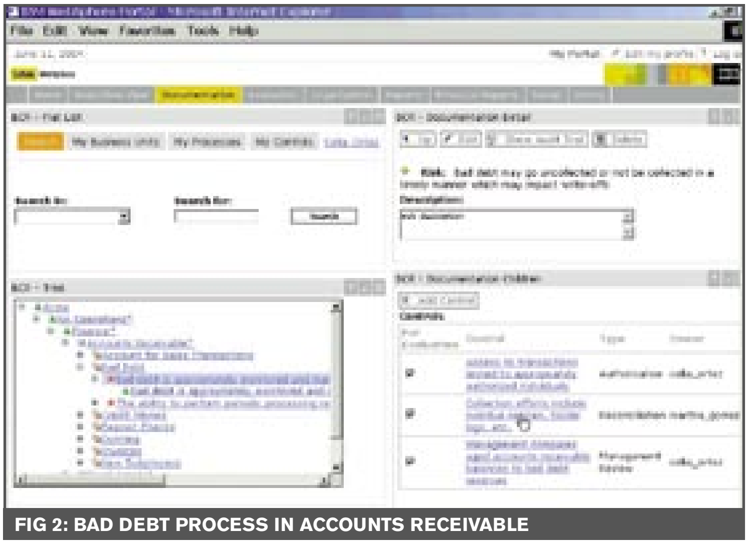


FIG 2: BAD DEBT PROCESS IN ACCOUNTS RECEIVABLE

WEBSPHERE PORTAL SERVER

A platform for the following components used in application presentation and business logic:

- WebSphere Portal 5.0.2
- DB2 8.1 ESE Fixpack5 (DB2 Client)
- Information Integrator for Content Developer Client (II4C) 8.2

CONTENT MANAGER SERVER

A server for data storage and for controlling user access based on their roles with the following components:

- DB2 8.1
- Visual C++ 6
- WebSphere Application Server 5
- Content Manager 8.2
- Information Integrator for Content Developer Client (II4C) 8.2
- IBM Directory Server SDK 5.1

CRYSTAL ENTERPRISE SERVER

The Crystal reporting engine is used for generating the reports that access the database on the content manager server and render the images over HTTP. The components installed are:

- DB2 8.1 (required for IDS 5.1)
- IBM HTTP Server 2.x
- Crystal Enterprise Server 10
- IBM Directory Server SDK 5.1

For a Windows installation, these servers should typically have 2.0 GHz P4 CPU with a 2GB RAM on each machine.

Installation

The installation of Lotus Workplace for Business Controls and Reporting is a very complex and sensitive process involving the correct installation and configuration of several IBM and third-party software components. It is recommended that you leverage IBM services or hire a Lotus Workplace consulting expert such as Prolifics to expedite the installation process.

Install the Workplace components in the following order:

Raghav Mathur is a senior architect for the Portal Consulting Practice at Prolifics. A certified portal solutions developer, Raghav has expertise in products and technologies including WebSphere Application Server, WebSphere Portal Server, WebSphere Studio Application Developer, DB2, J2EE, and Java. Raghav helps organizations to translate their business requirements into a robust portal architecture, providing the overall design specifications and validating the existing design.

rmathur@prolifics.com

1. WebSphere Portal on the Portal server
2. DB2 Installation on the Content Manager (CM) server
3. IBM Content Manager on the CM server
4. WebSphere Application Server installation on the CM server
5. Installation of Information Integrator for Content Developer Client (II4C) on Portal and the CM server
6. IBM Directory Server on the CM server
7. DB2 Developer Client on the Crystal Enterprise (CE) server
8. Crystal Reports Server Install on the CE server

Getting Started

After the software is installed and configured, you need to input the information specific to your business. The following steps provide an overview of the process you would follow in order to implement Lotus Workplace for Business Controls and Reporting.

1. Typically you start by defining your organization – the managers and employees that make up your company – and assign the ownership of business units to the corresponding manager.
2. After entering your organization information, you need to document your business processes, objectives, risks, and business controls. Alternatively, you can import best practices.
3. Part of defining your business processes and controls is to document how this control can be evaluated and tested along with a date for performing these tests. As you use the system, you will need to continue this evaluation process in order to determine its effectiveness.
4. You can attach or link documents to the processes with the necessary financial data.
5. In order to check the effectiveness of the system, generate reports for your processes and controls.

Summary

Even though publicly traded companies are required to comply with the Sarbanes-Oxley Act, this legislation does provide an opportunity for all organizations to review their business controls and streamline their business processes. For companies with paper-based financial reporting systems, this can be the time to migrate to online controls and the advantages that come with an online system. With Lotus Workplace for Business Controls and Reporting you have a tool to start and implement the process and a tool that can grow with your needs.

The following is a summary of some its key benefits:

- Lotus Workplace for Business Controls and Reporting leverages third-party control catalogs that are ERP and industry-specific, and internal controls knowledge from KPMG to create best practices and jump-start the process.
- The portal infrastructure aggregates information for an enterprise-wide view and provides a unified approach for companies to conduct self-assessment of internal controls at an entity-wide level.
- The role-based, real-time access enables companies to make more informed business decisions and help mitigate risks.
- Sophisticated content management and reporting capabilities allow companies to consolidate any format of content whether paper-based, voice-based, or online, and present a detailed assessment of compliance within the auditing reports.

With Lotus Workplace for Business Controls and Reporting, companies gain the information and control needed to assess internal controls for financial reporting, respond to Sarbanes-Oxley Section 404, and improve visibility into business processes.

Security, Firewalls, and Keeping the Door Locked

–continued from page 4

Beginning in early 2004, data obtained through court authorized intercepts revealed internal communications, transactions, and practices of the previously identified groups and other criminal organizations. The assistance of international law enforcement – particularly the United Kingdom's National Hi-Tech Crimes Unit, the Vancouver Police Department's Financial Crimes section, the Royal Canadian Mounted Police, and Europol – has been and will continue to be critical to the success of Operation Firewall. In addition, specific support was provided by Secretary General Boiko Borisov of Bulgaria, as well as police agencies in Belarus, Poland, Sweden, the Netherlands, Ukraine, and the U.S. Department of State.

Here is what is important – it took hundreds of people from over a dozen U.S. government agencies and who knows how many foreign government agencies over a year to catch 28 people. As criminals become more adept at e-business, they will quickly overwhelm all of the available law enforcement resources and you and I will be on our own. Just as in the material world, you will have to make sure you lock your own door.

From Chapter 6

Building a Combined WebSphere and Domino System

BY JOHN LAMB, MICHAEL LASKEY, & GOPAL INDURKHYA

This chapter, "Building a Combined WebSphere and Domino System" is excerpted from the new book, ***IBM WebSphere and Lotus: Implementing Collaborative Solutions***, authored by John Lamb, Michael Laskey and Gopal Indurkhya. Copyright International Business Machines Corporation 2005. ISBN 0-13-144330-5. To learn more, www.phptr.com/title/0131443305.

In this chapter, we describe various ways to build and combine the WebSphere Application Server (WAS) and the Lotus Domino Collaboration Server (Domino) products to present, from the application user's viewpoint, a single system image. We first discuss options for setting up WAS and Domino server systems. In particular, we present common ways WAS and Domino server systems are placed in a multiple tiered network configuration behind the Internet facing firewall. We then consider installation planning, including hardware and operating system requirements. Finally, we discuss product installation and migration issues.

Throughout the chapter, we refer to WAS and Domino features in the current product versions as of this writing, namely, WAS V5 and Domino R6.5. As a reference aid, we present the details for the installation steps for WAS V5 and Domino R6.5 on different computers in Appendix C, "Detailed Steps for Building a Combined WebSphere and Domino System."

Options for a Combined Server Configuration

There are various options for how to build the combined WAS and Domino system. These options differ mainly in what functions the WAS and Domino servers take on and where they are placed in the usual tiered network configuration. A common network configuration for providing a Web application to Internet users consists of three tiers (or layers or zones), each separated by firewall systems. The Internet facing tier is often referred to as the "demilitarized zone," or DMZ, and usually consists of load balancing or authentication systems. Web and application servers sit in a zone behind the DMZ, and data servers in a "data layer" behind that. The multitiered configuration provides increasing security protection from the DMZ through the data layer. The server systems are placed within the tiers according to security and inter-system performance considerations.

Depending on the roles they play, WAS and Domino usually sit in either of the two network tiers behind the DMZ. As a Web application server (for servlets, JSPs, or Web services), WAS is often located in the application zone. When WAS serves primarily as an EJB container, it is placed in the data layer, so as to be close to (or co-resident with) the database server. Domino used as an HTTP server can be placed in the application zone (or even DMZ if necessary). However, it is most common to find Domino in the data layer being used as a data or document repository or collaboration server.

In a production application environment, it is most often the case that WAS and Domino are installed on separate computers. This is done simply to allow the WAS and Domino systems to be placed in separate network tiers. It is also done for scalability and performance reasons and to make system management easier. Although with more sophisticated hardware platforms, it is possible to have all system components installed on one physical system, for example using hardware partitioning to create separate server images on a single system.

In some cases, such as for development or testing, it is convenient to install Domino and WAS on a single computer. The steps for installing Domino and WebSphere on the same computer are basically the same as for separate computers, although there are some simplifications in the install procedure when both applications share the same hardware platform. Those simplifications are described in the configuration and setup details given later in this chapter and in Appendix C.

WAS and Domino Installation Planning

HARDWARE/OPERATING SYSTEM REQUIREMENTS

Both Domino and WebSphere run on a variety of hardware and software platforms. In particular, Domino runs on Windows 2000/2003, AIX, Solaris, HP-UX, Linux, Netware, OS/2, OS/400 (IBM iSeries), and z/OS (IBM zSeries), while WebSphere runs on Windows 2000/2003, AIX, Solaris, HP-UX, and Linux for Intel and IBM iSeries, pSeries, and zSeries processors.

Linux is becoming an operating system to seriously consider for both WAS and Domino. Linux on Intel or AMD servers is popular for low to mid-range servers (up to eight processors) because of generally lower cost. Moreover, Linux spans the widest range of server types, from blade server to enterprise mainframe so that common system administration resources can be applied. It is important to note that as a complete operating system, Linux is provided in various combinations of specific component (especially kernel and library) versions, referred to as "distributions." For both WAS and Domino, IBM provides support for only certain Linux distributions, namely RedHat, SuSE, and distributions conforming to the United Linux 1.0 standard. It is likely, however, that WAS and Domino will run without problem on Linux distributions similar to these, such as Mandrake Linux, which is similar to RedHat.

The basic hardware requirements are what you would expect for large, sophisticated system applications. System memory size is the most important hardware requirement. Both WAS and Domino specify a recommended amount of 512MB, but for best performance you should provide the most memory you can afford. Processor speed is recommended to be at least equivalent to a 500MHz Intel Pentium processor but again should be as high as affordable.

Disk space requirements vary by the specific operating system platform but are in the range of 600-800MB for WAS and 500MB for Domino. The disk drives and interfaces themselves should be high-performance, as usually found in most server systems. Careful consideration should be given to the use of external storage systems such as storage area networks or network file systems as repositories for WAS or Domino data because they may degrade application performance.

A set of hardware requirements often overlooked in the design of WAS and Domino systems is that related to the network hardware. The speed and bandwidth capacity of the network routers and firewalls should be sufficient so as not to introduce delays in the data paths to or between the WAS and Domino system components.

WAS and Domino Product Coexistence

Domino and WAS have long had product features that allow them to work together, but in order for the features to work properly, the correct versions of Domino and WAS are required. Sometimes the versions must be at a specific service update level. For example, Single Sign-On (SSO) is one very important feature for applications built on Domino and WAS. (The SSO feature lets Domino and WebSphere recognize when either server has already authenticated a user, as is described in detail in Chapter 12, "Security and Single Sign-On.") The first version for which SSO could be enabled for Domino was R5.0.5. The first version of WAS (Advanced Edition) for which SSO could be enabled was version 3.5 with service update (fix pack) number 1, or version 3.5.1. The SSO feature of iSeries was first available with Domino for AS/400 5.0.6a and WAS 3.5. Servlet access to Domino databases was first available on iSeries with Domino for AS/400 5.0.4 and WAS 3.0.2. The ability to use Domino HTTP Server to access WAS resources was first available on iSeries with Domino for AS/400 5.0.5 and WebSphere 3.5.1.

In order to have Domino and WAS work together, the requirements are as follows:

- A minimum Domino level of R5.0.5
- A minimum WebSphere level of version 3.5.1

But this is somewhat ancient history now. Significant new releases of both Domino and WAS were made available at the end of 2002 with the Domino R6 and WAS V5 products and continue on almost a yearly basis.

HTTP Server Considerations

Another area of interoperability between WAS and Domino is the HTTP server plug-in, which enables J2EE servlet and JSP requests to be forwarded to WAS for processing. WAS provides various plug-in modules for different HTTP server products such as Apache, IBM HTTP Server, Microsoft IIS, and Domino's integrated HTTP server. In the case of the Domino HTTP server, the WAS plug-in is provided as part of the Domino product package. Various plug-in modules are provided corresponding to the different WAS versions and operating systems. Table 6-1 lists the set of plug-in modules provided by WAS V5.

With the recent versions of WAS, the communication mechanism between the WAS HTTP server plug-in and WAS itself has been changed from a proprietary mechanism to HTTP. This change has greatly simplified the configuration and setup of WAS and the plug-in. The use of HTTP makes it easier to configure the network layer, especially firewall systems, to support the plug-in connection to WAS. The implication that WAS now provides its own HTTP server is true. Although the WAS HTTP server is far from a full-fledged HTTP server, it is sufficient to provide a quick and easy means for accessing WAS without having to set up a separate HTTP server. As we will see later, the built-in WAS HTTP server is also used to provide a Web administration interface instead of a separate administration process and corresponding client.

Given that the Domino HTTP server is the only HTTP server that can provide Web access to Domino databases (like a client's mail or address book database), the situation arises where you need to provide both a "standard" HTTP server and the Domino HTTP server. In this case, you can simply configure the Domino HTTP at a different port or (virtual) host from the other HTTP server. Then the application can supply URLs for Domino resources specifying this alternate port or hostname. Another approach available on Windows systems is to use Microsoft IIS as the HTTP server and configure it to use a plug-in that redirects Domino requests. This IIS Domino plug-in can be used in addition to the IIS WAS plug-in.

For certain IBM platforms, namely the iSeries (OS/400) and zSeries (z/OS), it is difficult to run multiple HTTP servers because the HTTP server is essentially built into the operating system. In these cases, there are also HTTP plug-in modules available for Domino on the iSeries platform (refer to the IBM Redbook, "Domino and WebSphere Integration on iSeries," for details) and the zSeries, similar to the Domino plug-in for IIS. These Domino plug-ins can be run in combination with the WAS plug-ins on these platforms.

See the section on configuring the Domino HTTP server plug-in later in this chapter for details on how the HTTP plug-in can be configured to work with WAS.

Networking Considerations

Prior to installing WAS and Domino, it is a good practice to establish your network naming conventions. For most applications, you will want to use fully qualified host names for the WAS and Domino servers, that is, names consisting of a TCP/IP host name and domain name. If you don't have a domain name system (DNS) server set up, it is possible to configure each server operating system with the host/domain name to IP address mappings (e.g., via the /etc/hosts file on both Unix and Windows systems). Certain features in WAS and Domino require fully-qualified host names to work, especially SSO.

If Domino servers don't already exist in your network, you'll want to establish a Domino naming convention for the servers. Here you'll need to pick a common Domino domain name and server names for the planned Domino servers. For those new to Domino, the Domino domain name is altogether different than a DNS domain name, although conceptually they are quite similar. The Domino servers will need to be given fully qualified TCP/IP host names in addition to their Domino server names.

If you plan to use SSO across your WAS and Domino servers, you will need to put all of the WAS and Domino servers in a single DNS domain and all of the Domino servers in a single Domino domain.

One area of network planning often overlooked until deployment is that of firewall rule configuration. As mentioned previously, WAS and Domino servers are usually deployed to a tiered network where the tiers are separated by firewall systems. As a result, a Web application can easily span every tier, and the communication between the components running on WAS and Domino will need to be permitted to pass through the firewalls. Since the firewall systems control TCP/IP sessions by their source and destination host addresses, and ports and TCP/IP protocols are usually associated with specific ports or port ranges, you should understand which protocol types must be allowed to pass to, from, or between WAS and Domino. Besides HTTP, WAS and Domino servers may make use of several other Internet (and proprietary) protocols, such as LDAP, SMTP, DIIOP, SSL, etc. TCP/IP protocols used by other components such as database servers or chat servers must be taken into account as well.

Domino Server Configuration and Set-Up Considerations

The Lotus Domino R6 Server family consists of the Domino Mail Server, the Domino Application Server, and the Domino Enterprise Server. Most likely you will want to do more with Domino than just use its HTTP server to integrate with WebSphere, so you will want to install either the Application or the Enterprise server.

The Domino server supports various features, especially Internet protocols, which may be useful for your planned applications. These features and protocols include the following:

- Access to the Domino directory via the Lightweight Directory Access Protocol (LDAP). LDAP access may be used with SSO. (We discuss SSO in detail in Chapter 12.)
- Simple Mail Transfer Protocol (SMTP). Web applications could make use of SMTP, especially via the JavaMail API, for sending e-mail.
- Post Office Protocol (POP). An e-mail client-to-server protocol.
- Internet Mail Access Protocol (IMAP). Another e-mail client-to-server protocol.
- Domino Internet Inter-Object

■ 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.

Protocol (DIIOP). DIIOP is a communication mechanism for invoking Domino functions in a client-server model. It is required if J2EE components running on a WAS server make use of the Domino Java API to access remote Domino servers.

Depending on your application requirements, you may want to enable or disable a specific set of Domino features or protocols.

WAS Configuration and Set-Up Considerations

The deployment options for a WAS installation are many. With just the WAS Advanced Server edition, it is possible to set up multiple application server instances on a single system. This can be done even among different WAS versions. Or it can be done where the instances are at the same level and share a common runtime. With the Network Deployment edition, WAS can be installed and managed across multiple systems. A network deployment consists of "cells" containing "nodes" containing application server instances. The WAS deployment options are flexible enough to accommodate almost any installation requirements. You can plan a deployment to suit your needs—be they performance, capacity, or compatibility with previous deployments. We do not cover the deployment options in detail in this book and restrict our discussion to a single-server deployment.

The WAS administrative interface allows an administrator to manage WAS nodes, cells, and servers, as well as the applications running in them. It provides the means to monitor application server activities, manage security, and adjust the configuration. With WAS V5, the administrative interface is now a Web interface, the administration application is a Web application running on the application server, and the configuration data is maintained as a set of XML files on the server's filesystem. It is accessed directly on the server by specifying the server's host name and the port for the administrative application. Note that with the use of XML files to store configuration data, WAS is no longer dependent on a database system such as IBM's DB2. WAS requires a database system only if applications require it, such as for EJB entity beans.

For SSO support, there are a number of considerations involving the set up of the user registry used for user authentication. Please refer to Chapter 12 for the details.

General Installation Procedure for WAS and Domino

This section describes in general steps the installation of WAS and Domino on separate systems. The steps presented here will give you an overall idea of how to build each server. For a detailed description of the installation for WAS 5.0 and Domino 6.5, see Appendix C. The steps are as follows:

1. Check for necessary hardware and software prerequisites.
2. Verify the network configuration.

Working with the WAS system:

3. Log on as a user with administration privileges (e.g., as "root" for Unix systems).
4. If a database is required, install the database software.
5. Install WebSphere via the install utility (Java GUI).
6. Install the latest service update (fixpack) and any relevant e-fixes.
7. Start WAS, and verify that WAS resources are accessible. The "snoop" servlet sample can be used for this.

Working with the Domino computer:

8. If not available on any system, install a Notes Administration client.
9. Install the Domino server.
10. Using the Notes Administration client, configure the Domino server, and populate the directory.
11. If the Domino HTTP server will be used with WAS, ensure the plug-in module and configuration file are available to the server, and configure Domino to use the plug-in via the Administration client. Modify the plug-in configuration file if necessary.
12. Verify Domino resources can be accessed. If the plug-in was installed, verify that WAS resources can be accessed (e.g., the "snoop" servlet).

There is nothing in the installation procedure that prevents WAS and Domino from being installed on the same computer.

HARDWARE AND SOFTWARE PREREQUISITE DETAILS

In this section, we list the hardware and software requirements for the computer you want to install on, as well as the different product software levels required.

PLATFORM HARDWARE

- Pentium III or higher, with an absolute minimum of 256MB, but 512MB is recommended for WebSphere 3.5. We used 512MB for the computer on which we installed WebSphere and DB2 and 385MB for the computer on which we installed Domino for our testing, with satisfactory results.
- At least 600MB free on the drive or drives used to install the products. The disk space requirements of the products after installation are DB—475MB, Domino—300MB, IBM HTTP Server—20MB, and WebSphere—220MB.

The DB2 space requirements can be reduced by 125MB if you choose to install the DB2 Administration client on a separate machine. The space requirement for Domino also can be reduced to a certain extent by choosing to install fewer components. For example, it is possible to not install the help files. These files alone require 50 MB. Similarly, the space for WebSphere specified above includes documentation files of about 60MB, which would not be needed in a production installation.

PLATFORM SOFTWARE

- Any of the operating system software supported by WAS and Domino. For Microsoft Windows, either workstation or server code can be used. TCP/IP networking with a fixed IP address for each machine.

PRODUCT SOFTWARE LEVELS

The product software levels we used were as follows:

- DB2 Universal Database Version 8.1 Enterprise Edition plus Fix Pack 1
- Domino 6.5
- WebSphere Application Server V5

Higher versions of the products should also work. The highest currently available product level should be used except where specifically stated otherwise. Care should be taken, however, to check WebSphere for recent e-fixes.

CREATING A USER WITH ADMINISTRATION RIGHTS

WAS, Domino, and DB2 must run under the permissions of a user or as system services. For testing purposes, it is more flexible to use a user ID with rights to run as an extension of the operating system than load the products as system services (however, WAS and DB2 must run as system services). For a production system, these products should run as services so that they will automatically load when the system is started without operator sign-on.

Installing WebSphere Application Server V5

You can either install WebSphere from a product CD or one large installation file. If you use a product CD, the installation program should start automatically when you insert the CD. If not, you have to run the setup.exe program on the CD. If you have one big file, simply start the installation by running the file. It will automatically unpack itself and start the installation program. Note that the installation process itself requires 70MB or more free in the system temporary directory, even if installation is to another drive or volume. Step by step details are given in Appendix C.

IT solutions
GUIDE
Top IT Solution Providers!
TECHNOLOGIES YOU NEED
NOW!
How to Manage Your Ideas Using Today's i-Technologies

SYSTEM	WEB SERVER	PLUG-IN EXECUTABLE FILE
Windows	IBM HTTP Server	mod_ibm_app_server_http.dll
	Lotus Domino	libdomino5_http.dll
	Apache	mod_app_server_http.dll
	iPlanet (Netscape)	libns41_http.dll
	Microsoft IIS	iisWASPlugin_http.dll
IBM AIX	IBM HTTP Server	mod_ibm_app_server_http.so
	Lotus Domino	libdomino5_http.a
	Apache	mod_app_server_http.so
	iPlanet (Netscape)	libns41_http.so
	HP-UX IBM HTTP	Server mod_ibm_app_server_http.sl
	Lotus Domino	libdomino5_ http.sl
	Apache	mod_app_server_http.sl
	iPlanet (Netscape)	libns41_http.sl
Solaris	IBM HTTP Server	mod_ibm_app_server_http.so
System	Web Server	Plug-in Executable File
Lotus Domino	libdomino5_http.so	libdomino6_http.so
	Apache	mod_app_server_http.so
	iPlanet (Netscape)	libns41_http.so
Linux	IBM HTTP Server	mod_ibm_app_server_http.so
	Apache	mod_app_server_http.so

Note that there is no WAS plug-in provided for the Domino HTTP server on the Linux platform. Most of these modules are also shipped with the Domino product (except for the HP-UX modules).

TABLE 6-1 PLUG-IN MODULES PROVIDED BY WAS V5

Installing and Configuring Domino 6

We now turn our attention to our Domino computer and the installation of Lotus Domino.

The Lotus Domino R6 Server family consists of Domino Mail Server, Domino Application Server and Domino Enterprise Server. If you want to do more with Domino than just use its HTTP stack to integrate with WebSphere, you should install the application or the enterprise server.

To use the Domino HTTP stack and enable Single Sign-On, WebSphere V5 requires Domino Server R5.0.5 or higher. Step by step details for the installation of Domino Enterprise Server R6.5 are given in Appendix C.

You need to install the Domino administration client on your server or another workstation (Lotus recommends using a separate workstation for administration). This will allow you to change the server's settings easily, especially those in the Domino directory; although much, but not all of the testing we describe can be managed by direct access to the server's text console.

We will not describe the installation of Domino Administrator in detail. Be sure that the Administration Client is selected for installation; you can accept all other default values during the installation. One of the first things you can do after installing Domino Administrator is allow your users to run Java programs on the Domino server.

Configuring Domino to Use the WebSphere Plug-In

The plug-in itself is provided as a shared library (DLL for Windows, .a for AIX, .so for Solaris, and as of this writing no plug-in exists for Domino on Linux) and a default configuration file: plugin-cfg.xml.

The plug-in files are packaged with both the Domino and WAS product. The plug-in files can be found under the Domino install path: ./Domino/Data/domino/plugins/was5.

There are two essential steps to installing the plug-in: 1) specifying the plug-in library file to the Domino HTTP server and, 2) specifying the location of the plug-in configuration file to the plug-in itself. The first step is straightforward and simply requires a change to the Domino server document in the Domino Directory database (names.nsf). You specify the plug-in library file as a DSAPI filter file on the Internet Protocols/HTTP page of the Server document.

The second step is not as straightforward and is operating system-dependent. The plug-in library itself looks for the plug-in configuration file via an operating system environment variable (registry key on Windows). Unfortunately, these variable names are not well advertised in the WebSphere or Domino documentation. For IBM iSeries systems, the configuration file location is specified in the Domino notes.ini file. See Chapter 10 and Appendix C for the details on how to specify the plug-in file location and the plug-in configuration settings themselves.

Once these settings have been made, you can restart the Domino HTTP server task to have it load the WebSphere plug-in as a DSAPI filter. If the settings are correct, you will see the following messages at the Domino console:

```
HTTP Server: Java Virtual Machine loaded
WebSphere HTTP DSAPI filter loaded
HTTP Server: DSAPI WebSphere HTTP DSAPI Filter Loaded successfully
HTTP Server: Started
```

web services EDGE conference&expo

Web Services Edge 2005 East

International Web Services Conference & Expo

New for 2005:

- **Colocating with LinuxWorld Conference & Expo** badges give access to both shows
- **Guaranteed Minimum Attendance 3000 Delegates**
- **New, Hot Sessions & Seminars**

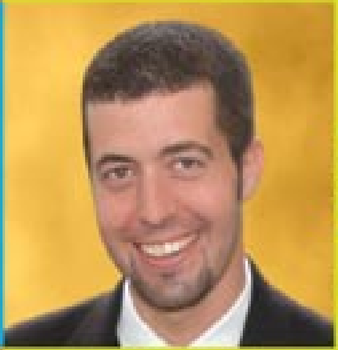
The Largest *i*-Technology Event of the Year!

Keynote Speakers & Featured Guests

Tuesday, February 15
11 a.m.

Matt Ackley
Senior Director, eBay Developers Program

Topic:
Web Services for e-Commerce



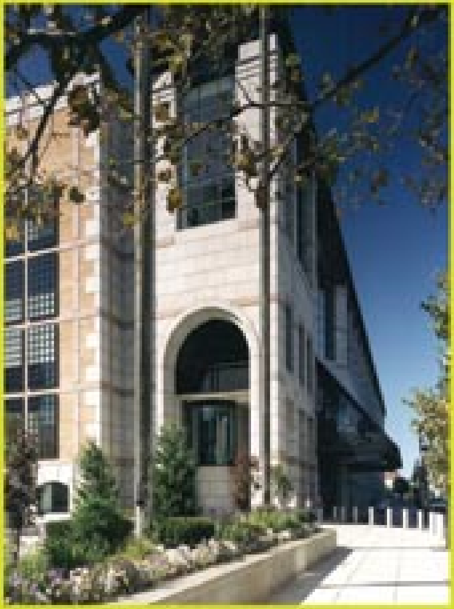
Wednesday, February 16
11 a.m.

Ari Bixhorn
Director, Web Services Strategies, Microsoft Corporation

Topic:
Indigo and the Future of Web Services

Anne Thomas Manes
Burton Group

Application Server Shoot-Out Facilitator



Hynes Convention Center
Boston, MA
February 15-17, 2005

For Exhibit and Sponsorship Information:
Jim Hanchrow, 201-802-3066, jimh@sys-con.com

Register Now! Hot Sessions & Seminars

- Ensuring Web Services Interoperability
- Web Services Standards: Going Behind the Mask
- The Role of Policy in Web Services Integration – It's More Than Just Security
- Four Abilities SOA will lack without a Registry
- Driving SOA Governance
- BPEL Best Practices from Real-World Projects
- SOA: From Pattern to Production
- Lessons From the Front Line – Building Interoperable Web Services
- Developing E-Commerce Applications with Web Services
- Developing Enterprise Class Web Services
- Orchestrating FORCEnet Engagement Packs with BPEL for Web Services
- CPI: A Global Integrated Problem Tracking and Resolution System using Java Web Services
- Using Service-Oriented Architecture and Web Services to Issue Business Licenses in the District of Columbia
- Developing Web Services with Eclipse
- The Interoperability Challenge of Web Services Security Standards
- Securing Web Services with WS-Security
- Anatomy of a Web Services Attack
- Using a Mobile Phone as an SSO Authentication Device in SOA Solutions
- XML Content Attacks
- XACML and Agnostic Authorities

Register Today: www.sys-con/edge2005east/registernew.cfm

Come and see what everyone is talking about…

The Application Server Shootout!

Find out which server best fits your performance and ROI metrics

Join us in delivering the latest, freshest, and most proven Web services solutions…

At the *Fifth Annual Web Services Edge 2005 East – International Conference & Expo* as we bring together IT professionals, developers, policy makers, industry leaders and academics to share information and exchange ideas on technology trends and best practices in secure Web services and related topics.

Exhibit in *The Web Services Pavilion* and see thousands of buyers!

Becoming a Web Services Edge Exhibitor, Sponsor or Partner offers you a unique opportunity to present your organization's message to a targeted audience of Web services professionals. Make your plans now to reach the most qualified software developers, engineers, system architects, analysts, consultants, group leaders, and C-level managers responsible for Web services, initiatives, deployment, development and management at the region's best-known IT business address– The Hynes Convention Center in Boston.

3-Day Conference & Education Program features:

- Daily keynotes from companies building successful and secure Web services
- Daily keynote panels from each technology track
- Over 60 sessions and seminars to choose from
- Daily training programs that will cover Web Service Security, J2EE, and ASP.NET
- FREE full-day tutorials on .NET, J2EE, MX, and WebSphere
- Opening night reception

PRODUCED BY SYS-CON EVENTS

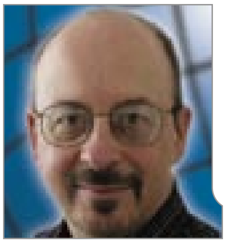
www.sys-con.com/edge

Sponsored by: wldj JDJ WebServices SD Times .NET Journal Information Storage+Security Journal WebSphere Journal XML Journal MX eaiIndustry asp.netPRO

All brand and product names mentioned above are trade names, service marks or trademarks of their respective companies.

Fire Your QA Department and Increase the Quality of Your Software

BY MIKE TAYLOR



For years you've heard some variation of the message communicated by the title of this article. The answer always involved getting developers to change their processes, think harder, use better methodologies, create better architectures, and the like…honestly, how well has that worked for you and your organization?

The challenge of changing human behavior, the complexity induced by far-flung development teams, downsizing, outsourcing, open source, and feature-rich software systems (not to mention the development environments used to create them) all increase the difficulty of building high-quality software systems on time and within budget.

Assuming that you and your development team(s) don't reside on another planet, you already know about the problems induced by the above. What you may not know is how software development technology has advanced to the point where a whole lot of quality and efficiency can be "automatically" built in while your teams do development.

Ever heard of *Continuous Collaborative Code Analysis* (C^{3}A)? No? That's okay since it's a new term we're introducing here. It may be a new term, but the ideas it embodies and the technology to do it are well proven and being used today in leading-edge enterprise development organizations around the world.

A primary objective of C^{3}A is to aid developers in finding and fixing problems in their code earlier in the development process. Pushing the resolution of quality issues into your QA and Testing organizations, or worst case, onto your customers, is at best expensive and can certainly be damaging to your reputation. It's common knowledge that if problems can be found and fixed early in the development process, it will result in better software, lower costs, faster time-to-market, and most importantly, happier users.

C^{3}A uses leading-edge technology to, in effect, put an experienced software quality mentor on the shoulder of each developer, providing expert guidance with respect to the code under development…and it happens while that code is being typed for the very first time. C^{3}A is like an automated, personal, real-time code review. It happens either continuously as code is written or "on-demand" when the developer chooses to activate it.

Continuous Collaborative Code Analysis begins with a powerful set of code standards, quality metrics, and best practices that have been encoded into your development toolset. These "rules" can be based on industry standards, company standards, and/or accepted best practices. A critical aspect is that, in addition to the hundreds of rules that typically come with C^{3}A products, you should be able to create new rules and customize existing rules so they directly meet your organization's specific needs.

Modern software development environments like Eclipse (www.eclipse.org) and IBM WebSphere Studio have a powerful capability called "quick fix" that is linked to C^{3}A systems so that when a problem is detected the developer is automatically offered a corrective "fix." No complex debugging process or searching through voluminous documentation; many problems are corrected totally automatically, long before the code is turned over to your QA group.

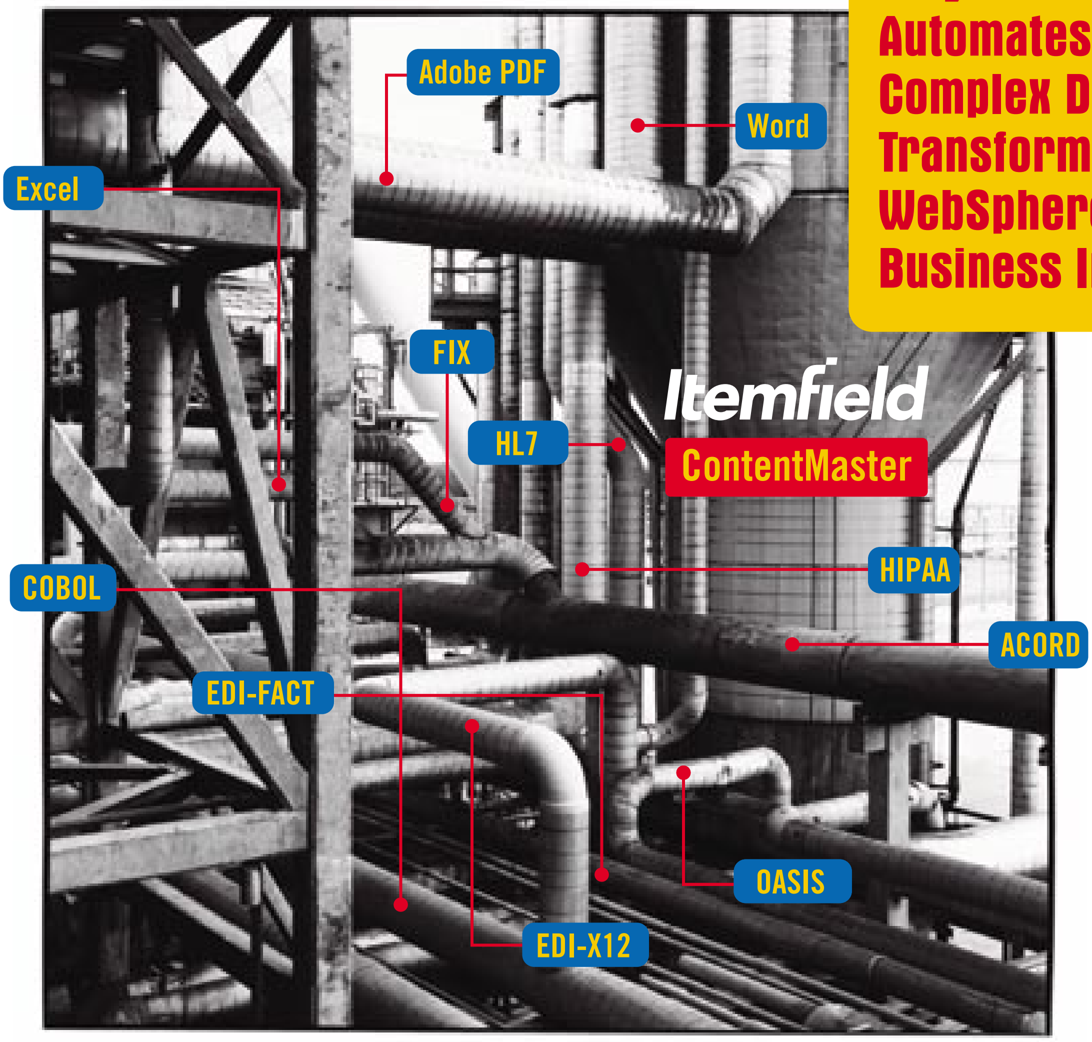
In addition to sophisticated analytical capabilities, a key element of successful C^{3}A usage is collaboration technology that enables easy communication of rules to all developers and the automated generation of reports that can be shared with managers and other team members. The most sophisticated systems allow automatic distribution and enforcement of chosen rule sets across a team of developers and projects. Lead developers, and those given the authority, can actually load approved rule sets directly into the development environments of those under their purview, assuring that the latest standards are in use and consistently enforced. Collaboration technology can reach out and unite your developers around a common set of standards whether they are down the hall or across an ocean.

A less obvious, but very important, benefit of C^{3}A is that it teaches developers to write better code. As the automated analysis cues a developer on issues, that developer quickly learns how to write good code the first time and avoid that problem in the future. Whether in Bangor or Bangalore, developers universally are very smart people who quickly internalize advice that leads to increased quality.

So can you eliminate your QA department? Probably not; but by implementing Continuous Collaborative Code Analysis you may be able to refocus it, putting part of its emphasis on defining and developing the rule sets and standards that will result in higher-quality code and lower costs for your organization and more satisfied customers. And you may find that you have some experienced QA people that would revel in the opportunity to join your development team…and that's a win for everyone.

Mike Taylor is president and CEO of Instantiations, Inc. For more information on Instantiations, please visit www.instantiations.com. mike_taylor@instantiations.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 in Business Integration”* and to view the ContentMaster Solution for WebSphere Business Integration demo.

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