PART 2: INTERVIEW WITH THE FATHER OF WEBSPHERE PAGE 30

# WebSphere® DEVELOPER'S JOURNAL

*The World's Leading Independent WebSphere Developer Resource*

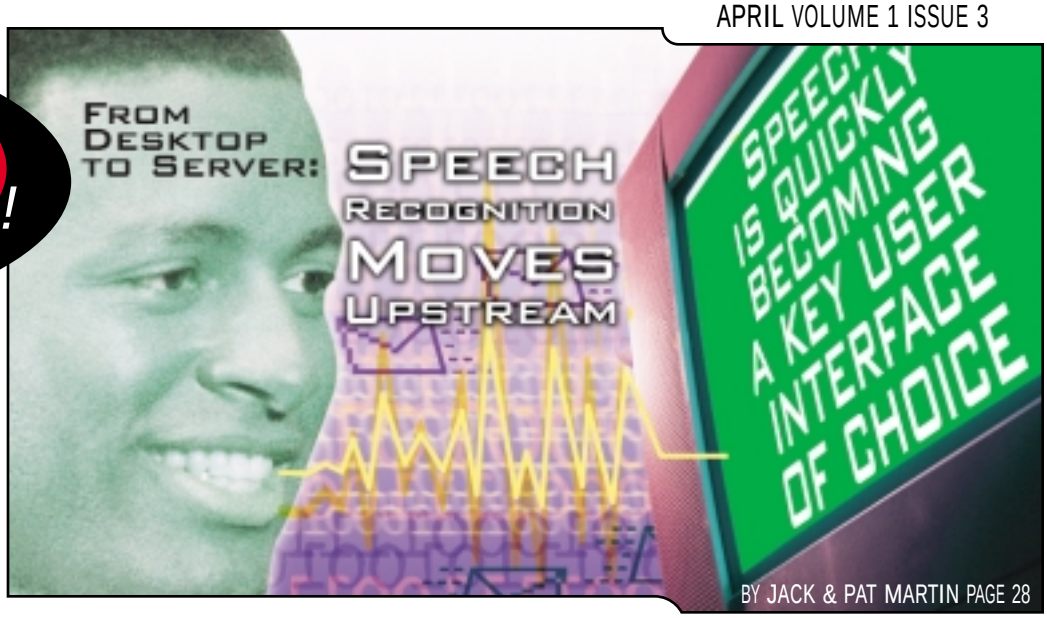
WebSphereDevelopersJournal.com

APRIL VOLUME 1 ISSUE 3





SYS-CON MEDIA




BY JACK & PAT MARTIN PAGE 28

WebSphere® DEVELOPER'S JOURNAL

ADVISORY BOARD
Richard Arone, James Martin, Greg Williams

EDITOR-IN-CHIEF JACK MARTIN
EDITORIAL DIRECTOR JEREMY GEELAN
Executive Editor GAIL SCHULTZ
Contibuting Editors PATTI MARTIN
JAMES MARTIN
Special Projects Editor RICHARD ARONE
MANAGING EDITOR CHERYL VAN SISE
SENIOR EDITOR M'LOU PINKHAM
EDITOR NANCY VALENTINE
ASSOCIATE EDITORS JAMIE MATUSOW
JEAN CASSIDY

WRITERS IN THIS ISSUE
Ron Ben-Natan, Joe Farsetta, Craig Flannagan, Greg Flurry, Jay Johnson, Kimberlee Kemble, Jack Martin, Jim Martin, Patti Martin, Jasmine Noel, Prasad Thammineni, Lou Van de Water

SUBSCRIPTIONS
For subscriptions and requests for Bulk Orders, please send your letters to Subscription Department.
SUBSCRIBE@SYS-CON.COM
Cover Price: $15/Issue Domestic: $49/YR (12 Issues)
Canada/Mexico: $169/YR Overseas: $179/YR
(U.S. Banks or Money Orders)

PUBLISHER, PRESIDENT AND CEO FUAT A. KIRCAALI
VP, PRODUCTION JIM MORGAN
SENIOR VP, SALES & MARKETING CARMEN GONZALEZ
VP, SALES & MARKETING MILES SILVERMAN
VP, EVENTS CATHY WALTERS
VP, BUSINESS DEVELOPMENT GRISHA DAVIDA
CHIEF OPERATING OFFICER MARK HARABEDIAN
CHIEF FINANCIAL OFFICER BRUCE KANNER
ASSISTANT CONTROLLER JUDITH CALNAN
ACCOUNTS PAYABLE JOAN LAROSE
ACCOUNTS RECEIVABLE JAN BRAIDECH
ACCOUNTING CLERK BETTY WHITE
ADVERTISING ACCOUNT MANAGERS MEGAN RING
ROBYN FORMA
ASSOCIATE SALES MANAGERS CARRIE GEBERT
ALISA CATALANO
KRISTIN KUHNLE
LEAH HITTMAN
CONFERENCE MANAGER MICHAEL LYNCH
EXECUTIVES, EXHIBITS MICHAEL PESICK
RICHARD ANDERSON
ART DIRECTOR ALEX BOTERO
ASSOCIATE ART DIRECTORS AARATHI VENKATARAMAN
LOUIS CUFFARI
CATHRYN BURAK
RICHARD SILVERBERG
ASSISTANT ART DIRECTOR TAMI BEATTY
WEBMASTER ROBERT DIAMOND
WEB DESIGNERS STEPHEN KILMURRAY
CHRISTOPHER CROCE
CONTENT EDITOR LIN GOETZ
JDJSTORE.COM ANTHONY D. SPITZER
CUSTOMER SERVICE ANTHONY D. SPITZER

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 Developer's Journal (ISSN# 1535-6914)
is published monthly (12 times a year)
Postmaster send address changes to:
WebSphere Developer's Journal, SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645



SYS-CON MEDIA



# The Good Thing About Being Capitalists

BY JACK MARTIN

"Wealth is the result of man's ability to think." Ayn Rand, *Atlas Shrugged*.

In last month's issue Microsoft cited some close-up magic white papers in an attempt to claim that WebSphere was fifteen times more expensive and one-fortieth as fast as .NET. Give me a break.

That left me shaking my head in disgust. Does Microsoft think this is a format to manipulate facts and figures? If anyone can sort out the advantages of WebSphere and disadvantages of .NET, it's us! And that's the good thing about being capitalists.

In September 2001, eBay (you know them, they're the biggest and probably most viable consumer Web site) announced that after running a three-month benchmark test using their working model, WebSphere came out on top. They tested against WebLogic and .NET.

Chuck Geiger of eBay made the announcement that the analysis considered the general cost per transaction as it was derived from the *total cost* of hardware, OS, software, and development productivity as it related to time-to-market. The software was benchmarked for sessions, threads, and EJB performance exponentially and for horizontal and vertical scalability – clustered and nonclustered.

Now, I don't care about some lab example and I agree with Stefan Van Overtveldt that Microsoft is, in fact, missing the point. Give me real-world data with real-world results. When a powerhouse like eBay makes the time and resource investment to run WebSphere and its competitors through their paces, it's those results that I want to base my decisions on.

I feel better after getting that out of the way. Back to being capitalists: capitalism stresses freedom of individual economic enterprise. In this wonderful economy we vote with our dollars and don't have to take any anyone's attempt at marketing propaganda as fact. The .NET story is just beginning and I predict it will entertain and annoy until the customer has had the last word. Nothing will be force-fed to us.

Speaking of being fed, I had the opportunity to attend a duck roast that yielded surprising results. Check out "Why they Buy"(see page 20) and learn what we discovered about using professionals to assist where in-house skills may be lacking. A big part of IT success has been due to a combination of training in-house resources and obtaining assistance in planning and implementation from skilled vendors.

In the very-skilled department, we have the second part of Don Ferguson's interview. In this issue he begins to lay out some of IBM's plans for WebSphere. How it will play in Web services and what standards are being promoted. This is hot stuff and we still have another installment from Don on WebSphere opportunities in the pervasive market for next month.

Debuting this month is a column on Web services. Ron Ben-Natan starts us off with a brief history and then covers the standards upon which Web services are based. This is very apropos. After speaking with Don Ferguson I'm firmly convinced that my efforts are well served by committing to the success of WebSphere for Web services.

In my off hours (from editorial duties), I live and breathe WebSphere, in particular Portal Server. Whether it's in training people or assisting with technology assessments, Web services is on everyone's mind. The beauty of open standards for cross-platform deployment is truly a high value for technologists. I feel that with these standards and the commitment of industry leaders to work in a spirit of cooperation, the promise that Web services is making will in fact be delivered.

Continuing in this direction, ***WSDJ*** is working on a Portal extravaganza to be featured in the coming months. We'll turn the spotlight onto WebSphere Portal Server, portlets, and the myriad portlet developers. If you want to get your head around this expanding market, make plans to get your hands on the June issue.

Finally, this capitalist is off to make his way through the opportunities available in this open market. Good luck and success to you too!

ABOUT THE AUTHOR... Jack Martin, editor-in-chief of ***WebSphere Developer's 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. E-MAIL... jack@sys-con.com

## *Creating Web services from existing JavaBean components*

# Web Services Development with WebSphere Studio Application Developer

BY GREG FLURRY

WebSphere Studio Application Developer (WSAD) is IBM's newest J2EE e-business application development tool. WSAD, based on the open-source Eclipse tools platform, offers the e-business application developer a number of capabilities, including Web application development and testing, XML development and testing, and Web services development and testing, the focus of this article .

ABOUT THE AUTHOR

Greg Flurry is a member of the IBM Emerging Technologies area in Software Group. His current responsibilities include introducing Web services technologies into the IBM WebSphere product family.

E-MAIL

flurry@us.ibm.com

WSAD supports all phases of Web service development, including the initial development of components such as JavaBeans or Enterprise JavaBeans, the transformation of those components into Web services, the testing of the Web services, and the publication of the Web services in a UDDI-compliant registry.

WSAD supports a number of models for creating Web services. For example, using an XML document for parameters and return values is a common model. (You can find additional information about using WSAD for this model in the References section.) This article examines WSAD support for an alternative in which a JavaBean is used for parameters and return values – a natural model for creating Web services from existing JavaBean components.

We'll start with an existing JavaBean that extracts weather forecasts from the National Weather Service and returns JavaBeans from its methods. I assume you're familiar with the basics of developing Java components in WSAD and will offer details only on the Web services–related aspects of WSAD, in particular version 4.0.2.

### Getting Started

We'll assume that an existing Web project contains the JavaBeans to be used for the weather forecast Web service. Figure 1 shows the *Packages* view of the Java perspective for the WF Web project, which contains the five JavaBeans that contribute to the Web service:

- ***WeatherForecast:*** The Web service "business logic"; it accesses the National Weather Service (NWS), extracts forecast information, and formats the information appropriately
- ***States:*** Holds the states covered by the NWS
- ***Cities:*** Holds the cities for a particular state covered by the NWS
- ***Forecast:*** Holds the forecast for one day for a particular city and state combination
- ***Forecasts:*** Holds the forecasts for several days for a particular city and state combination

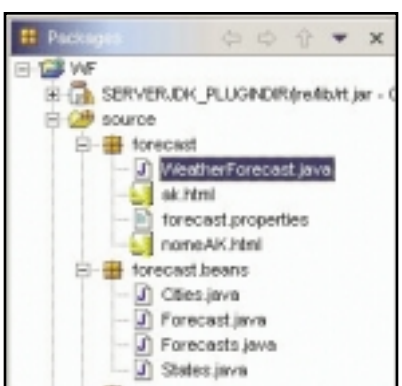



**FIG. 1: JAVABEANS FOR WEATHER FORECAST WEB SERVICE**

Listing 1 shows the States, Cities, Forecast, and Forecasts beans. (All of the listings for this article can be found at www.sys-con.com/websphere/sourcec.cfm.) Listing 2 shows the WeatherForecast bean. Note that the getStates()method returns a States bean, the getCities()method returns a Cities bean, and the getForecast() method returns a Forecasts bean. For the sake of brevity, Listing 2 does not show the complete WeatherForecast bean. Note the use of caching in the getCities()and the getForecast() methods; this reduces the network traffic for repeated requests for forecasts from the same state.

### Creating a Web Service from the Bean

We will use the WSAD Web Service wizard (see Figure 2) to create the Web service from the Weather/Forecast bean. To create the Web service, go to the *Web* perspective, select the WF project and right-click to select New>Web Service. You will see the dialog in Figure 2. Since you're going to create a Web service from a JavaBean

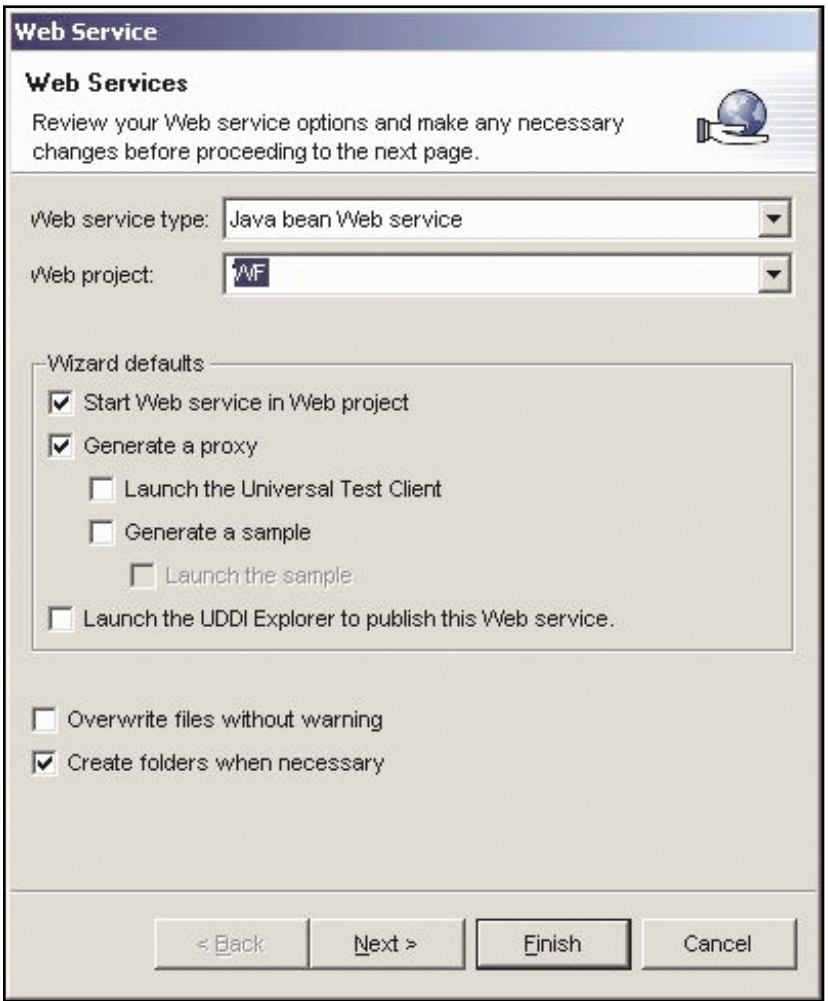



**FIG. 2: WEB SERVICES WIZARD: INITIAL DIALOG**

and the bean resides in the WF Web project, simply hit Next to go to the next panel. In the resulting *JavaBean Selection* panel, hit the Browse files button.

In the *Browse Files* dialog (see Figure 3), expand the source folder and then expand the forecast folder and select WeatherForecast.java, then select OK. Hit Next to go to the next panel. You can accept the defaults presented by the Web Service wizard until the *Java Bean Identity* panel, shown in Figure 4.

If this were more than just an exercise, you would also want to ensure that the Web service URI is unique, but that isn't necessary in this case. This panel allows the developer to use HTTP-like scope for a Web service, i.e., request, session, or application scope. The WeatherForecast bean is designed to operate with session scope, so you must change the Scope field to "Session". Session scope ensures a new instance for each user, allowing caching of information in the WeatherForecast bean to work properly. The default values of the rest of the fields are acceptable, so select Next.

You will now see the *JavaBeans Methods* panel (see Figure 5), which allows you to select which methods in a JavaBean to expose in the Web service. In this case, we want to expose all the methods, so no changes are needed. Notice that SOAP encoding is automatically selected for both the input strings and the output JavaBeans. This is exactly as desired, so select Next.

You will now see the *Binding Proxy Generation* panel (see Figure 6), which allows you to select which type of binding you want for the Web service. In this case, we want the SOAP binding and proxy, so no changes are needed. You can select Finish.

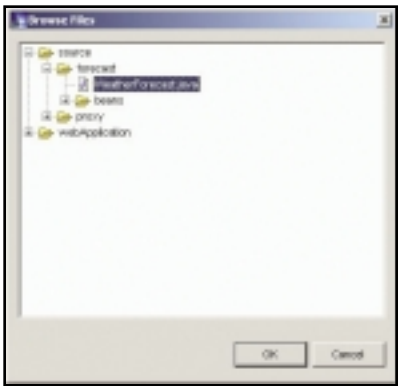


**FIG. 3: WEB SERVICES WIZARD: BROWSE FILES DIALOG**

The Web Service wizard now generates a proxy for accessing the Web service, the WSDL files describing the Web service and a SOAP deployment descriptor for deploying the Web service in the WebSphere Application Server. It also deploys the Web service in the WebSphere Test Environment provided by WSAD and starts the WebSphere Test Environment, which makes the Web service available for use.

WSAD generates a two-part WSDL description of the Web service, in accordance with the recommended practice for publishing services in a UDDI-compliant registry. The "binding" WSDL file describes the abstract interface definition and the binding of that definition to a concrete transport; the "service" WSDL file describes the actual endpoint information for the Web service. It's instructive to examine the WSDL generated by WSAD to see how the returned beans are described. Listing 3 shows the excerpts from the "binding" file relevant to Forecasts, the most complex of the JavaBeans used by the Web service.

The response from the getForecast() method is described as type "forecast.beans.Forecasts" in the namespace defined by the schema file generated by WSAD (WeatherForecast-schema.xsd). The schema file is generated independently of the WSDL "binding" file to make reuse of the type information easier. Listing 4 shows the excerpts

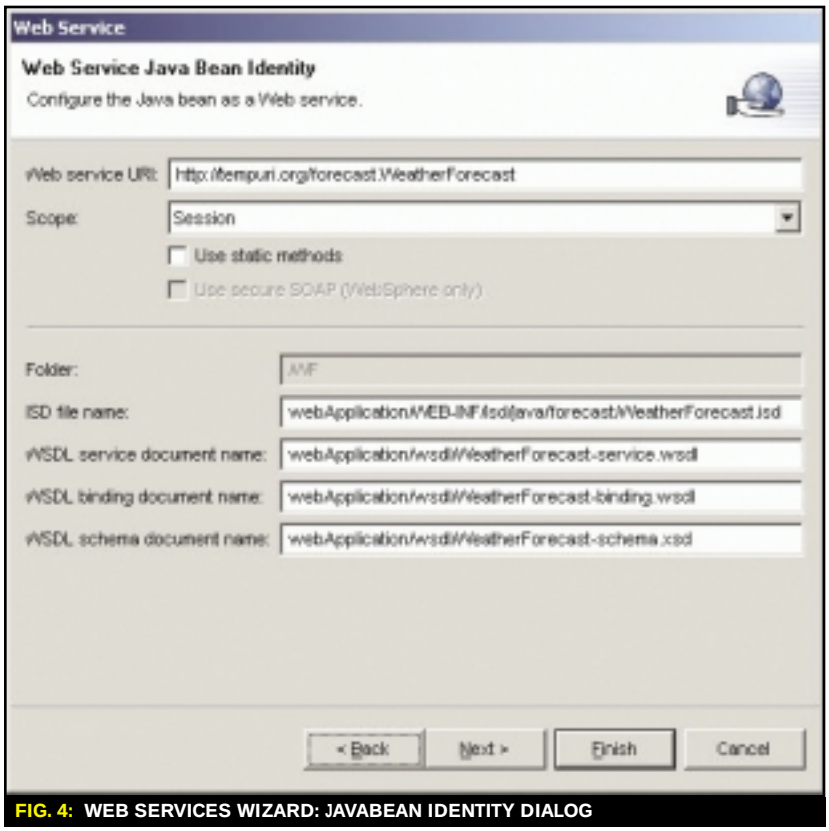



**FIG. 4: WEB SERVICES WIZARD: JAVABEAN IDENTITY DIALOG**

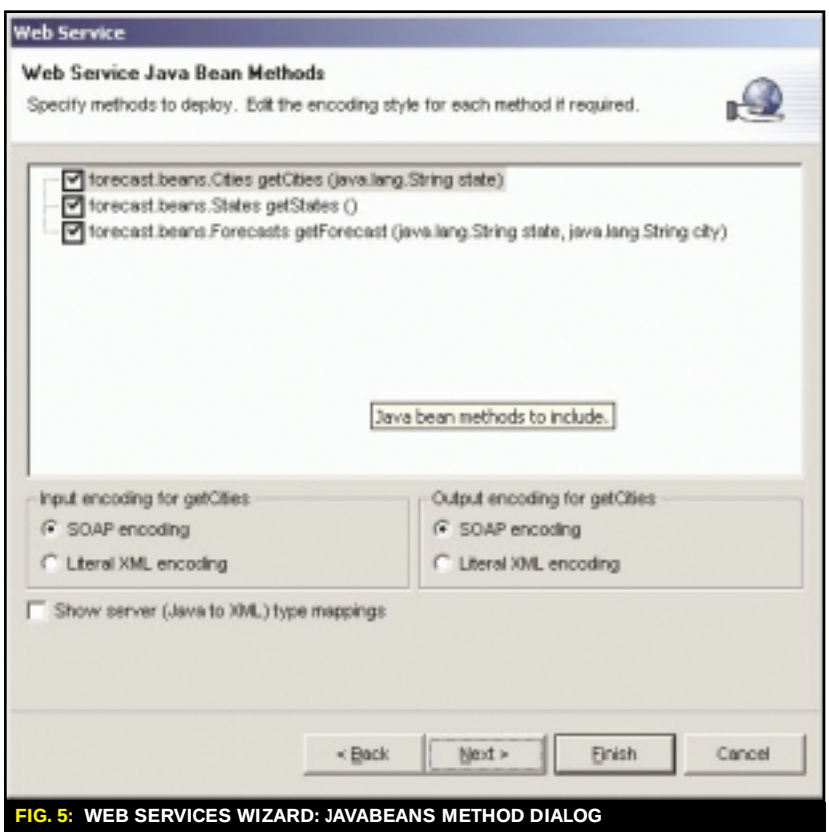



**FIG. 5: WEB SERVICES WIZARD: JAVABEANS METHOD DIALOG**

from the schema file relevant to the Forecasts bean. The schema defines the "forecast.beans.Forecast" type and an array derived from that type and then defines the "forecast.beans.Forecasts" type using the array type.

## Testing the Web Service

Listing 5 shows a simple test program for the Web service; it uses the proxy generated by the WSAD Web Services wizard. The test program is constructed to use the WSAD TCP/IP Monitor (see the WSAD help for more information), so we can examine the actual requests and responses. The WebSphere Test Environment by default listens on port 8080, and the monitor can be configured to listen on port 8081 and pass requests on to port 8080. The statement marked with a ">" in the first column causes the proxy to contact the Web service on port 8081, thus routing requests and responses through the monitor.

You can run the program with the Web service configured to run against some "canned data" used when the Web service doesn't have access to the Internet (access is controlled by the OFFLINE property in a properties file). Note that you will have to add the XSDBEANS and the XSD_BEAN_RUNTIME variables, and the mail.jar and activation.jar files to the Java Build Path for the project containing the test program. The test program calls all three operations on the Web service. Listing 6 shows that for the getStates() operation, the request has no input parameter, and the response has an array of strings as the return value. Note the "Set-Cookie" HTTP header in the response that establishes the session.

Listing 7 shows the getForecast() request, with an input parameter containing the city and state for which to retrieve a forecast, and the response, with complex type containing the forecast as the return value. Note the "Cookie" HTTP header in the request to maintain the session.

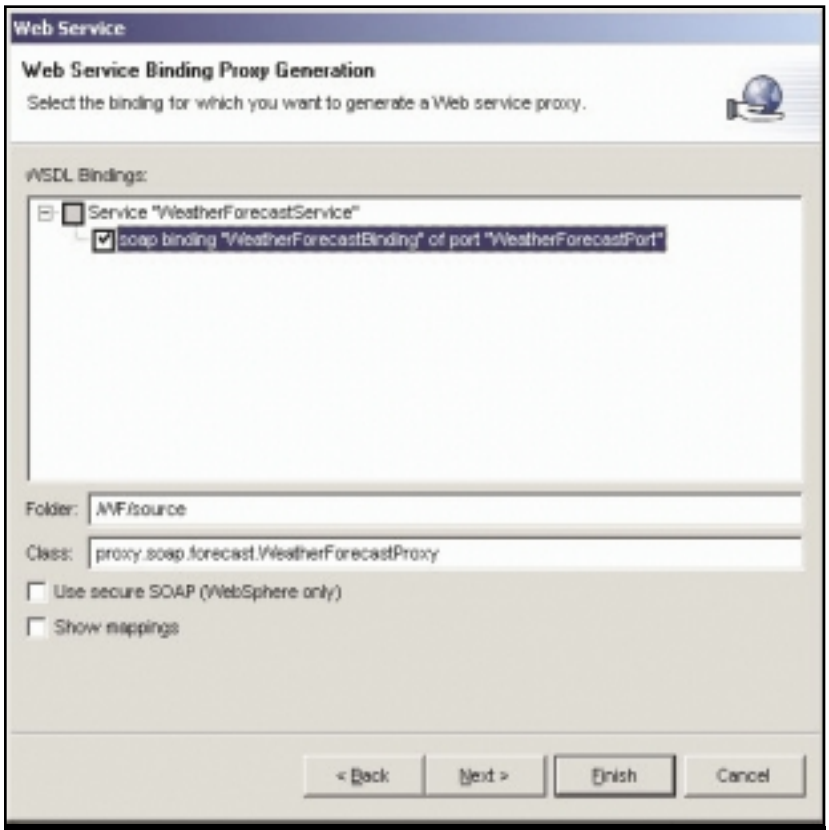



**FIG. 6: WEB SERVICES WIZARD: BINDING PROXY GENERATION DIALOG**

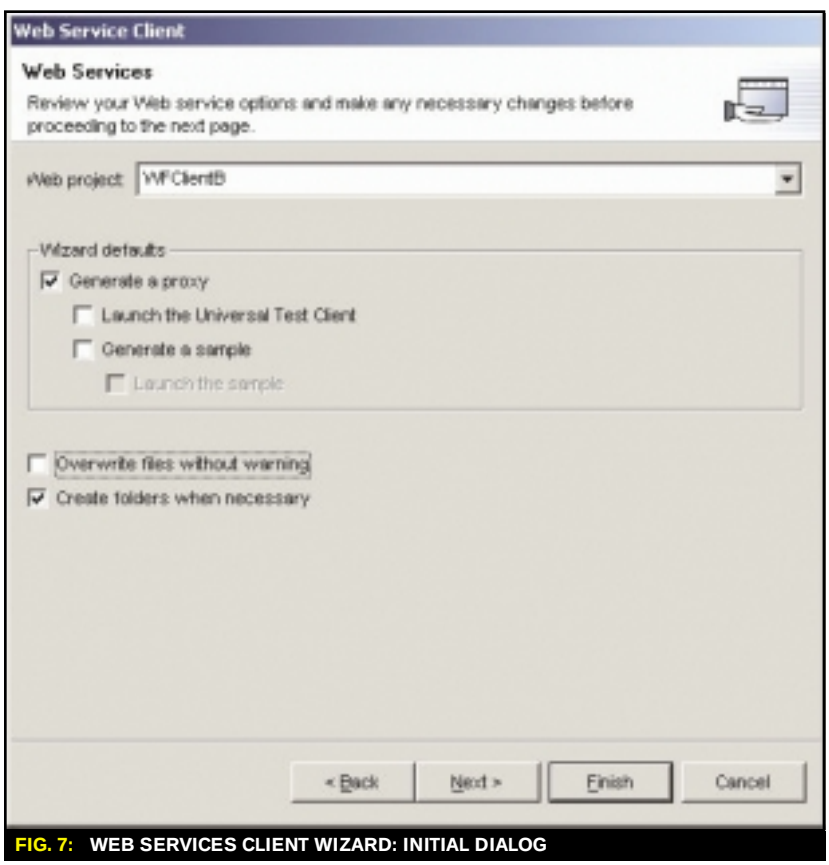



**FIG. 7: WEB SERVICES CLIENT WIZARD: INITIAL DIALOG**

The test program produces the following result:
The first state: AK
The first city: Anchorage
The first forecast: Wednesday – Sunny, high 79, 0% chance of precipitation.

## Using the Web Service

Now we will create a client of the Web service using only the WSDL, as would be done by finding the WSDL for a Web service in a UDDI registry. First create a new Web project named WFClientB; you can put the new client in the same EAR project as the WF project, or in a different EAR project. Copy WeatherForecast-service.wsdl from the WF project into the WFClientB project; this simulates importing a WSDL file from a UDDI registry. Next, select the WeatherForecast-service.wsdl file and right-click on it and select New>Other. On the resulting panel, select Web Services on the left side and Web Services Client on the right, and then select Next. You will see the Web Services Client wizard shown in Figure 7. Simply select Finish.

The result produced by WSAD is shown in Figure 8. In addition to the expected proxy, notice the beans in the mappings package. These beans are returned by the generated proxy and correspond to the beans returned by the Web service. The mapping beans deserialize the unique SOAP responses, but are derived from a common bean deserializer. Each mapping bean has the same method signature as the corresponding original beans. For example, the Forecast_beans_CitiesContentType bean has the same property methods as the Cities bean, i.e., setState(), getState(), setCities(), and getCities(). A client using the generated proxy must use the beans in the mappings package rather than the original beans. Note that this is quite convenient because generally a client won't have access to the original beans, especially if the client used WSDL from a UDDI registry to produce the proxy used to access the Web service.

Listing 8 shows how the test program above must be modified to use the new proxy and the mapping beans. The import statements must refer to the new proxy and the mapping beans, and the mapping beans must be used instead of the original beans; otherwise, the revised test program is identical to the original. The changed statements are marked in bold type.

When you run the revised program, you'll get the expected result from the first test program (assuming the same configuration properties). Likewise, the details of the requests and responses are the same:

The first state: AK
The first city: Anchorage
The first forecast: Wednesday - Sunny, high 79, 0% chance of precipitation.

## Summary

This article showed that WebSphere Studio Application Developer Web service tools can successfully handle somewhat complex parameters and return values. See the "Readme" file installed with WSAD and also the Mapping Properties topic in the WSAD help information for limitations on the handling of simple and complex data types. If you find something WSAD cannot handle, you can always create your own serializers and deserializers; WSAD allows you to specify the (de)serializers in the Java-to-XML mappings and XML-to-Java mappings panels in the Web Service wizard, but that is beyond the scope of this article.

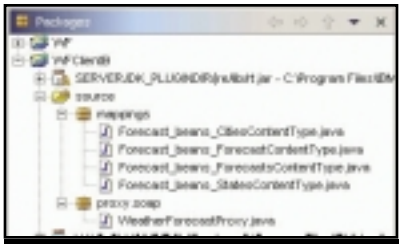


**FIG. 8: MAPPING JAVABEANS PRODUCED WEB SERVICES CLIENT WIZARD**

## References


- *Eclipse:* www.eclipse.org/
- *WebSphere Studio Application Developer:* www-3.ibm.com/software/info1/websphere/index.jsp?tab=products/studio
- *Web Services Development and Deployment with IBM Tools and Technologies, Part 1:* www7b.boulder.ibm.com/wsdd/library/techarticles/0111_flurry/part1.html
- *Web Services Development and Deployment with IBM Tools and Technologies, Part 2:* www7b.boulder.ibm.com/wsdd/library/techarticles/0111_flurry/part2.html
- *Web Services Development and Deployment with IBM Tools and Technologies, Part 3:* www7b.boulder.ibm.com/wsdd/library/techarticles/0111_flurry/part3.html
- *Web Services Development and Deployment with IBM Tools and Technologies, Part 4:* www7b.boulder.ibm.com/wsdd/library/techarticles/0111_flurry/part4.html
- *Web services zone on IBM developerWorks:* www-106.ibm.com/developerworks/webservices


LISTING 1

```
package forecast.beans;
public class States {
    protected final static String[] state =  {
        "AK","AZ","AR","CA","CO","CT",
         ...
        "WV","WI","WY"
    };

    public String[] getStates() {
        return state;
    }
}
package forecast.beans;
public class Cities {
    private String state = "none";
    private String[] cities = null;

    public void setState(String state) {
        this.state = state;
    }
    public String getState() {
        return state;
    }

    public void setCities(String[] cities) {
        this.cities = cities;
    }
    public String[] getCities() {
        return cities;
    }

    public void addCity(String city) {
        if (cities == null) {
            cities = new String[1];
            cities[0] = city;
        } else {
            String[] temp = new String[cities.length + 1];
            System.arraycopy(cities, 0, temp, 0,
cities.length);
            temp[cities.length] = city;
            cities = temp;
        }
    }
}
package forecast.beans;
public class Forecast {

    protected String period;
    protected String forecast;

    public void setPeriod(String p) {
         period = p;
    }
    public String getPeriod() {
        return period;
    }

    public void setForecast(String f) {
        forecast = f;
    }
    public String getForecast() {
        return forecast;
    }
}
package forecast.beans;
public class Forecasts {

    protected String location = null;
    protected String issued = null;
    protected Forecast[] forecasts = new Forecast[5];
    int index = 0;

    public void setLocation(String l) {
        location = l;
    }
    public String getLocation() {
```

```
        return location;
    }

    public void setIssued(String i) {
        issued = i;
    }
    public String getIssued() {
        return issued;
    }

    public void setForecasts(Forecast[] f) {
        forecasts = f;
    }
    public Forecast[] getForecasts() {
        return forecasts;
    }

    public void addForecast(Forecast f) {
        forecasts[index++] = f;
    }
}
```

LISTING 2

```
ackage forecast;
public class WeatherForecast {

    public States getStates() {
        return new States();
    }

    public Cities getCities(String state) {

        Cities returnCities = new Cities();
        returnCities.setState(state);

        int i;
        for (i=0; i<URL_MAP.length; i++) {
            if (URL_MAP[i][1].equals(state)) {
                break;
            }
        }
        String result = null;
        if (!offline) {
            result = harvestResults(URL_BASE +
              URL_MAP[i][0]);
        }
        ...
        stateCached = state;
        cityCache = findCities(result);
        Enumeration cities = cityCache.elements();
        cityCache = new Vector();
        // formulate a response
        while (cities.hasMoreElements()) {
            String [] ce = processCity((String)
             cities.nextElement());
            cityCache.addElement(ce);
            returnCities.addCity(ce[1]);
        }
        return returnCities;
    }

    public Forecasts getForecast(String state, String city)
{
        Vector tmpCities = null;
        Forecasts returnForecasts = new Forecasts();

        if (!state.equals(stateCached)) { // used cached
             city info
            getCities(state);
        }
        tmpCities = (Vector) cityCache.clone();
        // find the city url in the cache
        String url = null;
        for (int i=0; i<tmpCities.size(); i++) {
            String [] info = (String [])
             tmpCities.elementAt(i);
            if (info[1].equals(city)) {
                url = info[0];
                break;
            }
        }
        // get the forecast
        String result;
        if (url == null) {
            throw new Exception("Invalid city");
        } else {
            if (!offline) {
                result = harvestResults(url);
            } ...
        }
        // process the forecast
        int beg = result.indexOf("<PRE");
        String [] content = null;
        // get the header
        content = extractContent(result, beg, "<B>", "\n", "</F");
        returnForecasts.setLocation(content[0]);
        returnForecasts.setIssued(content[1]);
        // get the "middle" day forecasts
        beg = Integer.parseInt(content[2]) + 10;
        for (int j=0; j<4; j++) {
            // extract DOW forecast
            content = extractContent(result, beg, "<B>",
              "</FONT></B>", "<B>");
            beg = Integer.parseInt(content[2]);
            Forecast f = new Forecast();
            f.setPeriod(content[0]);
            f.setForecast(content[1]);
            returnForecasts.addForecast(f);
        }
        // get the last day forecast
        content = extractContent(result, beg, "<B>",
          "</FONT></B>", "</PRE>");
        Forecast f = new Forecast();
        f.setPeriod(content[0]);
        f.setForecast(content[1]);
        returnForecasts.addForecast(f);
        return returnForecasts;
    }
}
```

LISTING 3

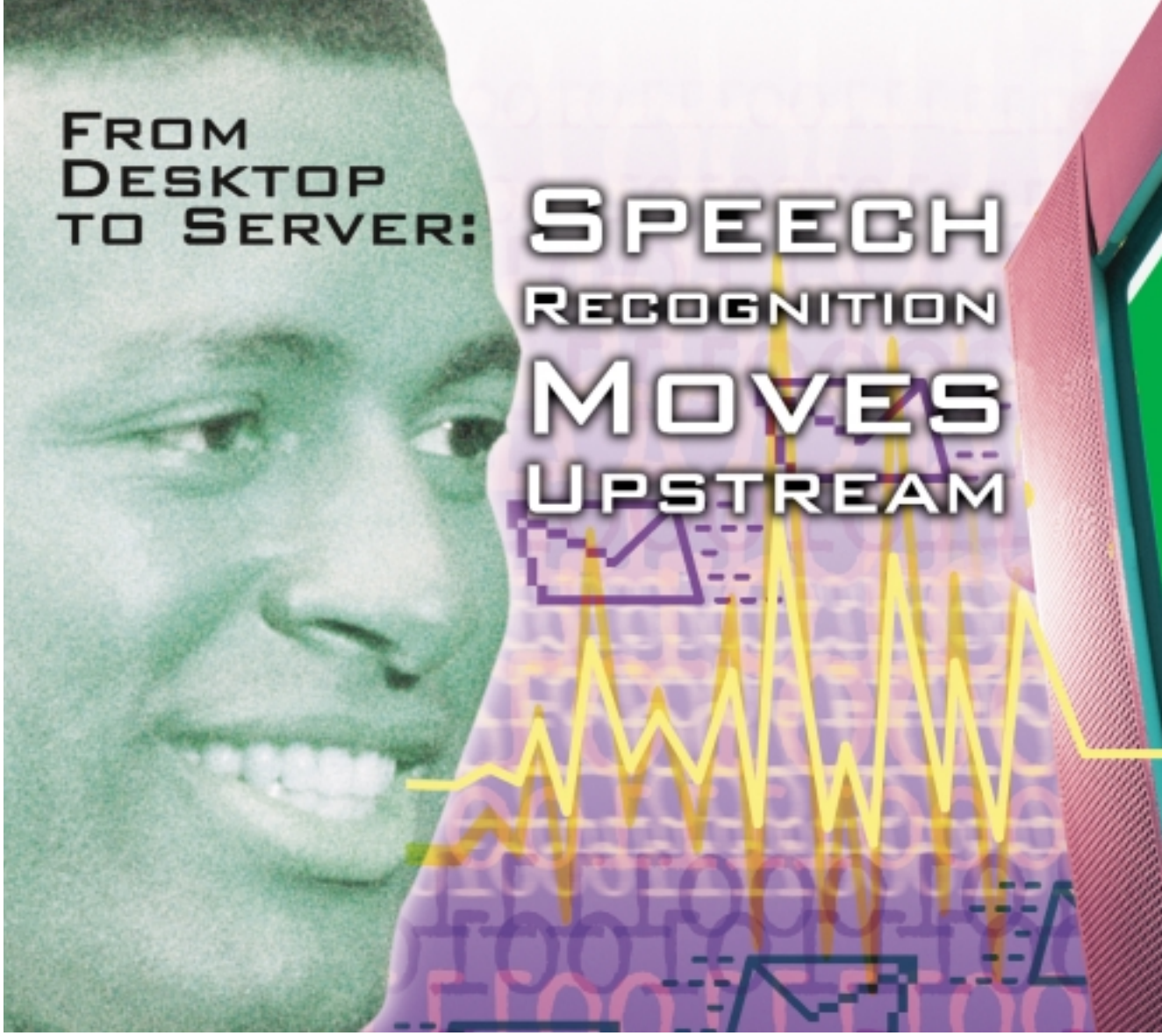
```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="WeatherForecastRemoteInterface"
  targetNamespace="http://www.weatherforecast.com/defini-
tions/WeatherForecastRemoteInterface"
  ...

xmlns:xsd1="http://www.weatherforecast.com/schemas/WeatherFor
ecastRemoteInterface" ...>
  <import
namespace="http://www.weatherforecast.com/schemas/WeatherFore
castRemoteInterface"
    location="http://localhost:8080/WF/wsdl/WeatherForecast-
schema.xsd"/>
  ...
  <message name="getForecastResponse">
    <part name="result"
type="xsd1:forecast.beans.Forecasts"/>
  </message>
  ...
</definitions>
```

LISTINGS 4 – 10

The remainder of the listings for this article can be found at www.sys-con.com/websphere/sourcec.cfm

# From Desktop to Server: Speech Recognition Moves Upstream



ABOUT THE AUTHOR

Kimberlee Kemble is program manager for Technical Marketing, IBM Pervasive Computing, Boca Raton, Florida. Currently, she coordinates education and training programs for several IBM voice products and technologies, with special focus on VoiceXML and voice user interface design. She also manages the marketing programs for the IBM WebSphere Voice Server for Transcription. Kim is active in the VoiceXML Forum Publication Review Board and Education Committee.

Speech recognition is the process by which computer-based software converts audible voice into digital text. When you think of computer-based speech recognition, most people picture someone sitting at a desk, wearing a headset microphone, dictating large volumes of text into a desktop system. But speech-recognition technology, over the past decade, has moved from the desktop to the server, from use by an individual to use by the enterprise.

Take the following examples:

*"Can you tell me what IBM stock closed at last night?"*
*"I'd like to transfer two thousand dollars from my savings to my checking account please"*
*"Connect me to Kim Kemble in Boca Raton"*
*"Subjective colon 32-year-old white female in for evaluation period Patient describes injury of left ankle period*
*"Objective colon Patient presents in mild distress and pain period Heart colon Regular rate and rhythm period"*

What do the above have in common? They're all snippets of dialog that have been extracted from actual applications that utilize speech recognition. What's more, all of these applications implement speech recognition over the telephone. That means callers can conduct business with voice-enabled automated systems over the phone, simply by using their voice.

Before we go any further, let's take a look at some basic speech terms.

- ***Dictation:*** The composing and recording of thoughts into audible voice files
- ***Transcription:*** The process of transcribing dictation into text, either manually (i.e., by typing) or by using speech recognition
- ***Command and control:*** The use of spoken words and phrases to direct an application to perform a task
- ***Speaker independence:*** The extent to which a speech recognition system must understand each individual speaker's voice characteristics to be able to process speech


BY KIMBERLEE A. KEMBLE

## A Short History

With more than 150 patents in voice recognition, IBM has been researching speech for more than 30 years. It wasn't until the early 1990s that faster hardware and improved software made speech technology practical to implement.

The first speech systems ran on large UNIX servers. Resource requirements were high. First, it required special digital signal processing (DSP) hardware to assist in the speech-recognition process, as most systems in those days simply weren't powerful enough to support very computationally intensive speech recognition. It required several hours of "training," and you also had to talk using a style known as discrete speech, where you inserted short pauses between your words ("you…had…to…talk…like…this"). Even though it was not the most natural way to speak, for people who routinely dealt with large volumes of text (doctors and lawyers, in particular), it was a significant breakthrough.

Then came continuous speech, which meant users didn't have to insert pauses between words. This continuous speech was limited to short commands and phrases, and dictation still required discrete speech. In 1993, a personal voice product, IBM's Personal Dictation System, was released for OS/2. It was one of the first commercially available, high-accuracy voice-recognition products. The following year saw the announcement of IBM VoiceType Dictation for Windows and OS/2 ("You talk, it types"). Four years later, IBM released the industry's first desktop continuous-dictation product. Users no longer had to pause between words, whether they were dictating or using commands, and could speak at a natural pace. Training requirements were decreased from several hours to several minutes, and recognition accuracy continued to improve.

Over the past several years, voice technology has moved from the desktop to the enterprise in the form of voice middleware. Voice middleware encompasses platforms and applications that run on servers, such as IBM's WebSphere Voice Server, serving hundreds or thousands of customers via the telephone or Internet. Generally, these server-based voice applications are written to service a limited vocabulary and a large number of users, such as bank customers. No "training" of a caller's voice is required.

An example of voice middleware is a customer service application that uses Web technology. This new application might give customers a voice interface to the same Web application content that had previously been accessible only through the Internet. For instance, a customer may now call a voice-enabled Web application server at a brokerage firm and complete a trade – without operator assistance. This is done by speaking commands and listening to the same information that might normally be "seen" using a browser on a PC or workstation.

Another example of voice middleware is a voice-enabled flight information system, where a caller can receive flight information directly (such as late arrivals) rather than waiting on hold to speak to an agent. Today a caller can simply call a number, state the flight number and city, and receive the flight information audibly over the phone.

At the same time that speech recognition was finding its way to the enterprise, it also moved to the device. Embedded speech technology now enables mobile devices, which are typically constrained by the amount and type of system resources available (memory, processor speed, and storage space) to deploy voice. Speech recognition can now be used on devices – providing low-resource, small vocabulary command-and-control speech recognition in a variety of languages. The software also supports a variety of real-time operating systems and microprocessors, making the development of robust mobile speech solutions easy and practical for both device and application developers.

The convergence of computers with telephones and handheld devices continues. The human voice becomes a Web browser. Surf the Web in the car while a text-to-speech application reads back the content, then tell the car to turn on the radio when finished. Check the status of an order without having to punch a telephone keypad. Use a PDA to conduct a banking transaction without touching the keyboard. What could be easier than just talking?

## Trends and Directions

Voice technology, which for a long time had been confined to research, is now putting a natural interface on the computing environment, from end-user devices to infrastructure, crossing national boundaries. Worldwide spending on voice recognition will reach $41 billion by 2005,

according to the Kelsey Group, a market research firm. There are several forces driving the growth:

- Companies view voice as a way to improve their call-center service while reducing costs. Voice recognition allows companies to use automation to serve customers over the phone, 24/7, without subjecting them to hold times or requiring them to respond to rigidly structured menus. Then there are the business savings: a typical customer service call costs $5 to $10 to support; automated voice recognition can lower that to 10 to 30 cents. The market-research firm Datamonitor says call center managers are seeing an increase in customer acceptance of automation and self-service, along with cost savings.
- The rise of *telematics*, which combines computers and wireless telecommunications with motor vehicles to provide customized services such as driving directions, emergency roadside assistance, personalized news, sports and weather information, and access to e-mail and other productivity tools. The Kelsey Group predicts U.S. and European spending on telematics will exceed $6.4 billion by 2006
- Companies looking to voice-enable the Internet and their IT establishments, whether it's providing information to consumers through "voice portals" or allowing employees to access corporate databases over the phone through spoken commands
- The ability to squeeze convenient speech recognition into ever-smaller devices, such as phones, PDAs, and other mobile devices

This is happening not only in the U.S., but also across the globe. For the most part, companies looking to deploy voice face a lot of similar concerns. They want to know what business applications will bring more value to their customers and set them apart from their competition.

Key to the growth of voice is VoiceXML. VoiceXML has been the catalyst for the deployment of speaker-independent, limited-vocabulary automatic speech-recognition systems in recent years. It has allowed for applications such as voice portals and speech-enabled call centers to grow, paving the way for Web access via phone. This allows call centers to automate simple customer requests, and use their live agents for more complicated tasks and inquiries, thus making more efficient use of them. The significance of VoiceXML is manyfold:

- It's an industry-standard language (sponsored by the VoiceXML Forum – www.voicexml.org), and it's designed to leverage the skills of the Web development community.
- It makes speech application programming much more accessible to developers. If you know HTML or XML, VoiceXML will look very familiar to you. Not only does it use the well known element-attribute (or tag) style, it utilizes the very same infrastructure that exists within the Web today.
- VoiceXML takes speech application programming to a higher level. Previously, you had to know quite a bit about speech recognition and possibly even linguistics to be able to develop an effective speech application. With VoiceXML, the system does a lot of things for your application "under the covers" so your application can focus on the task at hand.

Another key factor in the growth of voice is Natural Language Understanding (NLU) technology. NLU enables applications to implement a conversational interaction with end users. No longer do users have to speak in short commands and phrases; they can now speak almost as if they were talking to a live agent. NLU combines breakthrough research and development in the fields of voice recognition, linguistics, statistics, human factors, and artificial intelligence. When NLU is combined with speech recognition, the user is able to speak in a more unstructured, conversational style, resulting in a more comfortable and productive user experience. For example:

*System:* "Welcome to the Mutual Funds Trading System. How can I help you?"
*User:* "I'd like to buy more Fidelity, please."
*System:* "You currently own 500 shares of Fidelity Magellan. How many shares would you like to buy?"
*User:* "What's the going price?"
*System:* "The current price of Fidelity Magellan is…What else can I do for you?
*User:* "Hmmmmm…I think I'll buy 300 more"
*System:* "Purchasing 300 shares of Fidelity Magellan at

market price…please say 'yes' to confirm."
*User:* "No, make that $2,000."
*System:* "Purchasing 2000 dollars of Fidelity Magellan at market price…please say yes to confirm."
*User:* "Do it."
*System:* "2000 dollars, or 125.25 shares, of Fidelity Magellan purchased at market price. What else can I do for you?"

When the user says "I'd like to buy more Fidelity please," the software must extract the meaning from what was said. Clearly, there are extra words in this sentence – words that don't affect the meaning of the sentence – and the software must extract the pertinent words from the sentence to determine its meaning. In this sentence, the key concepts are "buy" (an action), "more" (implying a fund the user already owns), and "Fidelity" (a fund company).

As you can see from its response, the application determined that the user owns Fidelity Magellan, and then asks how many shares the user would like to buy. In this example, the user responds NOT with the number of shares to buy, but with a totally different question, "What's the going price?" This means the caller is in control, rather than being driven by long menus. At this point, the application must "switch gears" and obtain the current price of a fund. From the context, or what was previously said by the user, the application knows that the user wants the price of Fidelity Magellan. You can see from the example how NLU technology can make the interaction between a user and the system more intuitive and effective.

And NLU is not just a vision of the future. One of the most sophisticated uses of NLU technology is being deployed by 401K management company T. Rowe Price. Their system is being rolled out to one million users who will be able to manage their retirement accounts using the enhanced Plan Account Line. The system doesn't require a caller to use a particular script. "We believe that most callers will save at least 30 percent of their time," says Heidi Walsh, vice president and senior marketing manager.

Developments in speech-recognition technology haven't been limited to telephony. Dictation has also leapt from PCs to the server. Most recently, IBM announced the WebSphere Voice Server for Transcription. This offering was introduced in early 2002 and provided large-vocabulary continuous dictation to the enterprise. Aimed at solution developers and service providers with document-workflow solutions, the Voice Server for Transcription can automate what has traditionally been a very manual and resource-intensive process – that of dictation transcription.

For many years, physicians, lawyers, and other professionals whose professions require the production of high volumes of text, have relied on typists and transcriptionists to convert their dictation into documents. With the WebSphere Voice Server for Transcription, the professional's dictation can be transcribed automatically, leaving transcriptionists to only correct and edit (rather than transcribing from scratch), thus improving their overall productivity and turnaround. Since skilled transcriptionists are expensive and hard to find, automated transcription makes the process more efficient.

So what exactly is transcription? Take the example at the beginning of this article. If this audio were sent to the WebSphere Voice Server for Transcription, it would result in something like this:

*Subjective:* 32-year-old white female in for evaluation. Patient describes injury of left ankle.
*Objective:* Patient presents in mild distress and pain.
*Heart:* Regular rate and rhythm.

A transcriptionist would edit and correct the transcribed text, and the workflow application would use the edited text to fill in the appropriate fields (e.g., "Assessment" and "Plan").

## The Case for Speech

With the emphasis on improving customer service and customer-relationship management during the past decade, Interactive Voice Response (IVR) systems ("press 1 for savings, press 2 for checking…") have become ubiquitous. Most IVR systems were originally installed to automate customer service, or contact center, applications and reduce call-handling time. However, callers often bypass IVRs because of complicated menus, resulting in a higher-than-anticipated volume of calls ending up with a

Customer Service Representative (CSR). Voice-enabled applications replace complex menu choices by allowing callers to go directly to a selection with a simple spoken request. These systems are usually preferred by customers, encouraging them to participate in self-service voice dialogues rather than opting out to talk to a CSR. Voice recognition reduces the length of calls by 50% or more over menu-driven IVRs, increasing the use of voice automation (versus CSRs) from 20% to 60%. This results in a much quicker return on investment (ROI) of the voice system and increased customer satisfaction.

Let's take a look at the financial industry specifically. With voice recognition, financial institutions can reduce contact center costs, particularly on nonrevenue-generating calls. The typical fully-loaded cost of a CSR is roughly $45,000 per year (assuming a base salary of approximately $36,000), or over $1–$1.50 a call. While IVR systems drive this cost down, voice enablement reduces it further by flattening menus and speeding up navigation. With voice recognition, call costs can be reduced to around 30¢ a call or less, depending upon call volume.

Any increase in automated calls further drives down contact center costs. This kind of return on investment (ROI) means quick system payback. For example, some analysts claim that the payback on even "massive-scale, high-availability" voice-enabled contact center systems has been less than 18 months.

One large financial corporation is currently voice-enabling its automated basic banking services, including balance inquiry and funds transfer. Their cost justification for using speech recognition is based on three predictions. First, they feel they can shorten incoming call length by flattening touch-tone menus and allowing a caller to jump to a desired action. Second, they feel they can increase the percentage of automated calls on a yearly basis by 2%, which represents a large number of calls. Last, they feel they can capture roughly 20% of calls where the caller does nothing and simply defaults to a CSR. With a voice-enabled interface, they feel they can get callers to use the system rather than wait for a transfer.

There are many examples of enterprises adopting speech into applications in their organizations – and not just within the financial industry. While a business case can be built to show a quick return on investment, ROI is just one reason for justifying voice-enabled applications. Others include:

- Furthering a strategic or corporate goal
- Providing a service that previously could not be offered
- Enhancing the performance of an existing system

## What's Next

In the '80s, with the introduction of the PC, we increased the population of people who could access information. Then came the Internet, which used PCs as terminals to access the Web. Now we're moving to where we have small devices for an even larger population (including people who don't necessarily use a PC). Advances in voice technology have enabled people to speak directly to devices, rather than use traditional input methods such as the mouse or the keyboard. Soon people will be able to use speech when it's easier to say something than type it or wade through long menus, using a graphical interface when a visual representation serves your needs best, or using touch when that's the easiest way to make a selection. This is known as a multimodal interface. It combines all of the different ways to use technology, employing the most appropriate user interface to the task at hand.

For example, consider a busy mobile worker on the way to the airport who receives a call from a manager, wanting him in Hong Kong for a customer meeting – instead of Tokyo, as originally planned. Using a cell phone, the worker calls the voice-enabled, automated flight reservation number of an airline and requests a list of available flights to Hong Kong. Since the worker is using speech recognition, he gets immediate attention, rather than holding for the next available operator. Shortly after hanging up, a schedule of all available flights is displayed on his wireless PDA. The worker taps a selection, sending it back to the airline reservation server. The flight is booked. The worker used the interface (whether it was speech, graphics, or touch) that was the most convenient to use at the time, all within the same task.

In terms of the base speech technology, we'll continue to see improvements in recognition accuracy, including more languages and dialects supported and tools to make the creation of domain-specific language models easier. We'll see unstructured dictation (where you don't need to dictate punctuation and formatting) and we'll also start seeing speech recognition more fully integrated with other voice technologies, such as speech synthesis, language translation, speaker verification, and natural language understanding. Imagine getting the minutes from a conference call automatically transcribed for you, with individual speakers identified as they speak, and then automatically translating these minutes into other languages – all in real time while the conference call is going on.

Speech recognition technology is being used on the desktop by doctors, lawyers, and even students to input large quantities of text...It's being embedded in PDAs and smart phones to make mobile computing easier and more natural…It's being deployed in cars…It's being used by enterprises to enhance their self-service customer-facing applications…It's being used in real time by professionals to fill in forms, such as insurance claims and trouble reports…It's being used at kiosks in airports, shopping malls, movie theatres, and theme parks so customers can get real-time information…It's being used in the home to control appliances…the possibilities are endless. Speech is quickly becoming a key user interface of choice. And, given its history in the past decade, significant progress will continue to be made; by 2010, speech recognition will truly be pervasive.

## Conclusion

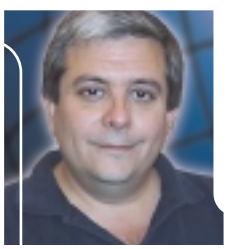
Today, faster chip speeds and more sophisticated algorithms mean voice recognition is performing better than ever before. New speech-enabled applications are hitting the market as businesses and consumers realize that voice is the most natural way to access information from the Internet, mobile phones, car dashboards, or handheld organizers. Voice technology may have started with desktop computers, but today, speech is making its way beyond the desktop world to the various touch points of an increasingly mobile e-business world.

*There's no substitute for experience*

# A Lesson From Grilled Ducks

BY JIM MARTIN

This month we're going to examine why customers make technology-buying decisions based on both the target technology and on their or someone else's ability to implement it. Bad software implemented by anyone is still bad software. Good software deployed by a knowledgeable team is what creates winning solutions. Good software implemented by less than the best is what can create dark scenarios that range from limited functionality to deep black holes that absorb revenue and resources into an unknown vortex.



ABOUT THE AUTHOR

Jim Martin has worked in the system integration and communications industry for the past 15 years. Working on design and implementation teams, he has been instrumental in deploying Web based mission-critical systems. Jim currently works as the director of sales at SKC.

E-MAIL
jim@skc.com

## Tale of The Ducks

Here's a quick story that demonstrates the point. One of my favorite foods is duck; my brother likes duck too, and we've both eaten duck in some great restaurants over the years. Duck is fun to eat, it's always served in an elegant fashion, and seems to be cooked just right with tender meat inside and crispy skin an the outside. Expert chefs take great pride in their ability to serve a finely crafted duck dish to their patrons.

I like to cook and fortunately I've only had a few kitchen disasters when I had to say "Hmm… this didn't work out at all." My brother likes to cook too. Last month we decided that we would cook some duck for ourselves. We were sure we could cook a duck as fine as any we'd eaten in restaurants. My new high-tech gas grill was out on the deck-ready to cook its first duck. The cookbook that came with the grill says that you can cook anything on it, burgers, steaks, fish, vegetables – even duck.

This was going to be great. I purchased fresh ducks from a poultry farm, planned the rest of the meal, and was all set. We followed the recipe; we were confident in our abilities and were guided by our collective cooking instincts. We knew that the cooking process would take hours, but we were prepared. In the beginning, everything went fine; we were rendering fat out of the ducks, they were starting to cook, and we were having a good time. My brother told me stories about ducks he had cooked in a conventional oven and how wonderful they were. About halfway through we made an adjustment to the heat and everything was still going fine. Day faded into night and we stood in the dark watching those ducks. This was taking longer than we had expected and we were still waiting to eat. All we really had was a bunch of rendered duck fat. It was getting late, wives and children were growing impatient; but the ducks weren't crispy. We had to do something.

We thought about what we could do to make the ducks crispy. We huddled together and discussed various options, then came up with a plan. We decided that the meat was cooked but the skin needed to be crispy. If we increased the heat to a very high level for a short time, the skin would crisp up and we would soon be savoring duck. We increased the heat and debated how high we should go and for how long. We decided to let it go to 500° for a minute and then pull the ducks out. Unknown to us was that around 500° is the flash point of duck fat. When we opened the grill, fire raged in and around our poor ducks. The tray beneath them was a seething inferno. I grabbed the tongs and attempted to rescue them from the flames. Flaming duck fat flowed out of the body cavity and onto my deck. Disaster had befallen our ducks. There would be no crispy skin this night, only the charred remains of what should have been delicious ducks. The ducks were still edible, but only as a shadow of what they could have been. Not very good at all but hey, it was still duck.

At this point you must be wondering why I've regaled you with this sad story of ducks and fire. The moral of the story is: even if you know what you're doing, when you step out of your range of expertise and experience, you inherently create some degree of risk. This rule holds true for almost every endeavor we undertake, including large-scale IT projects.

## Helping Customers Spread Their Wings

Many customers will spend countless hours planning IT rollouts only to have eleventh-hour issues crop up that threaten to scuttle everything they've worked so hard to achieve. The risk isn't increased because the deployment team isn't good or because the software is bad, but

rather because they are attempting to do something new. Sometimes when we attempt something new we're fortunate to taste success on the first try; sometimes we walk away with a burnt duck.

So how do we ensure that a project planned becomes a project successfully executed? There are several methods we can employ to ensure that we get our desired results; however, the method with the highest rate of success is to include the customer in the sales process. Last month we talked about how by including the customer as part of the sales team we can minimize closing objections because the customer has been actively involved in the value proposition and in determining how ROI will be recognized. Now we have a different reason to include our customer. We need to be sure that we're not selling a product that doesn't fully address customer needs – not because the product doesn't measure up, but because the customer may not possess the needed resources to address implementation issues that could arise or those resources aren't yet ready to fully maximize the potential of the project.

As we work with a customer to understand their business objectives, part of the goal is to understand what they envision the new technology initiative will bring them and how they intend to deploy and support that technology once it's in place. This can become a strange study of human nature. Some customers will divulge trade-secret information on their business plans but really don't want you to look too deeply into how they actually run their business. They don't want you to see what they see when they look in the mirror. It's easy to say, "By expanding our marketplace and making it easier to do business with us we will grow by XX% because we deployed this technology." It's hard to say, "This is great technology that we really need, but we may be at risk because we may not have the right resources to handle this."

I've seen sales stall at this point because either the customer or the sales team didn't find a path to create success out of limitation. Sooner or later someone has to add up all the costs related to implementing a project. In the end, how closely this number resembles the numbers used in the decision process is all that matters. Part of the ROI evaluation will include a calculation of not only the foundation technology but also the deployment and support costs.

Let's say you work for a software reseller; you sell a customer at a fair price and they're happy – great. The dark cloud appears when a troubled deployment adds additional dollars to the real cost of ownership. This can stall additional business by creating a financially gun-shy customer who thinks, "Great idea, but did you look at the numbers for what it cost to implement that stuff?"

This is when the participation of the technical members of the team becomes invaluable. In most cases it will be those team members who will have the greatest insight into what the customer's true technical resources are. Once you make an assessment that the customer doesn't have adequate resources, it becomes paramount that the sales team understand that things have evolved from a base software opportunity to include a services sale as well. If the customer has a relationship with an integrator or development company, you should try to meet with them to ascertain their abilities as well. Any faults on their part will reflect on the entire project, so make sure your customer has the technical resources they're going to need. As we noted earlier, good software implemented by less than the best can lead to dark scenarios.

This doesn't mean that the customer has to break with their trusted integration/development partners. With the proper skill sets they will continue to be valuable partners.

## Open Standards

How? WebSphere is built on open standards that allow application integration with tools that may already have been built. The open-standards approach is what makes WebSphere such an attractive choice for dynamic, mission-critical environments. For

example, when eBay decided to migrate to WebSphere, as the architecture for its auction site, the open-standards approach was key to the decision process. Chuck Geiger at eBay explained, "There were several factors that required us to take a systemic look at the entire business. We stopped, took a breather, and asked the question: If we were to do it today, how would we do it? The solution required a different approach using a three-tier architecture to expand the business horizontally and vertically using open standards, particularly J2EE."

Open standards make it possible for customers to not only reuse tools that are already in place, but also to adopt additional tools as their business evolves. Customers aren't locked into a proprietary technology that could limit or restrict their ability to chart the IT course of their choice. This is another reason that the integration/development partner is such a critical element to the success of a project.

IBM Global Services was on site at eBay for a skills transfer and to help with the J2EE architecture. "I want IBM to teach us to fish, not fish for us," Gieger explained. "eBay is still very much in control of its technological destiny." This is a winning approach that ultimately leads to a successful implementation. It expands the sales team to include not only the customer but also the service providers that will be contributing applications, processes, and support to the initiative.

The open-standards model is one of the most powerful selling points of the WebSphere product suite. It is critical that everyone involved in the deployment understand that the adoption of the WebSphere platform opens the door to untold ways of doing business with today's technology and with tomorrow's. The legacy technology that powers the customer's business today doesn't need to be cast aside. but rather it is incorporated into the new architecture. Open standards are the ultimate COTS (commerical off-the-shelf) solution. The residual value of IT assets is something that's often overlooked in the selling process; however, it's rarely overlooked in ROI.

## Learning From The Ducks

So how does this all fit together, and how does it relate to learning that sometimes it's better to order a duck in a restaurant instead of making it yourself? What does the duck have to do with WebSphere? The cooking of the ducks wasn't all that much different from planning an IT project. I knew my goal: to eat a delicious dinner. The customer knows their goal: to grow their business. I knew what I was going to make, just as a customer knows that they need to adopt a flexible, open platform. I thought I knew how I was going to accomplish the task, just as a customer often thinks they know how they're going to deploy a new technology like WebSphere.

In a nutshell, the lesson is to make the planning process part of the sales process. Take an accurate, holistic look at the customer. Understand the how, what, and why and make sure that the customer has the same fundamental understanding of these elements. Be alert; learn when to engage an IBM WebSphere Development Partner or IBM Global Services so the sales team can devise a complete solution for the customer, one that the customer is expecting because of their involvement in the sales and planning process.

Back to the duck for a second. If the man at the poultry farm had asked me, "How are you going to cook the ducks?" I would have told him I'd be using a gas grill. Perhaps he would have passed on a few tips on how to cook a duck on a gas grill. He took for granted that I knew how to cook a duck, just as we can take for granted a customer's resources to implement an IT initiative. Sure they have sophisticated IT departments that do a good job in the current comfort zones; given enough time and education they certainly could deploy the solution. However, we should always put the best, most complete, solution before the customer, and sometimes that involves proposing a service as well as an architectural solution. An experienced integration/development partner can provide the assurance of proven methodologies that have worked before. There's no substitute for experience. When eleventh-hour deployment issues crop up, and inevitably they do, that experience will keep the project moving forward, as opposed to stalling and increasing risk by altering the plan on the fly. Looking back, that late decision to turn the heat up on the ducks wasn't a good one. Why? The ducks burst into flames. An experienced chef would have known what to do.

# MESSAGE-DRIVEN BEANS IN WEBSPHERE 5.0

DEVELOP, ASSEMBLE, DEPLOY

BY PRASAD THAMMINENI

ABOUT THE AUTHOR

Prasad Thammineni is the founder of jPeople, a consulting services firm focusing on architecture, design, training, and mentoring services to customers focused on building enterprise Java applications. Prior to founding jPeople, Prasad was CTO of Specialized Software. Prasad holds a B.S in computer science and an M.S in mathematics from BITS, Pilani, India.

E-MAIL
prasad@jpeople.com

WITH THE RELEASE OF THE ENTERPRISE JAVABEANS (EJB) 2.0 SPECIFICATION A NEW CATEGORY OF ENTERPRISE BEANS WAS INTRODUCED – MESSAGE-DRIVEN BEANS (MDBs). MESSAGE-DRIVEN BEANS HAVE BEEN SPECIFICALLY DESIGNED TO PROCESS INCOMING JMS MESSAGES WITHIN AN EJB CONTAINER.

Although it's been more than a year since the release of the EJB 2.0 specification, previous versions of IBM WebSphere Application Server didn't provide support for it. The upcoming release of WebSphere Application Server 5.0 not only supports the EJB 2.0 specification but all the other specifications that make up the J2EE 1.3 specification. At the time of writing of this article, only a technology preview of WebSphere 5.0 was publicly available. You can use the preview to check out the new features that will be available in the final release of the product. In this article we'll use it to demonstrate developing and deploying message-driven beans in WebSphere.

This is the first of a two-part series in which I will introduce message-driven beans and discuss how to develop, assemble, and deploy them in WebSphere 5.0. In this article, I'll introduce concepts surrounding Java Message Service (JMS) and MDBs and present what you need to know in order to develop your first message bean. Since MDBs are designed to handle incoming JMS messages, having a prior understanding of JMS would help in understanding MDBs better. In this article I will provide a brief introduction to JMS. For a more extensive treatment of this topic, refer to the resources section at the end of this article.

## JMS Overview

In this section I will provide an introduction to JMS and how it relates to message-driven beans. Even if you're familiar with JMS, I would recommend that you read this section.

JMS is a standard, vendor-neutral API that Java programs can use to create, send, receive, and process messages from an enterprise messaging system. Enterprise messaging systems (sometimes referred to as message-oriented middleware) provide the infrastructure necessary to exchange messages asynchronously between distributed software applications. JMS is analogous to JDBC. Just as JDBC is a vendor-neutral API used for accessing a variety of relational databases, JMS is a vendor-neutral API used for accessing enterprise messaging systems. JMS is supported by many commercial messaging products, including IBM MQSeries, and Progress' SonicMQ. If you develop your messaging applications to use the JMS API, they become portable across messaging products that support JMS.

### JMS CLIENTS AND PROVIDERS

Java applications that use JMS are called *JMS clients* and the messaging systems that handle routing and handling of messages are called *JMS providers.* A *JMS application* is composed of one or more JMS clients and usually one JMS provider

A JMS client that sends messages is called a *producer,* whereas a JMS client that receives messages is called a *consumer.* Although from the definition it looks like you would need two programs, a single JMS client can behave as both producer and consumer. When a client wants to send or receive a message it needs to specify a *JMS destination.* In the case of a producer the JMS destination is where the messages are sent, while in the case of a consumer it identifies the source from which messages are received.

EJBs of all types can use JMS to send messages to various destinations. Those messages are consumed by other JMS clients, including message-driven beans. An MDB is the only type of EJB that can consume messages; entity and session beans cannot consume messages. This is because entity and session beans respond to synchronous Java RMI calls from EJB clients and cannot be programmed to receive JMS messages. In order to overcome this limitation, message-driven beans were introduced in EJB 2.0.

### JMS IS ASYNCHRONOUS

One of the key differentiators of JMS messaging is that it is *asynchronous.* That is, a JMS client can send a message and continue with its processing without having to wait for a reply. In other words, the client doesn't block for a response. This allows you to build applications where various subsystems can be loosely coupled and don't have to all be running at the same time in order for them to communicate. These subsystems communicate through a messaging server. Because of this, if one subsystem fails, it doesn't impede the functioning of another. Asynchronous messaging combined with *guaranteed delivery* and a *store-and-forward mechanism* provide the high-availability capabilities required by critical business systems to continue functioning without any interruption.

### JMS MESSAGING MODELS

JMS provides two types of messaging models: *publish-and-subscribe* (pub/sub) and *point-to-point* (P2P). Pub/sub is intended for a one-to-many delivery of messages, while point-to-point is intended for one-to-one delivery of messages (see Figures 1 and 2).

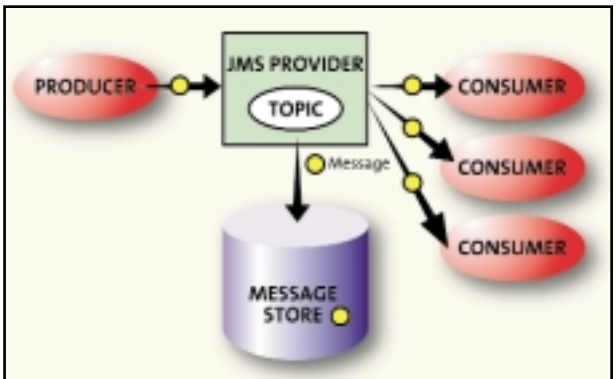



**FIG. 1: PUBLISH/SUBSCRIBE (1–> MANY) MESSAGING**

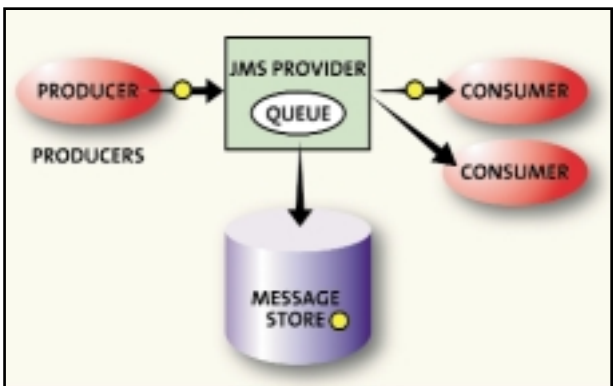



**FIG. 2: POINT-POINT (1–>1) MESSAGING**

In publish-subscribe messaging, one producer can send messages to multiple consumers. The destination to which a producer sends a message is called a *topic.* Consumers can express interest in receiving messages from a topic by subscribing to it. If one or more consumers subscribe to a topic, messages sent to the topic are delivered to each of the topic's consumers. Pub/sub is a push-based model, where consumers automatically receive messages without them having to poll the topic for new messages.

In point-to-point messaging, a producer can send a message to only one consumer. The destination to which a producer sends a message is called a *queue.* A queue can have multiple consumers, but only one consumer receives the message. P2P can be used as either a pull- or push-based model. Consumers either poll the queue or receive messages automatically when they arrive in the queue.

## Message-Driven Beans

Having seen some of the key JMS concepts in the previous section, we're now ready to talk about message-driven beans. In this section I'll introduce message-driven beans and what is required to develop one.

A *message-driven bean* is a new type of EJB component that can receive JMS messages. An MDB is a JMS client that can consume messages sent to either a queue or a topic by another JMS client (see Figure 3).

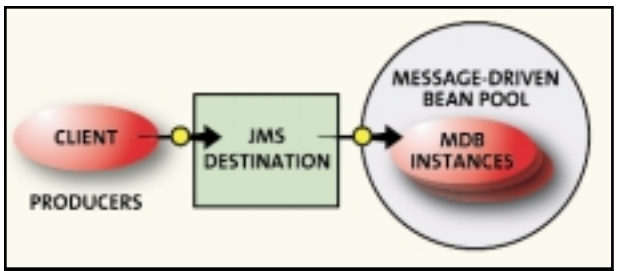



FIG. 3: A CLIENT SENDING/PUBLISHING A MESSAGE TO AN MDB

MDBs are stateless transaction–aware server-side components that run within an EJB container. While a message bean is responsible for processing messages, the EJB container is responsible for its environment, including transaction and resource management, concurrency control, security enforcement, and message acknowledgement.

Message-driven beans can consume messages concurrently. That means the container can instantiate multiple instances of a message bean to process messages in a JMS destination. An MDB can receive thousands of messages and process all the messages at the same time. The EJB container manages resources, transactions, and security in a multithreaded environment and leaves the job of processing the message to the MDB. On the other hand, traditional JMS clients have to custom-implement all these functions.

Message-driven beans, like entity and session beans, are enterprise beans. But message beans don't have component interfaces such as home, remote, or local interfaces. All they have is a bean class and an XML deployment descriptor. Since they aren't accessible using the Java RMI API and they respond only to asynchronous messages, they don't require component interfaces. The messages processed by a message bean can come from any messaging client, such as MQSeries or SonicMQ client.

Message-driven beans are stateless; this means they cannot hold any conversational state. In this sense they are similar to stateless session beans. Since the container could instantiate multiple instances of the message bean to process messages, they cannot hold any conversational state.

Having laid the groundwork, we're now ready to discuss what's involved in developing message-driven beans.

### MDB SEMANTICS

Message-driven beans at the minimum have to implement the javax.ejb.MessageDrivenBean and the javax.jms.MessageListener interfaces. In addition, each message-driven class should have one ejbCreate method that returns void and accepts no arguments.

Here's what the javax.jms.MessageDrivenBean interface looks like:

```
public interface MessageDrivenBean extends
javax.ejb.EnterpriseBean {

public void ejbRemove() throws EJBException;
public void setMessageDrivenContext(
MessageDrivenContext ctx)throws
EJBException;

}
```

The life cycle of a message bean is simple. The container creates an instance of a message bean in three steps. First, the container checks the message bean pool for an available instance. If it cannot find one, it creates a new instance of the message bean and adds it to the pool. Second, the container calls the setMessageDrivenContext method to pass the context object to the instance. Third, the container calls the instance's ejbCreate method.

The setMessageDrivenContext method is called by the bean's container to provide the bean with access to information about the container in which it executes. In the ejbCreate method you would code logic to initialize it. Since message beans are stateless and the ejbCreate method doesn't have any arguments, in most cases you wouldn't have any code in this method. The ejbRemove method is invoked when a message-driven bean is removed from the pool. You would code logic to clean up any resources you have acquired in ejbCreate.

Here's what the javax.jms.MessageListener interface looks like:

```
public interface MessageListener {

public void onMessage(Message message)

}
```

By implementing the MessageListener interface, the bean class is telling the container that it is a message-driven bean. Whenever a message arrives in the destination object associated with the bean, the container invokes the bean instance's onMessage method by passing the message as an argument. The onMessage method contains the business logic that handles the processing of the message.

## A Simple Example

In this section I'm going to illustrate message-driven beans using a simple example. In the example, a simple JMS client sends a text message to the MDB. The message bean, on receiving the text message, logs it to stdout.

### MESSAGE BEAN CLASS

In our example the com.jpeople.mdb.ejb.LogMessageBean is the message bean class. As I mentioned, the message bean doesn't have any component interfaces. Although this is a very simple example, it provides a good starting point for implementing your beans (see Listing 1). In each of the methods we display a message to illustrate the life cycle of the message bean:

- By implementing the javax.jms.MessageDrivenBean interface, our bean is now a message-driven bean.
- By implementing the javax.jms.MessageListener interface, our bean can consume messages.
- In the setMessageDrivenContext method, we save the context (passed as an argument in an instance variable of the bean class) in order for other methods to use it.
- Since this message bean is stateless, we don't have anything to initialize in the ejbCreate method. Similarly we don't have anything to clean up in the ejbRemove method.
- The onMessage method is where the business logic is coded. It receives the message as an argument and downcasts the message to an object of type

# WebSphere Infostructure

## THE COMPLETE MARKETING AND SALES ENABLEMENT PLATFORM

**WEBSPHERE**

*Products, Service & Solutions Provider*

**PRINT MEDIA**

**DIGITAL MEDIA**

**Synergistic Editorial Environment**

**Targeted Qualified Audience**

*25,000 qualified IT professionals & business executives*

**Value Added Benefits**

- *Online Banner Advertising*
- *e Newsletter Sponsorships*
- *Exclusive Targeted List Usage*
- *Events & Trade Shows*
- *Global press release Wire*
- *Extended Ad Usage in Digital markets*

**WebSphere Solutions Web**

**Company**

**Products**

**Services**

**Information**

- *Collateral*
- *White Papers*
- *Announcements*
- *Press Releases*
- *Presentations*

**Links to Sales Teams and Partners**

**Digital Ads**

**Marketing Tools**

- *Worldwide WebSphere User Group (websphere.org)*
- *SYS-CON.com*

**Target eMarkets**

- *Telecommunications*
- *Finance*
- *Technology*
- *Health Care*
- *Wireless Services*
- *Related User Groups*
- *Private Digital Value Chains*


SYS-CON MEDIA

Miles Silverman, VP

*miles@sys-con.com*

group intelligence

Peter Watts, CEO

*peter.watts@groupintelligence.com*

javax.jms.TextMessage. *TextMessage* is a particular type of JMS method that is used for encapsulating text messages. The message is retrieved from the TextMessage object and is written to the stdout. Notice that we don't specify the JMS destination anywhere in the class. The JMS destination is specified as part of the deployment descriptor and during deployment in WebSphere. The message bean's JMS destination type can be either a queue or topic.

### THE DEPLOYMENT DESCRIPTOR

MDBs have deployment descriptors like entity and session beans. The code snippet below shows only the portion of the deployment descriptor relevant to our simple message-driven bean.

```
<enterprise-beans>
     <message-driven>
       <display-name>DemoMessageBean</
        display-name>
       <ejb-name>DemoMessageBean</ejb-name>
       <ejb- class>com.jpeople.mdb.ejb.Log
        MessageBean</ejb-class>
       <transaction-type>Container</
        transaction-type>
       <message-driven-destination>
         <destination-type>javax.jms.Queue</
          destination-type>
       </message-driven-destination>
     </message-driven>
  </enterprise-beans>
```

For each message-driven bean you need to have a <message-driven> entry. The <message-driven> entry is declared in the <enterprise-beans> element alongside other entity and session beans. Just like session beans, it defines an <ejb-name>, <ejb-class> and <transaction-type>. It doesn't have any component interfaces. In addition, it also needs to define the <message-driven-destination> entry. The <message-driven-destination> element identifies whether the MDB is listening to a queue or topic. In our example, it is listening to a queue. You notice that nowhere in the deployment descriptor do we specify the name of the actual queue. It is specified when the MDB is deployed in WebSphere.

### THE JMS CLIENT

Having discussed the implementation of our simple message-driven bean, let's look at the bean client. An MDB client can be any messaging client. In our example, we have a JMS client that accepts a message as an argument and writes the message to a JMS destination queue. The MDB client source is shown in Listing 2. Since this article is about MDBs and not JMS, I won't discuss the client source in great detail.

In short, the bean client:

- Verifies that it has been invoked with an argument
- Obtains the QueueConnectionFactory and the Queue from the JNDI InitialContext
- Creates a QueueConnection and a QueueSession object
- Creates a QueueSender and a TextMessage using the QueueSession object,
- Sets it in the TextMessage object and writes it to the queue

## Deploying the Message Bean

Deploying an MDB involves creating an EJB JAR, packaging it in an enterprise application .ear file and installing it in WebSphere. In the second installment of this article, I'll discuss what is involved in packaging and deploying our simple message bean in WebSphere. I'll also discuss how to package the JMS client as a J2EE application client and how to execute it.

If you can't wait for the next installment of this series, download the already packaged enterprise application and client JAR and follow the included instructions on how to install and execute the message-driven bean. For location of the download refer to the Resources section.

## Summary

In this article we've learned about developing message-driven beans. We learned some key JMS concepts and how message beans compare to session and entity beans. Finally, we looked at what interfaces you need to implement in order to develop message beans.

## Resources


- Working sample of the simple message-driven bean: www.jpeople.com/articles.
- *WebSphere Technology for Developers:* www7b.software.ibm.com/wsdd/downloads/wstechnology_tech_preview.html.
- Monson-Haefel, R. et al. (2000). *Java Message Service.* O'Reilly & Associates.
- Monson-Haefel, R. (2000), *Enterprise JavaBeans, 3rd Edition*, O'Reilly & Associates.
- *Enterprise JavaBeans 2.0 Specification:* http://java.sun.com/products/ejb/docs.html.
- *Java Message Service Specification 1.0.2b:* http://java.sun.com/products/jms/docs.html.


LISTING 1: LOGMESSAGEBEAN CLASS

```
package com.jpeople.mdb.ejb;

import javax.ejb.MessageDrivenBean;
import javax.ejb.MessageDrivenContext;
import javax.jms.Message;
import javax.jms.MessageListener;
import javax.jms.TextMessage;

public class LogMessageBean implements
MessageDrivenBean, MessageListener {

   private MessageDrivenContext context;

   //

   // EJB Methods from MessageDrivenBean interface
   //
   public void ejbCreate() {
           System.out.println("ejbCreate called");
   }

   public void ejbRemove() {
           System.out.println("ejbRemove called");
   }

   public void setMessageDrivenContext(
    MessageDrivenContext ctxt) {
           System.out.println("setMessageDrivenContext called");
           this.context = ctxt;
   }
```

```
   //
   // Method from MessageListener interface – business method
   //
   public void onMessage(Message message) {
           System.out.println("onMessage called");
           TextMessage textmessage= (TextMessage) message;
           try {

                       // Display the text from the TextMessage
                       System.out.println("Message received:" +
                       textmessage.getText());
           } catch(Exception e) {
                          e.printStackTrace();
           }
   }
}
```

LISTING 2: MDB CLIENT

```
package com.jpeople.mdb.client;

import javax.jms.Queue;
import javax.jms.QueueConnection;
import javax.jms.QueueConnectionFactory;
import javax.jms.QueueSender;
import javax.jms.QueueSession;
import javax.jms.Session;
import javax.jms.TextMessage;
import javax.naming.Context;
import javax.naming.InitialContext;

public class MessageBeanClient {
   private static String CONNECTION_FACTORY =
           "java:comp/env/jms/theQueueConnectionFactory";
   private static String QUEUE_NAME =
           "java:comp/env/jms/theOutboundQueue";

   public static void main(String argv[]) {
           MessageBeanClient t = new MessageBeanClient();
           if (argv.length > 0) {
                       t.putMsg(argv[0]);
           } else {
                       System.out.println("Please specify a message");
           }
   }

   public void putMsg(String theMsg) {
           try {
                       Context ctx = new InitialContext();
                       JMSQueueTest(ctx, theMsg);
           } catch (Throwable t) {
                       t.printStackTrace();
           }
   }

   protected void JMSQueueTest(Context initCtx,
           String theMsg) throws Exception {

           // lookup the queue connection factory
           QueueConnectionFactory qcf = (QueueConnectionFactory)
             initCtx.lookup(CONNECTION_FACTORY);

           // lookup the queue
           Queue q = (Queue) initCtx.lookup(QUEUE_NAME);

           // create the JMS connection
           QueueConnection connection;
           connection = qcf.createQueueConnection();
           connection.start();

           // create the JMS session
           QueueSession session;
           boolean transacted = false;
           session = connection.createQueueSession(transacted,
            Session.AUTO_ACKNOWLEDGE);

           // create a sender, the text message & send message
           QueueSender queueSender = session.createSender(q);

           TextMessage outMessage = session.createTextMessage();
           outMessage.setText(theMsg);
           queueSender.send(outMessage);

           connection.close();
   }
}
```

EXCLUSIVE INTERVIEW

# Birth of a Platform

## Interview with Don Ferguson


PART TWO OF A THREE-PART SERIES

*In the last issue (**WSDJ**, Vol. 1, issue 2) Jack and Pat Martin, editors of **WebSphere Developer's Journal**, spoke with Don Ferguson about the beginnings of the WebSphere platform. This month, they look at Portal Server and what's happening with WebSphere today.*

### Five Benefits of WSFL

1. Allows developers to create a new service from an existing one
2. Allows ad hoc processes
3. Lowers development-cycle time
4. Moves application portability up a level
5. Interoperability; improves on SOAP by supporting semantic understanding

WSDJ: WHAT'S YOUR VIEW ON PORTAL SERVER?

**DF:** Portal Server is the most significant enhancement to the WebSphere family in a long time. It's a great product.

WSDJ: WHAT ARE THE MOST SIGNIFICANT POINTS ABOUT WEBSPHERE PORTAL SERVER?

**DF:** The first clever decision was to base it on the Portlet concept from the Apache Jetspeed project. The reason for that is if you are a portal and you cannot be transparently and independently extended by someone who wants to provide content, you're not going to be much of a portal. The Portlet model gave people an open way of adding elements to portal pages. Every other vendor's approach was proprietary. So the open-source Portlet concept actually built a community. People were contributing Portlets to the open-source portal server and WPS allowed a community to build up around Portlets. Portal servers have a tendency to be a "content drags product" sale. People will say "I need these 27 Portlets or things – if you support them, I'll buy your portal."

They really don't make a decision of which portal server is the best one. What are the things I need and which portal supports them? The openness of the Portlet model and WPS builds the content up. This is a big transformation of the way we do business in the Software Group. In the past we'd come in and say, "Look, we have the best product," and then we'd tell people to put their applications on the product. And now with the Portlet models, the nonvisual work we are doing in application integration and B2B and the CrossWorlds acquisition, we have the most complete set of front-end and back-end components running on our middleware products. Portal Server provides the customized, personalized workspace people need to integrate with a rich, complete set of applications. Portal complements our application integration and ISV partners, allowing us to provide all of the parts.

For example, we have great products to do application integration in WebSphere and MQ System Integrator. We have business-process templates that implement supply-chain management and other business processes

through CrossWorlds. We have Portlets that provide end-user interfaces to the processes and applications. All a customer or integrator needs to do is a little bit of customization – select an application adapter, maybe add a step in a business-process template, and they have a working solution. And by the way, it all runs on WebSphere message middleware products: MQ, MQ System Integrator, MQ Workflow, and DB2. We support other databases and messaging products, but the story is really powerful when they all come together.

The portal team has been hugely successful in building a community of Portlet developers. Larry Bowden, VP Portal Solutions Software Group, and Carol Jones, chief architect of Portal Server, can flip through pages of available Portlets.

### WSDJ: WHAT IS WEBSPHERE, WHAT IS YOUR SYNOPSIS?

**DF:** The standard answer after this is, "Gee, I was hoping you could tell me." WebSphere is a context-sensitive noun. Meaning, it is the server and all the other things that make it up and extend it.

When we say "WebSphere," we have a tendency to mix two terms. Sometimes with "WebSphere," we mean the Application Server and some of the Enterprise Extensions. And the other times we say "WebSphere," we mean the platform and all of the things that work together, like Commerce, Portal Server, Edge Server, and Business Integrator.

WebSphere is a family of products and a set of standards for doing application integration by supporting the development of new applications and integrating them with business logic and existing applications.

### WSDJ: INITIALLY, WHAT DID THE FIRST "WEBSPHERE" CUSTOMER LOOK LIKE?

**DF:** What this customer wanted to do was put an integration server in that supported their channels at contact time. The integration server had basic components that wrapped and encapsulated the existing applications and transformed them into a neutral model. If you were programming at this level all you needed to know was how to use a component, a service. You didn't need to know how to talk to CICS, how to talk to IMS or DB2. So it broke programming into two clear skills: people who know how to talk to CICS to produce components and people who know how to use services. We sometimes called this, "the lipstick on the pig layer." It didn't transform what you had, but made it modern and accessible to people through a common approach.

Other programmers did composite services that aggregated basic components. When you think about it, all the people downstream want to think in terms of portfolios and customers. They don't want to think in terms of customer in CICS and customer in IMS and customer in DB2.

We used to call these composites the IBM management objects because they didn't have a lot of implementation. If you asked them a question, "What's the checking balance?", they would just delegate it down to a basic component, so they didn't know how to do anything; they told everyone else what to do.

Then programmers built a process layer on top of the composites. The processes were CRM applications or account management, and used the composite services in their implementations.

With OO people at the time, there would be blood on the floor about what was a verb and what was a noun. The process layer implemented the verbs and the composites implemented the nouns.

The enterprise wanted to put channels in front of the processes and top-level composites. They wanted to do a plug-in to make them available for the Web and make another plug-in available for an analyst's database client. That was the basic problem.

Some of the things that got added as we moved beyond this type of problem was the concept of ISV applications. The big difference between Component Broker and EJB/J2EE is the packaging model. In J2EE, you could now package an application and deploy it. The story we had up until J2EE had been very focused on intraenterprise IT development and system integrators. J2EE enabled ISVs by giving a packaging model. The big insight into J2EE was how to package all of this together in a way that it could be deployed and then extended.

### WSDJ: DID YOU USE EJBS FOR THAT?

**DF:** Well, we use J2EE because in addition to deploying business logic you wanted to deploy servlets, JSPs, and data definitions. Then when you really think about it, part of what portal does is it puts a registration layer between the display layer and the business logic layer. This allows administrators to register a new Portlet for new business logic, and configuration can determine which


ABOUT THE INTERVIEWERS

Jack Martin, editor-in-chief of ***WebSphere Developer's 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.

Patti Martin is cofounder of Simplex Knowledge Company, where she is vice president of Creative Services. She manages the company's Web servers and oversees Web content and creation. Patti received her education at the New School in New York City and has taken continuing education classes at NYU and the School of Visual Arts.

E-MAIL
jack@sys-con.com
pat@skc.com


channels and users will see the application portal.

So what the Portlet concept adds is that you don't really need to extensively modify a configurations. The Portlet registers with the portal and becomes visible to users based on device types and roles.

And the next big insight was the Flow Composition insight. In the past, people actually wrote code for sequencing calls to existing objects and services; that is, building the process layer. But, since these composites all have well-defined interfaces, now with the Web services abstraction, you can actually build the processes visually through flow charts–type models. It becomes much more meaningful to people who aren't programmers.

### WSDJ: WHAT IS THE WEBSPHERE APPROACH TO WEB SERVICES?

**DF:** Historically, WebSphere applications focused on providing users with access to enterprise applications. People came in through Web browsers and used person-facing applications. The Web service support in WebSphere allows program-to-program interactions over the Internet and internal networks. This allows companies to directly link business processes, removing steps in which people "manually" use Web browser–based applications to exchange information.

### WSDJ: HOW IS WEBSPHERE INFLUENCING WIRELESS APPLICATION DEVELOPMENT?

**DF:** There are two main influences. First, we're producing small versions of WebSphere and J2EE that run on wireless devices. These support applications that use micro-browsers and forms to support interaction with local applications and data, as well as with Web services accessible over the network. Second, WebSphere has always had support for building server-side applications that supported access from wireless devices. We have functions in WebSphere Everyplace, Portal Server, Transcoding Publisher, and our application development tools for simplifying the task of developing an application that supports user access from multiple device types.

### WSDJ: WHERE DO YOU SEE WEBSPHERE AND WEB SERVICES GOING IN THE NEXT COUPLE OF YEARS?

**DF:** The next year is pretty well mapped out. It's continued leadership in standards and more of them. I think that as far as standards are concerned, you're only seeing the very beginning. There are lots of them teed up. There is Web Services Flow Language (WSFL). When you think about it, that's a very powerful concept. SOAP and WSDL are about, "Here's my service." But if you want to make new ser-vices from existing ones, you have to write code. WSFL is a modeling language. It's an excellent language for taking existing services and scripting them together to produce a new Web service.

### WSDJ: WHY IS THAT A PROPER CONCEPT?

**DF:** One, it allows you to build development tools that bring the flow-chart skills and business analyst folks into Web services development. It makes it more accessible.

The second one is that it allows ad hoc business integration. If some companies want to run a promotion for six months, WSFL lowers the bar for designing the common processes that support the promotion. Instead of setting up task forces and contracting with companies to develop custom business logic, everyone's going to look at the WSFL diagrams and say, "Yep, that's exactly how a car company, an airplane company, and a hotel company are going to run a promotion for Web site." It lowers the cycle time between concept and partnership. The partnership exists electronically. It's easy collaboration.


The final one, which is very powerful, is to use WSFL for "application portability." The definitions of the processes become portable between middleware and companies, moving the portability up from the details of J2EE and into a higher layer based on WSDL, Web services, and WSFL.

There are two things that have been key to the success of J2EE. One is application portability. The second is interoperability. With Web services, you're no longer worried about the details of the Java code, but what you are worried about is does everyone support Web services flow? Does everybody support Web choreography; do they support SOAP, do they support WSDL?

### WSDJ: WHAT ARE SOME OF THE DEFICIENCIES IN WSDL (WEB SERVICES DESCRIPTION LANGUAGE)?

**DF:** There is the concept of binding extensions to add additional information, but it isn't codified. So it's enabled but not standardized. For example, you'd want to say that if you invoke these operations you must use a digital signature–based signing and the certificate must come from one of these two certificate holders.

How do you express something like, "I support this operation but don't send it to me directly, send it through an intermediary?" So for services you need extensions to WSDL. WSDL defines the interface. We need some standard sets of extensions to document how transactions work, what is required for security, etc. This stuff is coming in 2002.

*Next month, Don looks at the future of WebSphere and plans for it down the road.*

THE LARGEST INTERNATIONAL

# WEB SERVICES CONFERENCE & EXPO IN THE WORLD!

web services EDGE conference&expo

**JUNE 24-27**
JACOB JAVITS CONVENTION CENTER
**NEW YORK, NY**

**OCTOBER 1-3**
SAN JOSE CONVENTION CENTER
**SAN JOSE, CA**

**WIN A $35,000 LUXURY CAR!**

ATTENDEES WILL BE INVITED TO TAKE A GOLF SWING TO WIN AND RIDE OFF IN A $35,000 LUXURY CAR!

## WEB SERVICES
SKILLS, STRATEGY, AND VISION

**REGISTER ONLINE TODAY**

**FOR LOWEST CONFERENCE RATES**
**EARLY SELL-OUT GUARANTEED!**

**VISIT WWW.SYS-CON.COM**

### Focus on Web Services

Web Services, the next generation technology that will enable the Internet to work for you and your business, and finally provide that ROI you have been after, will be put under a microscope at Web Services Edge East 2002.

Information-packed sessions, exhibits, and tutorials will examine Web Services from every angle and will provide cutting-edge solutions and a glimpse at current and future implementations. Hear from the innovators and thought leaders in Web Services. Enjoy a highly interactive CEO Keynote panel that will debate and discuss the realities and promise of Web Services.


### A Sampling of Web Services-Focused Sessions

- PRACTICAL EXPERIENCES WITH WEB SERVICES AND J2EE
- STATE OF THE WEB SERVICES INDUSTRY
- THE ODD COUPLE: MAKING .NET AND J2EE WORK TOGETHER
- EXPLORING THE .NET MY SERVICES INITIATIVE
- STANDARDS WATCH
- GUARDING THE CASTLE: SECURITY & WEB SERVICES

SEAN RHODY
CONFERENCE TECH CHAIR
WEB SERVICES TRACK CHAIR
EDITOR-IN-CHIEF
JAVA DEVELOPER'S JOURNAL

### Featuring...

- UNMATCHED KEYNOTES AND FACULTY
- THE LARGEST INDEPENDENT JAVA, WEB SERVICES, AND XML EXPOS
- AN UNPARALLELED OPPORTUNITY TO NETWORK WITH OVER 5,000 I-TECHNOLOGY PROFESSIONALS

### Who Should Attend...

- DEVELOPERS, PROGRAMMERS, ENGINEERS
- I-TECHNOLOGY PROFESSIONALS
- SENIOR BUSINESS MANAGEMENT
- SENIOR IT/IS MANAGEMENT/C LEVEL EXECUTIVES
- ANALYSTS, CONSULTANTS

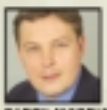
### Hear these thought leaders in interactive, cutting-edge keynote addresses and panels...

JEAN-FRANCOIS ARNAUDIC
SENIOR VP R&D, ILOG
FORMER CHAIRMAN, W3C

DAVE CHAPPELL
VP CHIEF TECHNOLOGY EVANGELIST
SONIC SOFTWARE

GREGG KIESSLING
CEO
SITRAKA

ANNE THOMAS MANES
CTO
SYSTINET



BARRY MORRIS
CEO
IONA

ERIC RUDDER
SENIOR VP DEVELOPER AND PLATFORM EVANGELISM
MICROSOFT

PATRICIA SEYBOLD
FOUNDER & CEO
SEYBOLD

**For Exhibit Information**

CONTACT: MICHAEL PESICK
135 CHESTNUT RIDGE RD.
MONTVALE, NJ 07645
201 802-3057
MICHAEL@SYS-CON.COM

**SPONSORED BY:**

IONA | END 2 ANYWHERE™
ADDS
bea
TogetherSoft
SilverStream
Actional
MERANT
Altaworks
PolarLake
ALTOVA

**MEDIA SPONSORS**

Federal Computer Week
Webservices.org
XML TIMES.com
Java Skyline
CF Advisor
WebServicesMall
WIRE
XMLBIZ
jCert
wrox
BASIS
SD Times
JAVA
wireless
XML
WebLogic
WebServices
WebSphere
ColdFusion

**OWNED AND PRODUCED BY**

SYS-CON MEDIA
SYS-CON EVENTS

*Autogenerating CMP entity beans from a relational database schema*

# Object/Relational Mapping

BY JAY JOHNSON

It's one of the greatest challenges in enterprise application development: object/relational mapping. Business information lives in relational databases, and applications are made up of objects. There is no shortage of products that attempt a systematic mapping between tables and objects, all with limited success.

EJBs and WebSphere add some new twists to O/R mapping, and WebSphere Studio Application Developer provides some new solutions, which we will explore here.

## Persistence of Vision

If you work on an EJB project, you'll run into this situation very soon, if you haven't already: a relational database exists, and you must access it using EJBs. One reaction to this task is panic, frantically writing SQL statements to be invoked from a session bean, or even worse, directly from the client. If this approach works at all, it will produce an application that cannot be expanded or maintained as database schemas and processing requirements change (almost constantly). There are better strategies to use, such as employing DAOs (Data Access Objects) to access the database tables, and shuttling data back and forth via POJOs (Plain Old Java Objects).

Whether container-managed persistence (CMP), bean-managed persistence (BMP), or another mapping strategy is used, the key to successful maintenance is autogenerating the beans when the underlying database schema changes. In BMP, most of the complexity is in the code and the deployment descriptor file is relatively simple. For CMP, the complexity is in the deployment descriptors, which can become a nightmare as they are edited to match changes in the database schema. This means we need to be able to autogenerate deployment descriptors as well as code.

CMP provides many advantages, especially on WebSphere Application Server. For example, the container can optimize queries as read-only so only the load method will be executed on the bean instead of both the load and the store, as would be the case with BMP. This can lead to a substantial performance increase. Perhaps the biggest advantage to CMP is that you can use a standard, built-in object-oriented query language (EJB QL) with CMP, but that's a topic for another article.

## Against the Grain?

When the EJB standard was new, J2EE pundits tended to recommend a "large-grained" approach. This meant that if a system consisted of many persistent JavaBeans, only a few would be EJBs. Usually stateless session beans would suffice to make JDBC calls. If entity beans were used at all, they would be responsible (via BMP) for persisting the simpler JavaBeans. This could become a maintenance nightmare. In a "fine-grained" approach, all relational tables are mapped to entity beans. This is more flexible and can be much more easily maintained. It can also have better performance and scalability. Note, however, that the ability to autogenerate code and deployment descriptor files is the key to success with fine-grained EJB assets.

In my opinion, all persistence and most transactions in a J2EE project should be handled via entity beans. There is considerable debate in the EJB/J2EE community about where entity beans belong, but there can be no doubt that one of the main purposes of entity beans is O/R mapping. Ideally, entity beans will make the persistence layer transparent to the rest of an application. Bean-managed persistence can be a reasonable solution, provided the BMP beans can be generated and regenerated automatically when the database changes. Making changes to BMP entities by hand can be a long, tedious, and error-prone process. Fortunately, there are commercially available tools to automate BMP creation, such as Borland's JBuilder Enterprise Edition. Of course, when it comes to creating and updating relationships between BMP entity beans, developers are largely on their own, often using data-access objects to perform joins between EJBs.

In a J2EE server that supports the EJB 2.0 specification, such as WebSphere 5.0 (coming soon), container-managed persistence for EJBs and the relationships between them is standardized and fully supported. If the database you're using doesn't require special kinds of access for the sake of maintaining isolation levels (locking) or some other feature, EJB

ABOUT THE AUTHOR
Jay Johnson is a J2EE architect at Qorusoft.

E-MAIL
aguy4ejbs@yahoo.com

2.0 CMP is a good option. Generating and updating the deployment descriptors needed for 2.0 CMP can be a challenge (especially if done in a text editor), but for WebSphere 4.0 developers, CMP can involve additional complexity. Since CMP for relationships between entity beans is not in the EJB 1.1 specification, WebSphere implements this feature via methods inserted into each entity bean in the relationship, along with relatively complex deployment descriptors.

## Up and Down with WebSphere Studio Application Developer

EJB developers almost invariably find themselves using two basic approaches to build an application: bottom-up and top-down. Most projects use a combination of both, but beginning from the bottom is the most useful since most enterprise projects involve connecting to at least one existing database.

The bottom-up approach starts with an existing database schema. Developers then use a tool such as WebSphere Studio Application Developer (WSAD) to generate EJBs and relationships from the tables in the schema. Each table must declare a primary key (compound keys are usually okay). By default, relationships are generated where foreign keys exist. The entire system can then be built on an entity-bean foundation that can be generated and regenerated automatically as the schema changes. Depending on where you are in the development cycle, the database schema may or may not migrate frequently. If the schema is volatile, or the project is entering into a maintenance mode, this is an ideal approach.

If the object model already exists, but the database doesn't (brand new project), it's also possible to build the system from the top down, using a tool such as WSAD to generate the database schema from the object model and map it back to the entity beans in the model. WSAD generates a set of tables to support the CMP entities inside the EJB project. In these tables, each column corresponds to a CMP field of the enterprise bean, and the generated mapping maps the field to the column. Relationship associations are mapped to foreign-key relationships. By default, EJB inheritance hierarchies are mapped to a single table; that is, the base and all derived enterprise beans are mapped to the same database table. Additional options exist that support generating joined tables for the enterprise beans.

WSAD 4.0 is based on the open-source Eclipse IDE, and as such, sup-

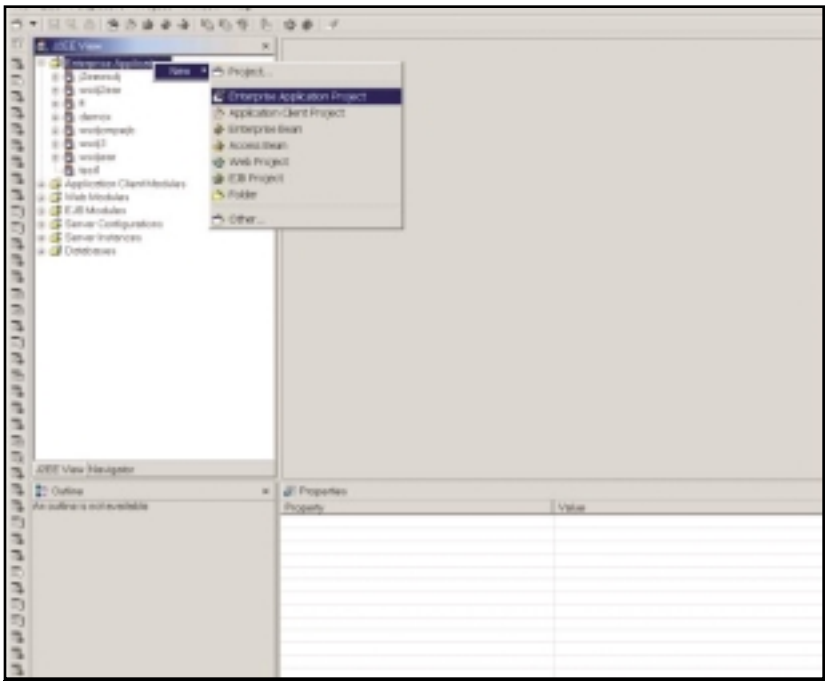


FIG. 1: THE J2EE VIEW

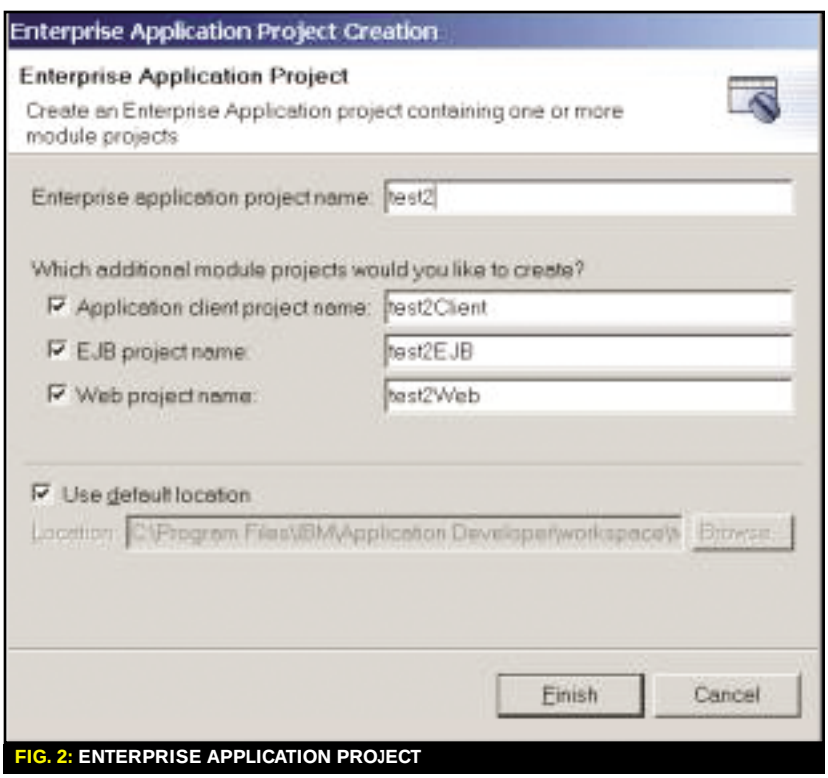



FIG. 2: ENTERPRISE APPLICATION PROJECT

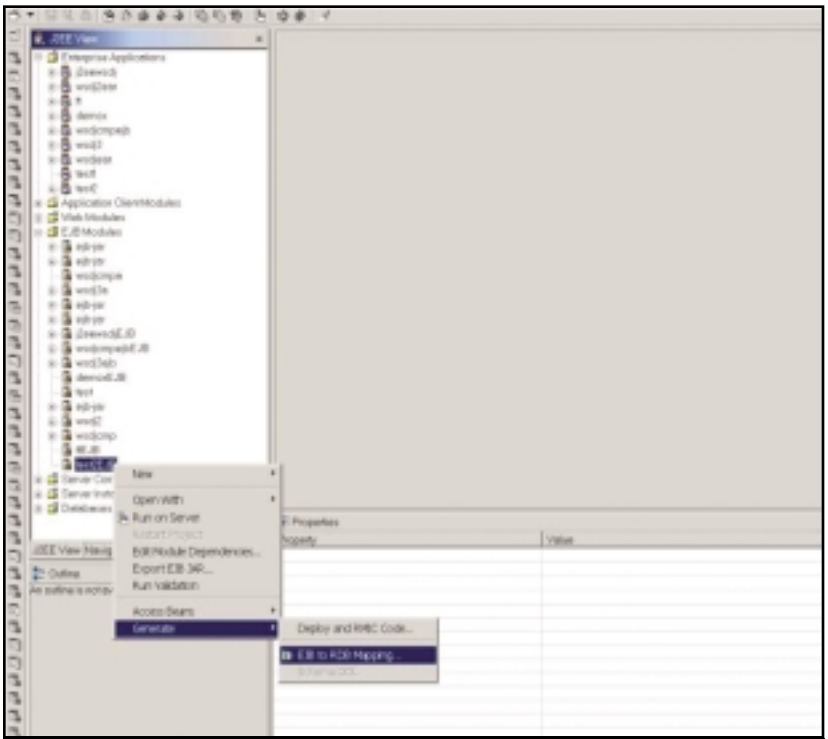



FIG. 3: RDB MAPPING

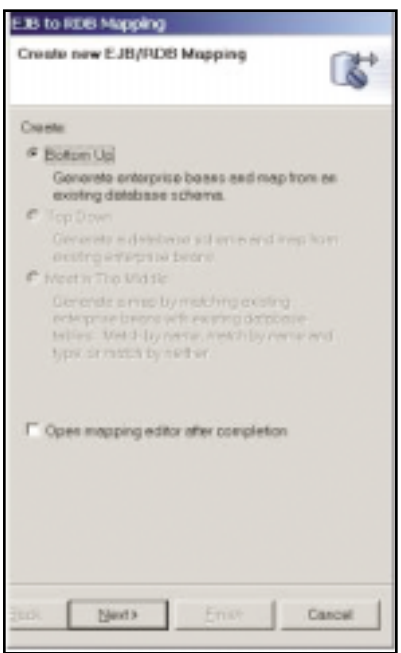



FIG. 4: BOTTOM-UP DIALOG

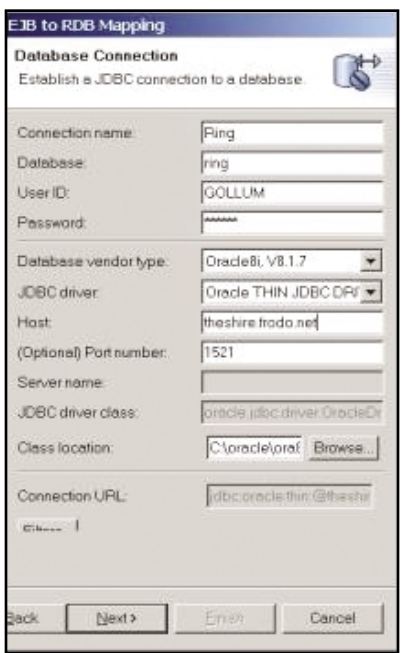



FIG. 5: DATABASE CONNECTION

ports a number of different perspectives for a project, and each perspective supports multiple views. Some of the perspectives available are *Data* (databases), *Java*, and *J2EE*. We can develop EJBs best in the J2EE perspective, and our first task is to create a new enterprise application project.

From the perspective menu on the menu bar, select the J2EE perspective, then left-click on Enterprise Applications in the J2EE view (see Figure 1). Now select New> Enterprise Application Project. You must supply a name for the project. If you click finish, then the default application client, EJB, and Web projects will automatically be created for this enterprise application project (see Figure 2). If you don't want these created, you can change the project names or deselect them here and create them later as needed. Click finish and you have a new enterprise application project. We will use this project to autogenerate entity beans from a schema.

To generate entity beans from a schema via bottom-up mapping, follow these steps:

1. In the J2EE perspective, select the EJB project from which you want to generate the bottom-up mapping. This must be a newly created, empty project with no EJBs in it.
2. Right-click on EJB project, then select Generate > EJB to RDB Mapping (see Figure 3).
3. In the Create new EJB/RDB Mapping dialog box, select Bottom up (see Figure 4) and then click Next.
4. At this point, you will immediately be confronted with a form you must fill out to connect to the database from which you want to generate EJBs (see Figure 5). To connect to an existing database schema, you'll need to provide the following information:
   - ***Connection name:*** This can be any name.
   - ***Database:*** The name of the database that contains the schema, such as bilbobaggins.
   - ***User ID, password:*** The user name and password needed to gain access to the database.
   - ***Database vendor type:*** The vendor type, such as Oracle8*i* 8.1.7.
   - ***JDBC driver:*** The driver you want to use, such as Oracle Thin JDBC Driver.
   - ***Host:*** The server where the database resides, such as theshire.frodo.net.
   - ***Port number:*** The port used to access the database server, such as 1521.
   - ***Class location:*** The ZIP or JAR file that contains the JDBC drivers for your database, such as c:\Oracle\classes12.zip.
5. Click Next. This step may take a minute to complete if the database is relatively large.

THE LARGEST INDEPENDENT

# JAVA DEVELOPER CONFERENCE IN THE WORLD!

JDJ EDGE conference&expo

**JUNE 24-27**
JACOB JAVITS CONVENTION CENTER
NEW YORK, NY

**OCTOBER 1-3**
SAN JOSE CONVENTION CENTER
SAN JOSE, CA

**WIN A $35,000 LUXURY CAR!**

ATTENDEES WILL BE INVITED TO TAKE A GOLF SWING TO WIN AND RIDE OFF IN A $35,000 LUXURY CAR!

## JAVA IN JUNE
ESPECIALLY IN NEW YORK

**REGISTER ONLINE TODAY**
FOR LOWEST CONFERENCE RATES
EARLY SELL-OUT GUARANTEED!

**VISIT WWW.SYS-CON.COM**

## Focus on Java

Java, now mainstream, is the dominant back-end technology upon which next-generation technologies are evolving.

Hear from the leading minds in Java how this essential technology offers end-to-end solutions to i-technology professionals and senior IT/IS strategy decision makers.

The Java Fast Paths on June 24, a Java-focused CEO Keynote Panel, and comprehensive conference sessions will focus you on Java-only information-packed day!


## A Sampling of Java-Focused Sessions

- JAVA 1.4: WHAT'S NEW?
- BUILDING TRULY PORTABLE J2EE APPLICATIONS
- JAVA TOOLS FOR EXTREME PROGRAMMING
- BUILDING ASYNCHRONOUS APPLICATIONS USING JAVA MESSAGING
- .NET VS. J2EE
- J2EE: SETTING UP THE DEVELOPMENT ENVIRONMENT
- BUILDING WEB SERVICES WITH J2EE
- DETECTING, DIAGNOSING, AND OVERCOMING THE FIVE MOST COMMON J2EE APPLICATION PERFORMANCE OBSTACLES

ALAN WILLIAMSON
JAVA CHAIR • EDITOR-IN-CHIEF AND DEVELOPER'S ADVISOR

## Featuring...

- UNMATCHED KEYNOTES AND FACULTY
- THE LARGEST INDEPENDENT JAVA, WEB SERVICES, AND XML EXPOS
- AN UNPARALLELED OPPORTUNITY TO NETWORK WITH OVER 5,000 i-TECHNOLOGY PROFESSIONALS

## Who Should Attend...

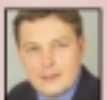
- DEVELOPERS, PROGRAMMERS, ENGINEERS
- i-TECHNOLOGY PROFESSIONALS
- SENIOR BUSINESS MANAGEMENT
- SENIOR IT/IS MANAGEMENT/C LEVEL EXECUTIVES
- ANALYSTS, CONSULTANTS

## Hear these thought leaders in interactive, cutting-edge keynote addresses and panels...

TYLER JEWELL
PRINCIPAL TECH. EVANGELIST • BEA

GREGG KIESSLING
CEO
SITRAKA

DAVID LITWACK
CEO
SILVERSTREAM



HARRY MORRIS
CEO
IONA

GREGG O'CONNOR
PRESIDENT
SONIC SOFTWARE

RICK ROSS
FOUNDER
JAXALOBBY

JAMES DUNCAN DAVIDSON
FATHER OF ANT

**For Exhibit Information**
CONTACT: MICHAEL PESICK
135 CHESTNUT RIDGE RD.
MONTVALE, NJ 07645
201 802-3067
MICHAEL@SYS-CON.COM

**SPONSORED BY:**

IONA | END 2 ANYWHERE™
ADDS
bea
TogetherSoft
SilverStream
Actional
MERANT
Altaworks
PolarLake
ALTOVA

**MEDIA SPONSORS**

Federal Computer Week
Webservices.org
XML TIMES.com
Java Skyline
CF Advisor
WebServicesMall
XML.ORG
WIRE
jCert
wrox
BASIS
SD Times
JAVA
wireless
XML
WebLogic
WebServices
WebSphere
ColdFusion

**OWNED AND PRODUCED BY**

SYS-CON MEDIA
SYS-CON EVENTS

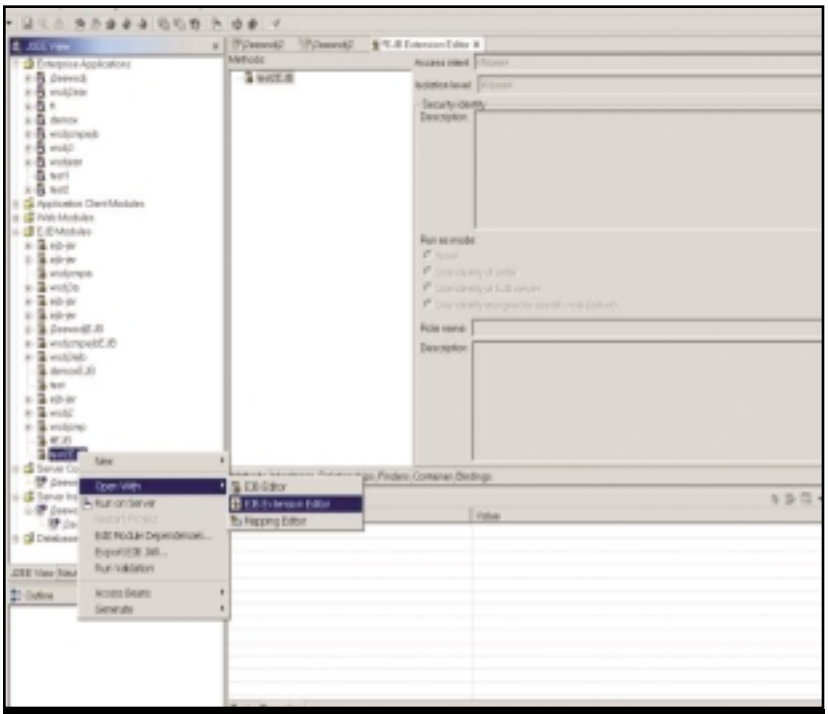


FIG. 6: EJB EXTENSION EDITOR

6. Select the tables from the database you want to import. You're probably only interested in one schema. You can select one table, all tables, or a combination of tables, and WSAD will generate EJBs accordingly.
7. The following rules apply to database tables: be careful that a single table or a subset of tables has no broken foreign-key relationships. Every table you want to convert to an EJB must specify a primary key. Columns must be convertible directly to Java types. Foreign keys must be specified as such in the table.
8. Define a package and prefix for the generated EJB classes, then click Finish.

To generate a database schema from an EJB project, simply right-click on the project and select Generate > Schema DDL. The resulting file will be stored as Table.ddl in the META-INF directory for the project.

When you create a new application client, EJB, or Web project you must specify the name of the enterprise application project in which it will be included. If you export an enterprise application project from WSADS, a corresponding .ear file will be generated, filled with .jar files for each of the EJB and Web projects belonging to the enterprise application project. This file is directly deployable to an instance of WAS 4.0.

## Deployment Descriptors

Whether you deploy to a test server within WSAD, or to an external server, you will need a flawless set of deployment descriptor files. These include the following:

APPLICATION PROJECT
- application.xml defines the components in the .ear file.
- ibm-application-ext.xml
- .modulemaps

EJB PROJECT
- ejb-jar.xml
- ibm-ejb-jar-ext.xml
- ibm-ejb-jar-bnd.xml
- Map.mapxmi
- Schema

The application.xml and the ejb-jar.xml files are standard, as described in the EJB 1.1 specification. The other files may need a little explanation. The EJB specification allows for vendor-specific descriptor files in addition to the standard J2EE deployment descriptors. They enable the use of vendor-specific EJB features such as the inheritance and association features IBM provides. The ibm-application-ext.xml and the ibm-ejb-jar-ext.xml files define the IBM-specific features used in this project. The .modulemaps file defines the contents of the application .ear file. The schema is a directory containing the definitions for each container-managed attribute generated, and the Map.mapxmi file provides the mapping between the schema and the EJBs.

Almost more valuable than generating the entity beans themselves, WSAD also automatically generates almost all of the deployment descriptor files necessary for your application to use CMP on WAS. It is far from trivial to create these correctly by hand, so this can save a lot of time, especially when it comes to maintenance.

Before our entities can be deployed, we must generate the stubs and ties for them. To do this, right-click the EJB project and select Generate > deploy and RMIC code. This will generate the stubs, ties, and other classes needed for the application to run in WebSphere. The generated code will automatically be verified at this point. To validate without generating code, lef-click on the EJB project and select Run Validation.

## Ties That Bind

The one file that isn't automatically created when you generate EJBs from the schema is ibm-ejb-jar-bnd.xml. This is the file that tells the server/container where the database (data source) resides. Instead, you must create it by right-clicking on the EJB project and selecting Open With > EJB Extension Editor (see Figure 6). Once the extension editor is visible, select the tab marked Bindings. Now click on the EJB project. If no information is visible, type in the JNDI name of the data source defined in the server instance you are using, along with the user name and password (see Figure 7). Finally, click on the file menu and select Save>EJB Extensions Editor.

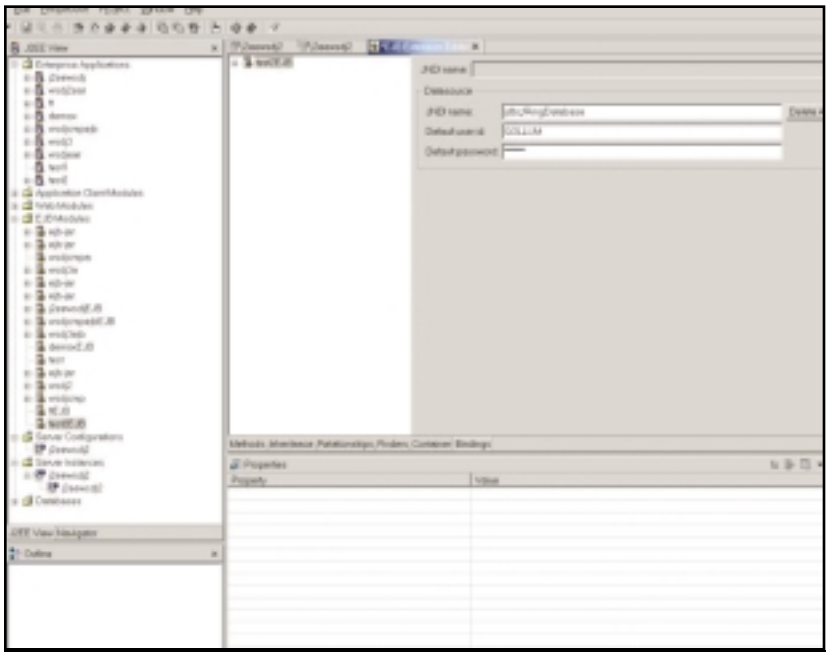


FIG. 7: JNDI NAME OF DATA SOURCE

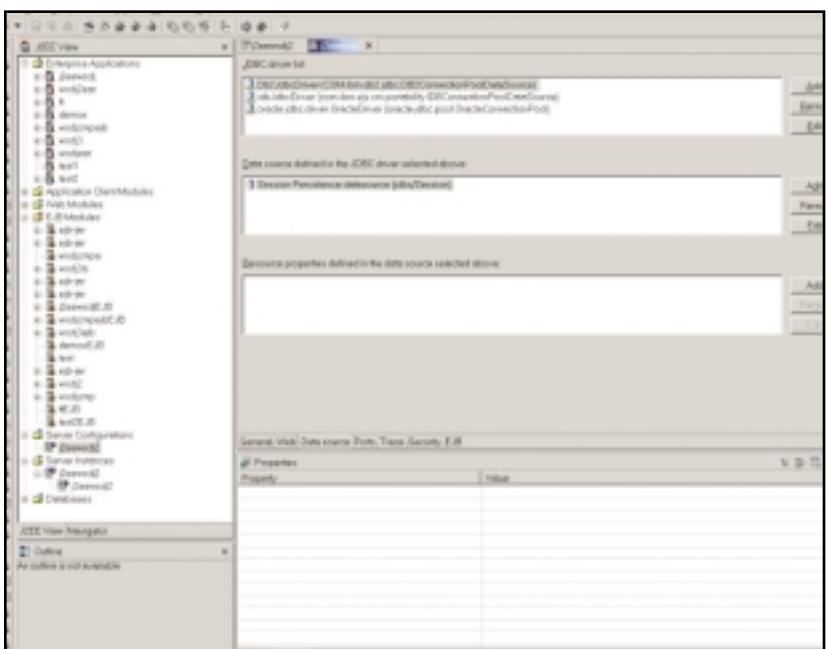


FIG. 8: ADD DATA SOURCE FOR EXISTING OR NEW DRIVERS

If no data source is defined in the server instance on which you will be installing this EJB project (as part of an application project), you will need to define it as follows: create a JNDI name for the data source. This can be any name you want, but every EJB project that uses the data source will need to include the JNDI name and login information. In WAS, data sources are tied to defined database drivers, so one driver can have many data sources. One way to specify the server data source is to edit the server-cfg.xml file as shown in the following:

```
<resourceProviders xmi:type=
  "resources:JDBCDriver"
  xmi:id="JDBCDriver_3"
name="oracle.jdbc.driver.Ora
  cleDriver" description=""
implementationClassName="ora
  cle.jdbc.pool.OracleCon-
  nectionPool" urlPrefix="">
<factories
  xmi:type="resources:Data-
  Source"xmi:id="Data-
  Source_2"
name="RingDataSource"
  jndiName="jdbc/RingData-
  Source"description=""cate
  gory=""
jtaEnabled="false"
  databaseName="ring"
  minimumPoolSize="1"
  maximumPoolSize="10"
  connectionTimeout="180"
  idleTimeout="1800"
orphanTimeout="1800"
  statementCacheSize="100"
  defaultUser="GOLLUM"
  defaultPassword="{xor}
  Liw7Oik="disableAutoCon-
  nectionClean-up="false">
  <propertySet xmi:id="null_ps"/>
  </factories>
</resourceProviders>
```

The other way to specify the data source from WSAD is to open the server configuration corresponding to the server instance you want to use, select the Data Sources tab, and add the data source under an existing driver or create a driver, then add the data source. Select the file menu, then select save when you are finished (see Figure 8).

An example of entity beans and deployment descriptors that might be generated from a database schema using WSAD 4.02 can be found at www.sys-con.com/websphere/source.cfm. This example also includes the database schema generated from the entity beans. Which came first? You decide!

## Conclusion

J2EE development tools and the EJB specification have improved to the point where it is practical and beneficial to use entity beans as fine-grained, persistent components. Automatic generation and re-generation of entity beans from a relational database, and autogeneration of database schemas from existing entity beans are the keys to success in this approach.

*The need to understand power, its distribution, and its effect on data center design*

# Electrical Supply Versus Demand

BY JOE FARSETTA

Power. Electrical power, that is. The very lifeblood of any functioning data center. It doesn't matter if it's the electricity that keeps your server at home running, or the electricity that keeps your employer in business, power is probably the single most important necessity in any IT process. You see, it doesn't matter how well you've developed the application, or how powerful your processing engines are; without electricity to run them, they may as well be boat anchors.

ABOUT THE AUTHOR

Joe Farsetta is an engineer with more than 20 years of industry experience in telecommunications, networking, operations, business process architecture, applications, and support. An entrepreneur and inventor, Joe's past engagements have included Unilever, NJ Transit, and a regional directorship at Bell Atlantic Network Integration, and several key positions at Exodus Communications. He currently provides independent consulting services and seminars specializing in data center design, high-availability infrastructures, operational readiness, and service-level agreements.

E-MAIL
jjf10965@yahoo.com

Close examination of how your data center is electrically fed is an absolute necessity. If you're planning on bringing a second data center online, an understanding of how power is produced, transmitted, and distributed is doubly important. Electrical fault tolerance is the end-goal, but without an understanding of how things work, you'll be shooting in the dark.

This article examines power from the supply side. It explains how commercial power distribution works, and will help you plan alternate commercial power sources to your data centers. With summer quickly approaching and the demand on commercial power suppliers ever increasing, the possibility of rolling blackouts is, once again, an unfortunate reality. We'll examine power distribution from the local, state, and regional levels, to try and make sense of it all. Hopefully, some of the information here will help you examine your own situation, or ask the right questions to ensure that decision makers and designers are indeed making informed decisions.

## The Mechanics of Commercial Power

### THE SOURCE

Electrical power starts at a generating station, or power plant. Generators within the plant spin and produce electrical current. The generator spins via several means: large water sources (like the hydroelectric stations at Hoover Dam, Grand Coolee Dam, and Niagara Falls), steam (produced via gigantic boilers fueled by coal, oil, gas, or nuclear sources), or gas turbines.

PHOTO COURTESY OF ORANGE & ROCKLAND

Regardless of the process, the end result is to spin the generator, which produces the electricity. Raw power from these commercial generators is typically in the form of three-phase alternating current.

### TRANSMISSION AND DISTRIBUTION

#### *The Transmission Grid*

Raw power leaves the generators and enters a transmission substation, the beginning of the transmission grid. The purpose of this grid is simple; it provides the means to transmit power over long distances. Long distances are the first challenge.

Long wires result in power losses. This is a definite problem, unless you plan to transmit power over an extremely short distance. Not too practical? Well then,you need to boost the raw power into a higher range, which allows it to be transmitted further. It's similar to putting a nozzle on the end of a garden hose. The hose takes a certain water volume and pressure from the main. Water hits the nozzle and the resultant pressure is increased due to a smaller opening. The end result is that the water shooting out of the hose now goes much farther than it did before. All the while, the volume and pressure available from the source remain constant. Although this is a simplified explanation (any electrical engineers reading this article, please forgive me), the principles are accurate.

Again, the solution is to boost the voltage to allow it to go further. It's at the transmission substation that the thousands of volts produced by the generators are stepped up to ultra-high voltages (like 155,000 to 750,000 volts), which allow them to effectively be transmitted over long distances via the transmission grid. Electricity in this range is referred to as "high tension." These voltages allow the power to be transmitted for 300 miles or further.

Towers that carry these ultra-high transmission voltages are typically made of steel or concrete. One thing's for sure: they're huge. Three high-tension wires can typically be seen

between towers, for the three phases produced by the generators. Ground wires also run between the towers.

### *The Power Distribution Grid*

As most data centers (and toaster ovens) can't use 155,000 volts of electricity, it must be stepped down at some point to a more usable level. Remember, however, that the ultra-high voltage lines won't be seen coming down your street. They'll be distributed to a series of electrical substations. These substations, and related lines leading to homes and industry, make up the Power Distribution Grid.

A simplified explanation of the workings of a substation goes something like this: ultra-high voltage (high tension) is transmitted to the substation. Actually, the transmission grid "drops" power to the substation, and continues to the next and so on. Once inside the substation, transformers step down the voltages to a more usable range. At that point, this lower voltage is sent to a distribution bus, which allows it to leave the substation in a variety of directions and possibly at different operating voltages. For instance, the distribution bus may contain transformers that step down the voltage even further, to 7,200 volts. The distribution bus may also couple the power to voltage regulators to help keep things consistent. This voltage typically makes its way to the individual step-down transformers, which supply power locally (on poles or lawn transformers) to homes and small offices. Power coming into your home is no longer the three-phase high-voltage monster that originated at the power plant.

Remember I told you that an electrical substation steps down voltages, and may also distribute power at different voltages? Well, substations are also connected to, feed, and are fed by, other substations. Higher voltages may also be fed to other locations. This is where the high-grade commercial power used in industrial complexes and office buildings also comes from. In some instances, the substation may actually be built for, or located in, the industrial complex itself. This will satisfy the need for two- and three-phase power at some locations. Higher voltages from these substations are distributed to power-hungry sites, such as large office complexes, malls, factories, and yes, data centers.

THE LARGER PICTURE

So, how does this help you? Well, now that you understand the hows and whys associated with power distribution, you can help ensure that your primary data centers are fed via dual feeds, or grids. All this means, really, is that you want at least two electrical sources for your data center. It would be nice if the power originated from different substations if at all possible. If not, you can shoot for separate feeds and separate step-down transformers.

But are redundant feeds from the same substation an adequate solution if you need redundant commercial power? Well, not really. I mean,7 think about the last electrical storm that knocked out power in half your town. Chances are that the substation was hit, a switch was tripped, or a transformer failed. Separate substations, if possible, are the better bet. This brings us to what you're really looking for: diversity.

### *Diversity as a Concept*

Diversity is a key element to power distribution. It's the foundation that provides the reliability you seek. Diversity along the local grid exists in several forms. Most notable are the diverse means by which distribution substations receive and distribute power. You see, substations have the means to power adjacent substations. That's right. If a substation loses its high-tension feed, it will continue to receive operating voltages from other adjacent substations. Although it no longer receives ultra-high voltages, the substation still receives higher voltages from surrounding distribution grids. These voltages are high enough to step-down and distribute. So, think of the distribution grid as being meshed at the substation level, because it is.

Diverse routing to your site is an absolute necessity. Just as the substations must be fed via diverse electrical paths, so must your data center. This is a challenge at the local level. Unless the complex where your data center is located is fed via diverse routing (different substations), it's doubtful that

diverse feeds to your data center are possible. So this is something to really think about when deciding on a primary or secondary data center location.

Certain buildings in New York City, for example, may actually be fed via four distinct high-voltage feeds, sourcing from diverse substations. Distribution of this type is commonplace in high-occupancy buildings or industrial facilities. Speak to your facilities electricians or representatives from your local utility company for details on how your site is fed. Tell them what you want to do, and work with them toward a viable solution.

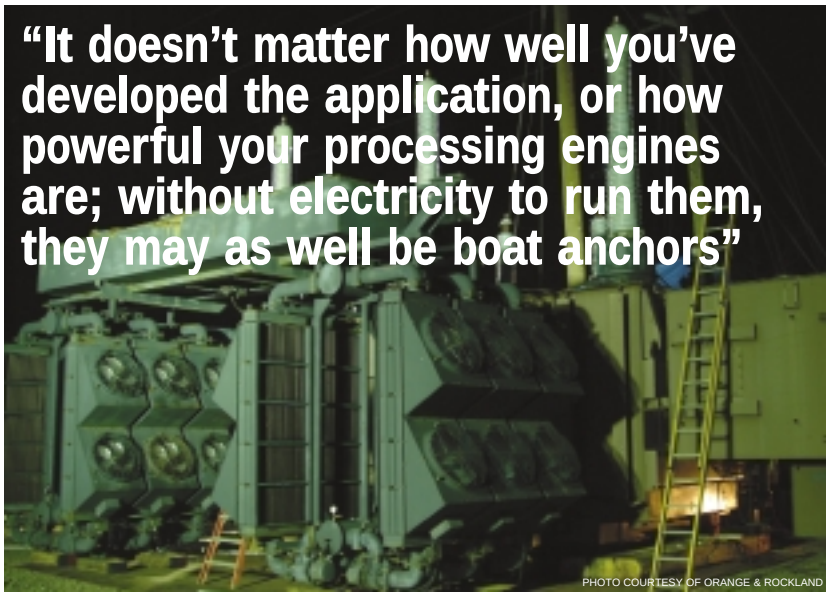
## Beyond the Primary Site: Secondary Data Centers

Let's say that your firm has decided to build a second data center. The first question to ask is how close the second site will be to the primary data center? If it's in the same general vicinity (like three blocks away), you need to see if it's fed by the same substation. If it is, you'll want to consider an alternate site. Location of your secondary data center is really important from a power distribution perspective. Let's look at power production, transmission, and distribution from three distinct viewpoints: local, state, and regional.

### LOCAL POWER

We're talking very local here.... Like from a local power plant or large substation to your locations. Substations are everywhere, almost to the point of being so commonplace that you probably don't notice them. So there's a chance that your sites may indeed be fed via alternate power distribution grids. Source the substation to the next transmission point up the line. Is it fed from the transmission substation at the power plant? If so, this is a good thing. A single power plant feeds many towns. Just ensure that your sites are (or will be) fed from different substations. Chances are pretty good that they are.

But will it be good enough? That depends on your firm's individual goals for electrical fault-tolerance. Remember that most substations are meshed with the next closest substation. You made decide that you need to have the secondary site fed from a different power plant.




### STATEWIDE POWER AND DISTRIBUTION

No power plant is an island. They feed, and are fed by, other power-producing plants within your state. Just as there is a local power transmission and distribution grid, there is a state-wide grid as well. Within that grid is a network of interconnected power plants that help to ensure power demands are met for the entire state. If the production capacity of a local plant is deficient at peak times, the statewide power grid augments the available supply of power to the area. Building additional power-generating plants is the best way to ensure that supply meets or exceeds the demand. But the construction of new powerhouses hasn't always kept up with the total demand for power. So the notion of regional grids has reached the forefront.

### THE REGIONAL APPROACH

California recently found itself in a power crunch. New York may soon be heading in the same direction. In these instances, the demand for power is outstripping the production capacity of the statewide power grid. So, what can be done to keep up with demand? Building additional power plants is the correct solution, but who wants one of these facilities in their backyard? These are legitimate concerns, and the end result is the delay of progress in any planning, permits, or new construction. For these reasons, states are looking to their neighbors for help.

Prodded by the federal government, action has been taken toward the creation of regional power grids. Hypothetically, all northeastern states, for example, may join (or be forced to join) an "official" regionalized power grid. Although a good idea, it has its own set of problems – rate control; local and state environmental authority; and the long-term role of state-sponsored public utility authorities that typically control the electric companies.

None of these obstacles are insurmountable. In actuality, the "grid" extends far beyond statewide boundaries already. Metering, economics, and regulation can sometimes be more complex than the technology needed for the actual generation and distribution of power, but these issues can and will be worked out. Whether it's for the better remains to be seen. The bottom line is that certain states are heading for an energy crunch and something must be done.

## *Federal Dams and Power Marketing Administrators*

Believe it or not, the federal government is in the power business. Huge hydroelectric power plants at the Grand Coolee and Hoover Dams are examples of the megawatts that these facilities generate. Transmission lines from these facilities provide power to a number of locales. In the case of Grand Coolee, this includes portions of Northern California. In this scenario, generators are phased in and out based on the load. Even with this power augmenta-

tion, California is still in a crunch. For this reason, with power being brought in using "regionalized" methodologies, a best bet may still be to build in another geographical location.

This is why an understanding of the production and distribution of commercial power is so very important for the IT professional. Asking the right questions can save you a lot of pain in the future, especially if your data centers aren't backed up by generators or a UPS capable of sustaining operations for an extended period of time.

### BROWNOUTS

Worse than a blackout is the infamous brownout. Sometimes there simply isn't enough electricity to go around. When this happens, unless power consumption is quickly brought under control, insufficient levels are distributed along the local grid. Remember that garden hose analogy? Now imagine trying to feed 100 lawn sprinklers with that same hose. Doesn't work too well, does it?

The term *brownout* was coined in the 1960s, named for its effect on incandescent lighting (bulbs burned dimmer, or brown, instead of bright white). Now a lightbulb may not care what voltage you feed it, but sophisticated equipment does, especially computers.

Computer power supplies may come with over-voltage protection (within reason), but consistently feed it with insufficient power and watch out! Everything from logic and memory processes to how fast the cooling fans and disk drives spin is in peril. And, while we're talking about cooling functions, think about how that power-starved motor, turning the data center's air conditioning compressor, feels. Brownouts have long-reaching effects. It may not affect your data center today, but it will shorten mean-time between failures of components, void manufacturer warranties, decrease the availability of common repair parts in a region (from all the repairs on everyone else's equipment), and who knows what else!

To prevent the damage associated with brownouts, power companies control power demand through a rolling blackout. A blackout occurs when power is absent from a location – your house, neighborhood, town, and so on. When the power grid can no longer keep up with demand, emergency procedures must be enacted to immediately reduce system drain. The resulting action is to shut off power to "pockets" at the source. Since no single area can be affected indefinitely, each "takes turns" during the day. The end result is a "rolling" blackout. While these blackouts may help the power authority, they can be devastating to local businesses. Think of yourself as an ice cream shop owner without power to your refrigerators for five hours a day, or a sales rep who can't process an order!

### BUILD FURTHER AWAY

Regional power grids may indeed help the situation, but may not be the "belt and suspenders" you're looking for. If your company has the real estate in a different region, you may consider building your secondary data center there. You'll be assured that power is supplied via different local, state, and regional power grids. The data center will also be insulated from the possibility of city, state, or regional disaster, either natural or man-made.

Once power has reached the data centers, it's important to decide, and design, effective distribution internally (e.g., within the data center itself). This is where the notion of harmonics, power balancing, load distribution, UPS, and generators come in. If you want to help ensure power continuity to mission-critical equipment, getting the power to your site is the first step, but a huge one!

## Conclusion

Now that you understand how power is produced and distributed, you can see the importance of comprehensive planning in data center design and construction. Proper design can save your job, and may someday save your company (literally). If your firm depends on its computing environment, a rolling blackout or any extended power outage can be devastating to your business. Keep this in mind. It's going to be a long, hot summer…

# Maximizing The Benefits of Web Applications

BY JASMINE NOEL

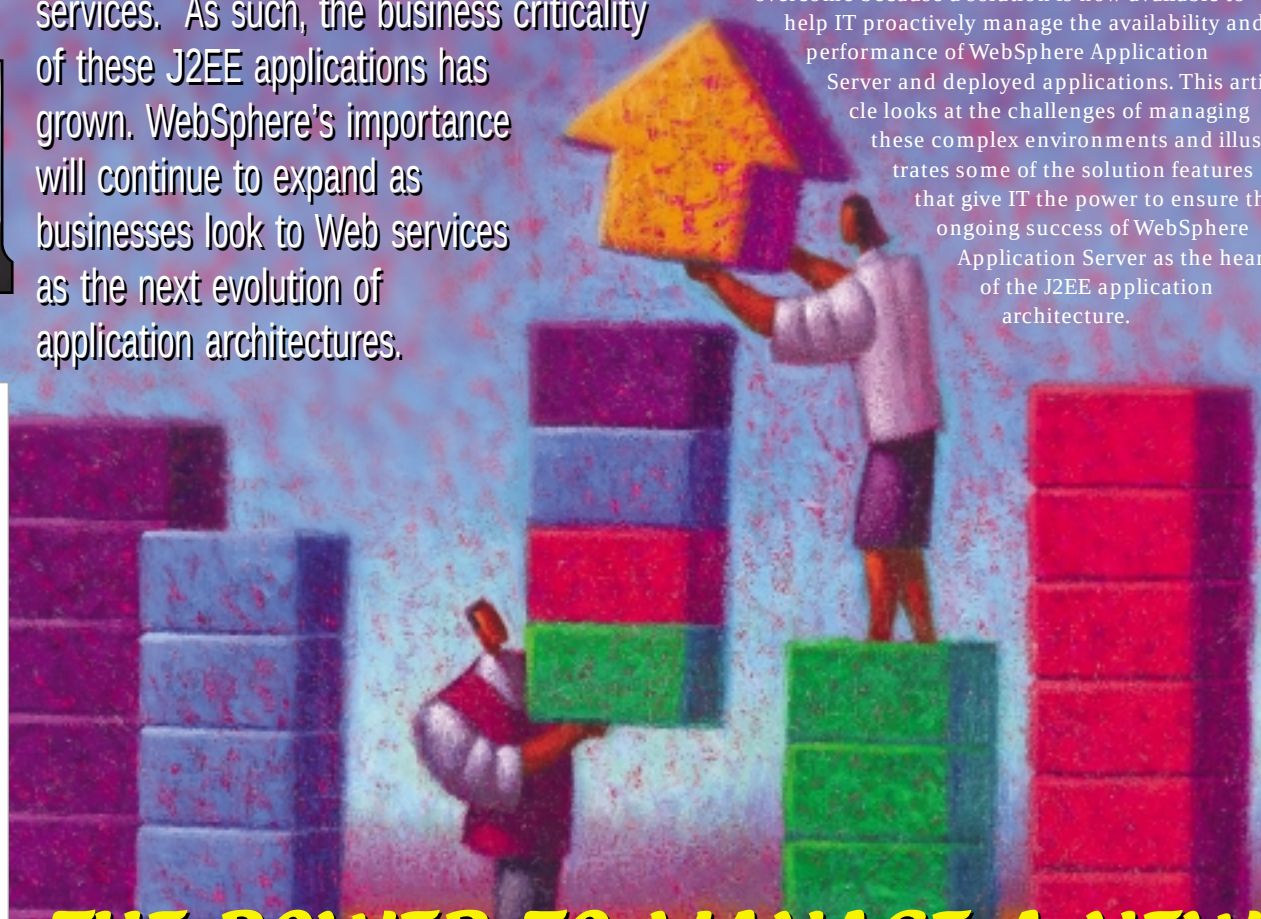
Enterprises rely on IBM's WebSphere Application Server to quickly build complex applications and highly scalable transaction services. As such, the business criticality of these J2EE applications has grown. WebSphere's importance will continue to expand as businesses look to Web services as the next evolution of application architectures.

ABOUT THE AUTHOR

Jasmine Noel, director of the Systems and Applications Management practice at Hurwitz Group, focuses on the financial impact of infrastructure performance issues and managing business services that are dependent on externally provided infrastructures. Jasmine has nearly 10 years of experience as both a technology analyst and consultant in the IT management arena.

E-MAIL
jnoel@hurwitz.com

Trusting your business to this future means that IT must guarantee a quality end-user experience by managing the application server and application components' performance. This challenge is daunting because proactive administration of complex and dynamic component interactions is required. It can be overcome because a solution is now available to help IT proactively manage the availability and performance of WebSphere Application Server and deployed applications. This article looks at the challenges of managing these complex environments and illustrates some of the solution features that give IT the power to ensure the ongoing success of WebSphere Application Server as the heart of the J2EE application architecture.



## THE POWER TO MANAGE A NEW REALITY IN THE ENTERPRISE

## Managing WebSphere-based Applications Is Business-Critical

WebSphere Application Server (WAS) has become the foundation that many enterprises rely on to connect to their customers, partners, and suppliers. It's used for applications as diverse as the number of companies using it; however, in most cases *businesses use these applications to drive their profitability*. For example, one tire manufacturer off-loads its 30,000 phone orders per day from dealers to a WebSphere-based solution to reduce costs; ensuring high performance is critical to achieving this goal. The flip side of these success stories is that it isn't possible to ensure high performance if the application's execution is invisible. This is the case for most IT organizations implementing WebSphere-based applications. The few embedded tools to monitor and control these critical applications have been slow to arrive and a problem has arisen: if IT cannot ensure high-performance applications, the business dependent upon them cannot thrive.

## Managing WebSphere-based Applications Is Challenging

The difficulties arise because WebSphere-based applications are both critical for business success and among the most difficult technologies to manage. There are three main reasons for this:

- Java application architectures are still rapidly evolving.
- Current management solutions often fall short.
- Custom EJB and servlet components are developed under strict time-to-market pressures and are not instrumented for management.

First, there is the rapid maturation of the technology from a servlet-based design to full-blown J2EE component architecture. The rise of interest in WAS lies in its support for other object technologies (CORBA and COM wrapping), transaction services that work with CICS, integration with MQSeries and adapters for SQL databases, ERP, SCM, and CRM packages. Developers can rapidly create EJBs and servlets to deliver new business services. Inherent in such component architectures are software dependencies that affect service performance. These dependencies aren't easily visible because traditional application management analytic processes do not apply, making it difficult to achieve the consistent application performance that leads to maximum business value.

Secondly, IBM has tools that provide visibility into the WAS infrastructure. WAS Resource Analyzer monitors the execution of EJBs, servlet resources, and Java Virtual Machine (JVM) resource usage. Log Analyzer allows administrators to sort WAS log entries based on severity, process ID, thread ID, and so on. Site Analyzer provides basic Web-site traffic measurements. The value of these tools is that they provide detailed information about the WAS infrastructure that administrators can use for performance analysis tasks; however, they don't:

- ***Automate and therefore speed the problem identification and resolution process:*** With these tools administrators still need to create a test case to reproduce the problem and then analyze execution and log information from every component to isolate the problem.
- ***Provide visibility into the overall infrastructure (e.g., back-end systems and databases) and the dependencies among the infrastructure components:*** Only by managing in this broader context can the application server's health and performance be correlated with the health and performance of other components and with end user–experience data so that IT can truly understand the magnitude of a problem and its root cause.

The inability to automate and provide end-to-end visibility results in longer downtimes or performance slowdowns that negatively impact business. Complicating matters is that many third-party solution providers focus on extending management functions to the WebSphere Application Server software itself. They don't concentrate on the *custom* EJB application components that encapsulate the business logic. As such they have difficulty providing visibility into the application components that deliver the actual business value. No visibility means no possibility of improving the performance of the most critical parts of the application.

Finally, companies choose to build custom applications rather than buy off-the-shelf software only when the application features are critical to delivering a competitive advantage. In other words, only the most important business functions are built internally and usually under extreme time-to-market pressures. Companies speed their development processes by leveraging the noncore functions supplied by WAS and focusing all of their creative efforts on the critical EJB components. This need for development speed, however, means that there is no time for rigorous quality assurance, performance testing, documentation creation, and technical support training. In many enterprises there is inadequate interaction between operations and development teams, which exacerbates the problem.

This compartmentalization of knowledge means that all too often no one truly understands how the custom components interact with the underlying WAS in production environments. The result is that operations inherit applications without code-level knowledge to troubleshoot runtime problems. Without a feedback loop, developers don't know when and which components fail to meet business performance requirements. Thus, optimizing components' execution for real-world conditions cannot be done effectively without tools that facilitate developer-operations communication in building and tuning high performance application components.

Enterprises that can handle these difficulties will avoid the revenue losses, user satisfaction issues, and spiraling cost of inefficient management associated with under-performing J2EE applications. Enterprise IT needs a solution that delivers component-level visibility into and control of IBM's software, the business-specific EJBs and servlet components built on the WebSphere platform. The solution must also correlate the application's performance within the broader context of the entire infrastructure delivering the service and its relation to the users' experience.

## Requirements for a Successful Management Solution

To address these problems IT requires a management solution that:

- Monitors end-user experience metrics against the business's goals
- Delivers detailed knowledge of the behaviors of specific technologies
- Captures the dependencies of the multiple technologies that deliver a business application
- Correlates the understanding gained through these three steps to rapidly pinpoint the root cause of problems
- Enables advanced manual or automated problem resolution actions

One example of a management solution that fits these requirements comes from Resonate. Resonate Commander Solutions relies on a service model to drive proactive management of the business services delivered to end users. This service model captures the business objectives and the rela-

tionships between the various technologies delivering those objectives. When Resonate Commander Solutions monitors the relevant technologies and end-user metrics it correlates the data with the service model to determine and correct the issue. For example, when diagnosing a performance problem with a catalog service, it uses the service model to automatically determine that the associated server operating system, relevant database, and network routers are operational, but the WAS application is causing problems. This service diagnosis eliminates the finger-pointing that traditionally lengthens the troubleshooting process and adds to downtime costs.

When selecting a management solution, consider a solution that continues its analysis by drilling down to identify the offending servlet and then notifies the application manager of recommended actions. Usually, management solutions accomplish this by providing an application-specific module that works within the broader solution to provide in-depth, service-based management capabilities to WAS administrators. This module would:

- Provide deep knowledge of WebSphere Application Server and custom component performance through its support of both IBM's proprietary APIs for WAS version 3.5 (e.g. Enterprise Performance Management and WebSphere Control Program) and their open PMI interface for WAS version 4.0. Through these, the module would access the same detailed information delivered by IBM's native monitoring tools (e.g., Resource Analyzer) to troubleshoot problems within the larger context of the actual WAS deployment.
- Apply the service model capabilities to capture the relationships between the various components that make up a WAS instance such as stateless EJBs, JMS sessions, JDBCs, etc. WAS administrators can, therefore, easily and quickly drill down to identify not only the specific offending EJB, for example, but also other things that may have been impacted, and perform actions on specific WAS instances or their dependent components.

The management solution, therefore, must combine both specific component details and the broader service context to identify problems and automate resolution of the specific WAS instances. This WAS-specific management data must be automatically rolled up into an end-to-end service view allowing administrators to effectively manage the WAS components in the broader context of end-user experience.

A more sophisticated management solution also gives IT real-time control over the production environment through a variety of corrective actions. IT managers should be able to specify the type and sequence of actions, such as restarting failed application processes or redirecting transactions to an alternate server until the problem is resolved. With these capabilities IT no longer simply reacts to service problems – it prevents them from occurring.

## Business Value of Management Solutions

Business performance now moves in lockstep with WebSphere application performance. IT, therefore, needs solutions able to both manage the user's quality of experience and troubleshoot specific technologies. A management solution that can aggregate data across an infrastructure stack (components, middleware, back-end systems), and across the IBM "stack" (WebSphere Application Server, MQ Series, and DB2) into a single service model, are invaluable because IT managers can rapidly pinpoint the infrastructure components negatively impacting the quality of experience. A solution that can then drill down to provide granular visibility into component performance enables WebSphere administrators to rapidly and accurately troubleshoot technology-specific root causes. Taken together, these abilities minimize the financial impact of technology problems on the business. This translates into lowered costs, revenue saved, faster business transactions, and improved customer satisfaction.

The ability to automate problem resolution for common problems gives IT proactive control over the capacity and performance of the underlying infrastructure. This improves the efficiency of IT staff, allowing reallocation of limited resources to strategic efforts. The automation also allows business managers to maximize use of available resources providing tangible control over the quality of service experienced by external users.

A more sophisticated management solution would also facilitate the "holy grail" of application development – code reuse. Component architectures have long promised reuse as a business benefit, which few enterprises have realized because the impact of specific components on overall application performance has not been visible. Developers building these high-performance components could not be rewarded and given the key to reuse the components. Management solutions give operations, architects, and developers a common language to discuss performance and development issues. This feedback loop delivers better understanding of component behaviors, faster tuning of under-performing components, and identification of best practices for component building. These benefits give businesses a means to make component reuse more of a core endeavor for their development staffs. Also, a solution that collects granular information facilitates long-term planning, general system tuning, and development. By knowing, for example, the amount of time an EJB spends in a queue, the most failing EJB, or the most invoked EJB, IT can optimize both the production infrastructure and application code to prevent failure altogether.

Furthermore, with the rising importance of Web services as executable components on the application server platform, enterprises must now also incorporate Web services as a well-understood and managed part of their application architecture. A solution that provides granularity of visibility and control provides the means to measure and manage the quality of service of Web services. Web services deployed on the application server will be visible, measurable, and manageable so that IT can confidently deliver high-quality Web services wherever they are needed.

## Conclusion

Enterprises cannot fully realize the business benefits delivered by their IT infrastructures without proactive management capabilities. The resulting business risks are enormous, as WebSphere-based services determine the business' ability to compete. Three degrees of IT manageability are necessary to control these risks:

1. ***Proactive management of the business services:*** Monitoring, troubleshooting, and correcting infrastructure problems based on their impact and role during the delivery of the business service
2. ***WebSphere Application Server management:*** Detailed visibility and control not just of the application server itself, but the custom components that encapsulate the business service.
3. ***Correlation with end-user experience:*** Enabling management from the end user's perspective.

IT needs a management solution that provides the power to make all three requirements a reality for the enterprise.

THE LARGEST INTERNATIONAL

# XML CONFERENCE & EXPO IN THE WORLD!

XML EDGE conference&expo

**JUNE 24-27**
JACOB JAVITS
CONVENTION CENTER
**NEW YORK, NY**

**OCTOBER 1-3**
SAN JOSE
CONVENTION CENTER
**SAN JOSE, CA**

## WIN A $35,000 LUXURY CAR!

ATTENDEES WILL BE INVITED TO TAKE A GOLF SWING TO WIN AND RIDE OFF IN A $35,000 LUXURY CAR!

## XML-NEXT G OF ENTERPRISE DEPLOYMENT

**REGISTER ONLINE TODAY**
FOR LOWEST CONFERENCE RATES
EARLY SELL-OUT GUARANTEED!

**VISIT WWW.SYS-CON.COM**

## Focus on XML

XML is today's essential technology to develop and deploy Web services. Here's your chance to learn from expert practitioners how XML is making the next phase of the Web a reality.

Focus on standards, interoperability, content management, and today's new quest for more efficient and cost-effective Internet and intranet-enabled business process integration.

Focus on XML during information-packed days while you enjoy an XML/Web Services Keynote panel, comprehensive conference sessions, and an unparalleled opportunity to exchange ideas with peers and industry leaders.


## A Sampling of XML-Focused Sessions

- XML STANDARDS - AN OVERVIEW
- OASIS STANDARDS UPDATE
- UBL - A UNIVERSAL BUSINESS LANGUAGE
- BRINGING XML TO PKI
- LINK MANAGEMENT WITH XLINK
- PRACTICAL XSLT AND XPATH
- ENTERPRISE CONTENT MANAGEMENT WITH XML
- XML IN THE ENTERPRISE AND INTER-ENTERPRISE WORLD

ROBERT MISKLA
XML CHAIR
BOARD OF DIRECTORS, OASIS
INDUSTRY EDITOR
*WEB SERVICES JOURNAL*

## Featuring...

- UNMATCHED KEYNOTES AND FACULTY
- THE LARGEST INDEPENDENT JAVA, WEB SERVICES, AND XML EXPOS
- AN UNPARALLELED OPPORTUNITY TO NETWORK WITH OVER 5,000 I-TECHNOLOGY PROFESSIONALS

## Who Should Attend...

- DEVELOPERS, PROGRAMMERS, ENGINEERS
- I-TECHNOLOGY PROFESSIONALS
- SENIOR BUSINESS MANAGEMENT
- SENIOR IT/IS MANAGEMENT/C LEVEL EXECUTIVES
- ANALYSTS, CONSULTANTS

## Hear these thought leaders in interactive, cutting-edge keynote addresses and panels...

JEAN-FRANCOIS ARNAUDIC
SENIOR VP R&D, ILOG
FORMER CHAIRMAN, W3C

TYLER JEWELL
PRINCIPAL TECH. EVANGELIST • BEA

DAVID LITWACK
CEO
SILVERSTREAM

ANNE THOMAS MANES
CTO
SYSTINET

BARRY MORRIS
CEO
IONA

RICK ROSS
FOUNDER
JAXALOBBY

PATRICIA SEYBOLD
FOUNDER & CEO
SEYBOLD

**For Exhibit Information**
CONTACT: RICHARD ANDERSON
135 CHESTNUT RIDGE RD.
MONTVALE, NJ 07645
201 802-3058
RICHARD@SYS-CON.COM

**SPONSORED BY:**

IONA | END 2 ANYWHERE™
ADOS
bea
TogetherSoft
SilverStream
Actional
MERANT
Altaworks
PolarLake
ALTOVA

**MEDIA SPONSORS**

Federal Computer Week
Webservices.org
XML TIMES.com
Java Skyline
CF Advisor
WebServicesMall
XML.ORG
WIRE
jCert
wrox
OASIS
XML.ORG
SD Times
JAVA
wireless
XML
WebLogic
WebServices
WebSphere
ColdFusion

**OWNED AND PRODUCED BY**

SYS-CON MEDIA
SYS-CON EVENTS

*The right tools for WebSphere development*

# Welcome to Web Services

BY RON BEN-NATAN


ABOUT THE AUTHOR

Ron Ben-Natan is CTO at ViryaNet Inc. and has been architecting and developing distributed applications for over 15 years. Prior to that he worked for companies such as Intel, AT&T Bell Laboratories, Merrill Lynch, and J.P. Morgan. He has a PhD in Computer Science in the field of distributed computing. He has authored and coauthored numerous books, including *IBM WebSphere Starter Kit* and *IBM WebSphere: The Complete Reference* (Osborne/McGraw-Hill) and *Integrating Service Level Agreements* (John Wiley & Sons).

E-MAIL

rbennata@hotmail.com


Welcome to the newest addition to *WebSphere Developer's Journal*. This column is devoted to the subject of Web services within the WebSphere family of products.

You must have heard about Web services, and maybe even written a few; well, it's time to address this important topic in the context of WebSphere development. In this monthly column I'll cover the entire software life cycle, including using WebSphere Studio tools to develop Web services, testing and packaging of Web services, deployment on the WebSphere application server, and publishing Web services on registries. I'll also cover how Web services are used within a client application, how to use proxies to invoke Web services, and how you can use a registry to look up a service provider. Since the topic of Web services is so central to the Web computing paradigm, I'll also cover many advanced topics, including the use of Web services in a mobile computing environment, tools and frameworks published on the IBM developerWorks site, and even interoperability with platforms such as Microsoft's .NET – in which Web services play a very central role.

Let's start with what Web services are and how they came to be. Web services are functional elements deployed on a node on a network and accessed over the Internet. This description is quite generic and doesn't say much; what makes an element a Web service is the *how*, not the *what*. Web services are based on a set of *standards* – specifically, the Simple Object Access Protocol (SOAP), the Web Service Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI). SOAP is the protocol by which a remote client can invoke the functionality implemented by the Web service. Developers of Web services use WSDL to define the metadata describing the Web service and remote clients use WSDL to learn what arguments are required for invoking the Web service (as well as other things required to make the remote call). Web services providers use UDDI to publish their Web services and clients of Web services use UDDI to discover where these Web services (and the metadata describing them) reside.

Web services seem to be the hottest technology these days – judging by the number of books, articles, analyst publications, and so on. Vendors and users alike are converging on this new set of technologies that promises simple and standards-based interoperability among distributed decoupled systems. It is by far the most talked about topic in the software technology space, and the investments being made in Web services technologies are considerable.

If you think you've heard it all before, you're absolutely right. The paradigm of remote invocation and application interoperability is certain-

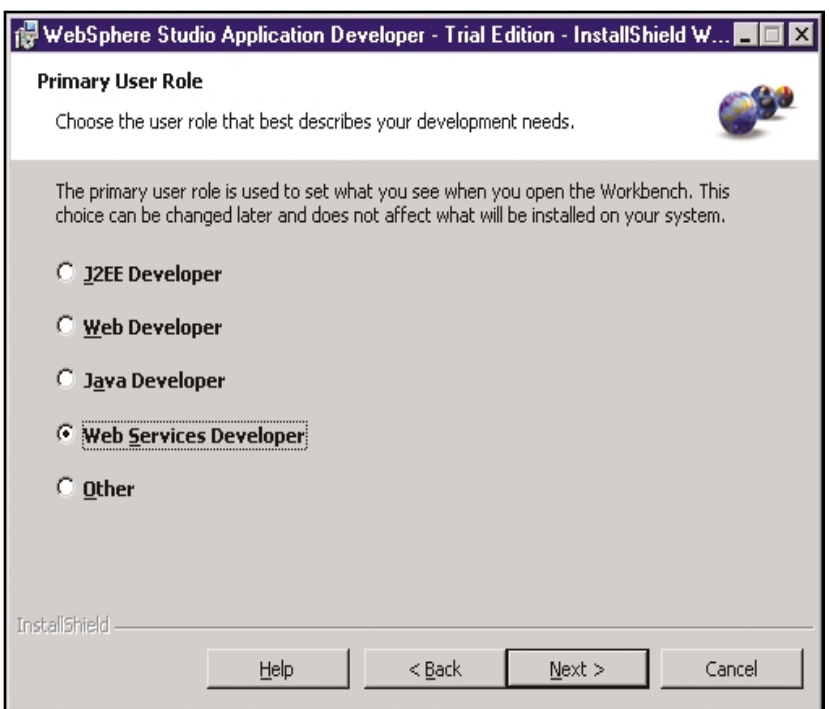



**FIG. 1: INSTALLING WSAD: SELECTING THE PRIMARY USER ROLE**

ly not new, and is full of failures. You need to go back only five years to remember the battles between the Object Management Group's Common Object Request Broker Architecture (CORBA) and Microsoft's Distributed Component Object Model (DCOM). Neither of these technologies, despite being good for interoperability between applications on a LAN, is really suited for interoperability on the Internet. More importantly, both are useful in tightly coupled environments but neither is really suited for the decoupled cooperation that can be achieved over the Web. Most importantly, they are competing technologies – you either used CORBA or DCOM. It's certainly not easy for applications using these interoperable technologies to interoperate (I wonder if there's a Dilbert comic on this subject?).

## Why Web Services?

So what makes Web services different? Why are they so important and why do so many people believe that they are the cornerstone of the future of distributed computing? Why do all these people think that this time it will work? It's quite simple – Web services are *simple*, they're *built for the Web*, and they're based on de-facto *standards*.

Web services technologies – SOAP, WSDL, and UDDI – are relatively simple. I use the word relatively because they are certainly not trivial and it takes a bit of learning to master them. But they are certainly much simpler than other distributed computing technologies that existed before them. The history of computing has proven that simple technologies succeed and highly complex technologies fail. Just look at the World Wide Web itself – can you imagine anything simpler than the HyperText Transfer Protocol (HTTP) and HyperText Markup Language (HTML)?

Web services are built from the ground up as Web technologies. SOAP, WSDL, and UDDI are all based on the eXtensible Markup Language (XML), which has taken the world by storm and become the *lingua franca* of the Web. If you know XML, then you can understand SOAP, WSDL, and UDDI. If you understand advanced XML-related technologies like XML schemas, then you are even better off – but that's not mandatory. SOAP is an invocation and routing protocol involving XML documents that are usually delivered over HTTP to an endpoint deployed on the Web. While the underlying transport isn't limited to HTTP in SOAP, the vast majority of Web services are accessed over HTTP.

And finally, the most important factor: the fact that Web services have become de-facto standards. Regardless of whether you're a seasoned software professional or a relative newcomer, I'm sure you're aware of the phenomena involving the Microsoft camp and the non-Microsoft camp (which then breaks down into some additional but smaller rivalries). Many software vendors have created "standards" in the past through consortiums, organizations, and more, but when was the last time you remember companies like IBM, Sun, and Oracle *together* with Microsoft pushing the very same set of technologies and promoting them as standards?

Well, it's true; all these companies are cooperating and driving the standards being formed for Web services. In fact, the two most dominant companies in this campaign are IBM and Microsoft, and many of the definitions and specifications result from joint work between IBM and Microsoft. As atypical as it is, the fact that IBM and Microsoft are cooperating on a technological basis gives anything produced by this process a tremendous amount of clout; and the result is widespread industry acceptance. Analysts, too, are in agreement. IBM's research indicates that Web services have become a top customer-buying criteria for 2002/2003 and a research report issued by Gartner in April 2001 estimates there's a 0.8 probability that IBM and Microsoft will exert leadership in defining Web services standards.

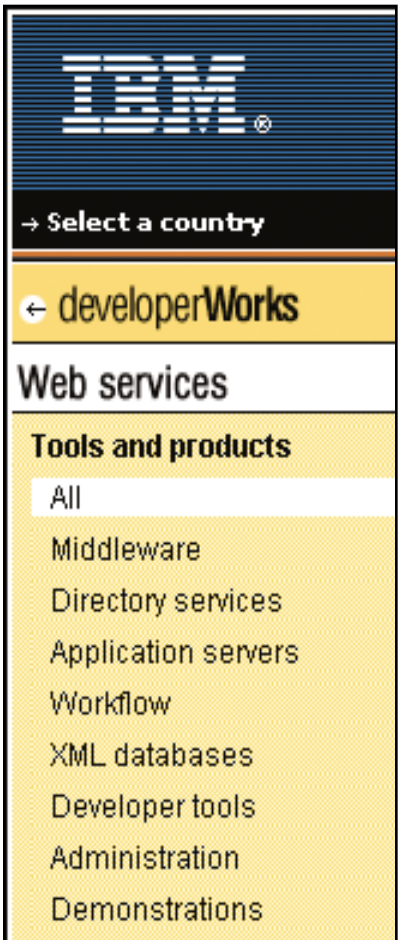



**FIG. 2: WEB SERVICES ZONE ON DEVELOPERWORKS - TOOLS AND PRODUCTS**

Anyone following analysts' estimates knows that for Gartner to give a 0.8 probability means that it's practically a done deal. As you know, such analyst reports often create reality, in addition to informing the business community on the state of the market.

## Web Services Within the WebSphere Product Family

IBM is working on all of the Web services standards – SOAP, WSDL, and UDDI. As a vendor, IBM is committed to delivering an implementation of each of these standards, including development tools, deployment tools and runtime servers. Obviously, because the WebSphere product family is the cornerstone of IBM's product family, Web services star in all of the WebSphere product line.

As of version 4.0 of the WebSphere Application Server (WAS), support for Web services is incorporated into every edition of WAS. This means that the Web services libraries come within WAS, that Web services packaged as enterprise applications and Web modules can be deployed over WAS, and that WAS takes care of the SOAP marshaling and unmarshaling.

In addition to WAS, you can use the WebSphere UDDI Registry to publish your Web services for others to discover. In version 4.0 this isn't part of the WebSphere application server but it should become an integral part of WAS in a future version.

In terms of development, Web services are supported through the WebSphere Studio Integrated Development Environments (IDEs), as well as toolkits. Both of the new WebSphere Studio tools – WebSphere Studio Application Developer (WSAD) and WebSphere Studio Site Developer (WSSD) – include full support for developing, testing, packaging, deploying and using Web services. For example, when you install WSAD you're asked to select your primary user role (see Figure 1). Guess what – Web services developer is right up there along with J2EE developer, Java developer. and Web developer.

In terms of toolkits, it's important to mention at least a few. The Web Services Toolkit (WSTK) is available from AlphaWorks (www.alphaworks.ibm.com/tech/webservicestoolkit) and includes a runtime environment, tools, samples, and documentation for Web services. Most of the toolkit has been assimilated into WSAD but it's useful if you aren't yet using WSAD or WSSD. The toolkit can even be used if you're still using VisualAge for Java and WAS 3.5.

If you develop for hosted or ASP environments you can use the Web Services Hosting Technology toolkit for supporting Web services in a hosted environment. This toolkit helps you do common tasks such as provision services, monitor, and charge based on usage, and so on. And finally, the Web Services Gateway can act as a redirector for Web service requests and can be used to set up multiple routing paths through intranets, extranets, and the Internet – all for building Web service–based interaction models.

Don't think that this is it. I recently went to the developerWorks site, which has a zone dedicated to Web services. I clicked on the All category (see Figure 2) just to get a feel of how many tools and products are listed. There were 119 downloads! You could spend a lifetime just going through all this. Instead, I'll use this column to show you some of the gems.

And finally, there's WoW. Web services on WebSphere (WoW) is an IBM initiative to establish the WebSphere community as the premier Web services solution providers. The initiative is in its infancy, and it's still unclear exactly what it means for developers, but it talks about technical support, marketing programs, business development initiatives, and more. Once it becomes a little clearer, I'll be sure to include a column discussing it.

## What Can You Expect Here?

This is a monthly column dedicated to the subject of Web services on WebSphere. You can expect articles showing you how to develop Web services using WebSphere tools and how to build applications that discover and make use of Web services using WebSphere tools. You can also expect articles devoted to the packaging and deployment of Web services on WAS, publishing services in private or public registries, and advanced topics such as workload management for Web services. I'll also try to insert examples and samples of Web services. I won't teach you SOAP, WSDL, or UDDI, nor show you lengthy XML documents. There are enough great resources on these technologies and you should have a basic understanding of them to make the most of this column. I look forward to seeing you here every month.

# Increasing your balance at the 'Knowledge Bank'

BY CRAIG FLANNAGAN

Picture it: you want to build a solution, so you visit the knowledge bank, only to find out that your organization has 'insufficient funds to complete this transaction'. To find out how to increase your knowledge bank balance, read on.

From a developer's perspective, of all the mediums of learning, one in particular stands out as a hub to which other forms of learning can be linked – the user group. You simply have to look at how much is spent annually on business conferences around the world to realize that human interaction is a necessary part of learning.

Human interaction through the concept of a user group can be likened to a "knowledge bank." The group is made up of individuals who possess knowledge about the subject. The sum of that knowledge can be thought of as the balance in the knowledge bank. The group meets, members share information, and exchange ideas and interests with each other. This leads to each member acquiring the knowledge of other members, and increasing their own personal knowledge bank. From time to time, the group brings in an outside expert to deposit new knowledge in the bank. This knowledge is absorbed by the members of the group and the balance of the group's knowledge increases. The individuals in the user group are compelled to increase their personal knowledge so that they may bring more to the group. As new members become part of the group, the bank of knowledge grows, and the cycle continues.

WebSphere User Groups exist around the world, and they mimic the cycle described above. In the early stages of a group, the focus is on acquiring enough knowledgeable members to share their insights on WebSphere. In the first meeting, the members may share the experiences they've had, or one member may present his or her experiences with WebSphere. As the group grows, the bank of knowledge grows. A subject matter expert may be asked to speak on a particular topic. Lately, the popular topics at WebSphere User Groups have been IBM WebSphere Studio and eclipse.org. The knowledge transfer is not unidirectional – the outside subject matter experts learn from the group as well. The group's feedback may be used to better design the next version of a product or solution from IBM, an IBM business partner, or an ISV. Some members may absorb information more quickly than others, and they take the time to explain the topic further at the break, or after a meeting. It's exciting to see a group of individuals learning, growing, and interacting. As a tool for overcoming business challenges, a WebSphere User Group is as knowledgeable as its most knowledgeable member.

WebSphere User Groups don't just meet in person. Many have e-mail discussion forums (check out websphere.org's listserv for one of the largest), newsgroups, or chat rooms dedicated to discussing WebSphere.

The individuals who get the most out of a WebSphere User Group are those who increase their personal knowledge banks by learning more about WebSphere. A great place to increase your personal knowledge bank balance is the WebSphere Developer Domain (www.ibm.com/websphere/developer).

Within the knowledge bank analogy, the WebSphere Developer Domain is the online knowledge bank Web site. You can make withdrawals by downloading the latest code, webcasts, tutorials, technical articles, Redbooks, and more. These withdrawals increase your knowledge about the WebSphere Software Platform for e-business. Make a knowledge deposit into a forum, or through the Feature Request Database (which allows you to suggest features for upcoming releases of WebSphere products). Check your knowledge balance by visiting the WebSphere certification Web site (www.ibm.com/certify) and take the precertification, self-assessment tests.

The WebSphere Developer Domain also houses the WebSphere User Group community Web site (www.ibm.com/websphere/usergroup). The WUG community site allows individuals to join or start a WebSphere User Group anywhere in the world. The site also links to websphere.org – the Worldwide WebSphere User Group portal. As a leader of a WebSphere User Group, websphere.org provides hosting of your WebSphere User Group site, suggested meeting topics, and day-to-day consulting for leaders.

Over the past 20 years, the ability to share information has grown at a phenomenal rate. Fortunes have been made by businesses that focus on data management and data mining. There's an entire school of thought called Knowledge Management and the concept of "Knowledge Capital," which states that knowledge, in and of itself, has a monetary value.

From Boston to Brisbane and San Francisco to Singapore, WebSphere User Groups exist for one reason, to increase the knowledge bank of the community around WebSphere – a community which includes everyone from students to enterprise developers, and from IBMers to industry analysts.

ABOUT THE AUTHOR

Craig is currently the marketing manager responsible for executing the IBM WebSphere developer campaign. He creates and manages marketing programs targeted at making WebSphere the world's leading e-business development platform, including the WebSphere user group program. Prior to this role, Craig was the WebSphere software marketing manager for IBM Canada.

E-MAIL
cflannag@ca.ibm.com

SPOTLIGHT ON:

# Richard Arone,
## Special Projects Editor, *WSDJ*

*Spotlight on WebSphere Success Stories* will be ***WSDJ***'s way of honoring those who are shaping the present and future of technology. If you work at a company that's running WebSphere or are in some way involved with the sales, marketing and development of WebSphere, feel free to submit a nominee. Just send us a bief description of the actions/performance you admired, including the name(s) of the person or team. They can be marketing, technical, or line-of-business people, or anyone who has excelled in the WebSphere arena within or outside your company. You can't nominate yourself or your supervisor. Also, don't forget to let us know when this all happened. Send the details to websphere@sys-con.com.

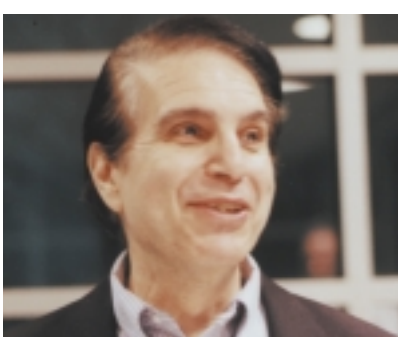
# *The birth of a new magazine*

BY JACK MARTIN

As this issue of ***WebSphere Developer's Journal*** goes to press, times are pretty tough for the majority of consumer and business magazines. Many titles are contracting or disappearing altogether. Yet here at ***WSDJ*** we're already on our third issue, and all the early signs point to a great publishing success story.

Many things have contributed to our success: the tremendous popularity of the WebSphere product line, the environment's use of open standards, the endless fountain of new products that work with WebSphere, and SYS-CON Media's expertise at developing leading edge technology publications. Last but not least, one man behind the scenes has been a great resource and friend to ***WSDJ***: IBM's Richard Arone.

Richard recognized early that WebSphere was going to change not only how people utilized Web-based technology, but also how IBM was positioned in the marketplace and how its business partners could leverage this technology.



I first met Richard Arone about six years ago when he came to my office to negotiate my IBM ISV contract. I told Richard then that my chief concern was that IBM was so much larger than my company that we could easily be devoured by the time or terms of the agreement. He took the time to explain how the process worked and that IBM would be fair to us. I put my trust in Richard and he has never let me down once. He became the human face of our relationship with IBM, providing us with a road map and a flexible vision that helped us chart a successful course for our business.

Two years ago, Richard introduced me to the WebSphere product line. I was aware of WebSphere but I really didn't know what it was, what it did, or how it would affect my business. Just as during that first meeting with him, Richard took the time to explain how IBM and my company could work together to promote WebSphere and how it could benefit both companies.

His enthusiasm helps shape every issue of ***WebSphere Developer's Journal***. Richard introduces ***WSDJ*** to WebSphere people within IBM and other businesses, and is the ideal Special Projects Editor. For this reason I've chosen him as the subjct of this first Spotlight on WebSphere Success.

## WEBSPHERE NEWS

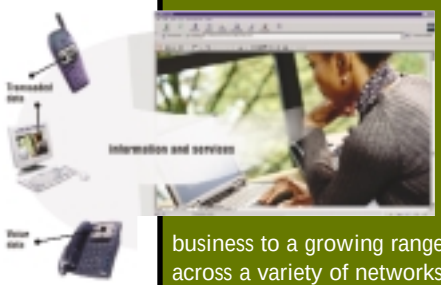
### COMPUTER-GENERATED SOLUTIONS TO LAUNCH FIRST VIRTUAL IBM WIC

(Saint John, New Brunswick) – Computer Generated Solutions, Inc. (CGS), a leader in online learning and technical training, will launch the first Virtual IBM WebSphere Innovation Center (V-WIC), a state-of-the-art education and knowledge portal. The V-WIC at CGS will provide IBM customers and business partners with remote, online technical training for their e-business platform. IBM WebSphere's open infrastructure software manages high-volume transactions and integrates different business applications across a variety of computing systems.
www.cgselearning.com/websphere

### IBM PARTNERS WITH SIEBEL TO DELIVER SOFTWARE FOR APPLICATION NETWORK

(Somers, NY and San Mateo, CA) – IBM and Siebel Systems, Inc. have announced plans to deliver IBM's CrossWorlds integration software for Siebel Application Network. The IBM CrossWorlds technology is an integral part of IBM's WebSphere platform.

Working closely with Siebel Systems, the leading provider of e-business applications software, IBM will upgrade its robust, scalable integration server to run standards-based business processes delivered by Siebel Application Network. As a result of the product integration, the IBM CrossWorlds InterChange Server will become an integration server for Siebel Application Network. By using IBM CrossWorlds integration software, customers can reduce integration costs and complexity to maximize their return on investment.
www.siebel.com



### MQSOFTWARE ANNOUNCES TELEMETRY INTEGRATION SOLUTION

(Minneapolis) – MQSoftware, Inc., a leading provider of EAI and middleware messaging application solutions for WebSphere MQ-based messaging products, has announced that its flagship Q Pasa! middleware messaging software is now part of an integrated solution with IBM WebSphere MQ Integrator and Arcom Control Systems, a supplier of pervasive device technology. The end-to-end solution offers organizations the ability to manage data from field network–connected devices back into enterprise systems, providing greater flexibility and control over data collected from pervasive devices.

The telemetry solution streamlines the process of collecting, distributing, and utilizing information from centralized and remote operations, thereby improving data availability across the enterprise.

In addition, MQSoftware is leading the knowledge-transfer effort by rolling out an education program that will help companies understand how the respective technologies can be applied and implemented for their specific projects.
www.arcomcontrols.com,
www.mqsoftware.com

### NEW SOLUTION ALLOWS MIGRATION FROM VISUALAGE FOR JAVA TO WSAD

(San Francisco) – Rational Software has announced a new automated solution that enables development teams in IT organizations to rapidly migrate from VisualAge for Java to WebSphere Studio Application Developer by moving software assets from the VisualAge for Java/Envy repository to Rational ClearCase.

By providing seamless migration support, the new tool assists project leaders who want to take advantage of the new developer productivity tools and team collaboration capabilities that support WebSphere Advanced Version 4.0.

CodePro Migrator for Rational ClearCase is available immediately from Instantiations at www.instantiations.com/codepro/migrator.
www.rational.com

### WILY TECHNOLOGY DELIVERS SOLUTION FOR WEBSPHERE APP SERVER 4.0

(Nashville, TN) – Wily Technology, a leader in enterprise Web application management, has announced a comprehensive performance management solution for the IBM WebSphere platform with the addition of a new PowerPack for WebSphere to the Introscope family of products.

The new product works with Introscope, Wily's flagship Web Application Monitor and Wily's family of management solutions, including Introscope SQL Agent for DB2 and the newly announced Introscope Power-Packs for CICS and WebSphere MQ. It combines Introscope's ability to monitor the performance of production J2EE components such as EJBs, JSPs, and servlets with measurements specific to WebSphere. Developed in cooperation with the WebSphere Development Team, PowerPack for WebSphere is preconfigured to offer instant component-level monitoring of WebSphere, including JDBC Connection Pools, HTTP sessions, and servlet threads.
www.wilytech.com

## IBM Simplifies Wireless for the Enterprise

(Armonk,, N.Y.) – IBM has announced the newest version of its popular WebSphere software family, WebSphere Everyplace Access. The new software will give employees access to data over any device on any network. Part of IBM's pervasive computing initiative, this is the latest in a string of moves by IBM to help companies extend e-business to a growing range of emerging devices across a variety of networks and connectivity.

WebSphere Everyplace Access helps companies embrace pervasive computing by allowing them to manage a variety of information needs, an ever-increasing multitude of devices and variable connectivity modes; and extend existing investments in applications, portals, and skill sets. This will be increasingly important as the industry moves toward multimodal interaction using several forms of input and output, such as voice and text, simultaneously, in the same application.

This new software platform provides the core services necessary to build, deploy, manage, and extend existing e-business and core line-of-business applications to mobile devices. As with all IBM technologies, Websphere Everyplace Access embraces key industry standards, such as SynchML, WAP, and LDAP. www.ibm.com

## WebSphere Developer's Journal | Coming Next Month...

— ***The Future of WebSphere***
*More from Don Ferguson on where WebSphere is going*

— ***Building Data Access Objects***
*An in-depth look at how VisualAge for Java makes data integration a snap*

— ***"Putting your money where your mouth is..."***
*Service Level Agreements and their effect on the pervasive computing environments of tomorrow*

— ***Rational XDE***
*Rational's first Java-focused UML tool looks good for WAS developers*

— ***Message-driven Beans in WebSphere 5.0, part 2***
*Install and execute your MDBs*

## WebSphere Developer's Journal | Advertiser Index...

# Customer Loyalty

## *Understand it – and keep it*

BY LOUIS VAN DE WATER

Years ago, retailers grew by opening new stores, thus gaining new customers and sales revenue from each new locale. As many retailers added stores, an overstored condition resulted. Retailing success then depended more on increasing revenues by maintaining customer loyalty and attracting customers from less successful retailers. This customer loyalty was achieved by providing acceptable products and competitive prices, and giving the customer a positive, even exciting, experience in the store. Thus, success came to the retailer that made customers feel special while providing convenience at an acceptable price.

The scrap heap is full of out-of-business retailers that didn't understand what drives individual customer loyalty and what motivates them to shop with one retailer versus another. It's not uncommon for a retailer to have as many former customers as current customers and not know why half of their customer base left them. The traditional method of attracting customers has been to advertise in newspapers/magazines and by mass mail to anyone who ever signed up for the retailer's credit card. This lack of personalized understanding of their customer's motivations leads to simple price-oriented promotions. The retailer that wins on price must also have large sales revenue to make up for its thin margins, so there are few winners and many losers for those that compete primarily on price. So if price is a losing strategy for most retailers, how can they make their customers feel special and provide convenience so as to succeed without having to just have the lowest price?

Having differentiated and good quality products overcomes price shopping, but to maximize success a retailer must focus on more than product uniqueness and price. Today's retailer also needs to understand changing buying patterns. Consumers have access to information like never before. Studies have shown that many consumers look at an item on the Web before buying it at a store. For some items, 60% of those who purchased that item in a store first researched it on the Web and 40% of those who bought on the Web first saw the item in a store. These two examples aren't unusual but are typical of today's more knowledgeable consumer who uses multiple channels in the process of making final product selections. A retailer that either doesn't offer the consumer multiple channels to buy or separates its channels into essentially different businesses is out of step with the way a growing number of consumers shop. This type of retailer could more likely find itself losing share and maybe becoming the next retailer to go on the scrap heap.

It's not easy to identify a specific customer's purchases unless he or she pays with the retailer's own store credit card, or use a frequent shopper card. These types of sales represent a small volume of a store's sales. Catalog, 800 phone number, and Internet sales require purchasers to identify themselves. Thus, by offering these methods, an understanding of these customers' buying preferences and buying frequency can be obtained. This understanding allows for a more customized marketing effort. For example, if a consumer started to show a decrease in their buying pattern, attempts could be made to understand why and potentially save the loss of a customer.

The solution is for the retailer to offer multiple sales alternatives such as the Internet, call center, wireless, or kiosk, as well as the standard brick-and-mortar stores to make their offerings available whenever, wherever, and however the customer desires.

MerchantReach is the architecture for a strategy that integrates the entire current enterprise from end to end, including the Internet and point of sale, while facilitating the addition of new channels such as wireless and kiosks. When customers enter a store, they can not only look at what's in the store but see what other items are available via a kiosk or Internet connection embedded in the point-of-sale register. Also, a sales associate could determine if an out-of-stock item is available in a nearby store or could be shipped from the retailer's warehouse. When not in the store, the customer can access the retailer's offerings via a PC, PDA, or advanced cellular phone, or this information can be made available by voice as well as text or graphics. It's even possible to have an e-mail or voice message automatically sent to the customer with information concerning a retailer's offering. For example, a voice mail could notify a customer that an item the customer was particularly interested in had just arrived. The software foundation for MerchantReach that makes all of this possible is WebSphere.

WebSphere is the retail industry's most comprehensive e-business software. It's the premiere platform that gives retailers the ability to create and deploy their e-business as well as provide the ability to scale their growth up to industrial strength volumes. Many retailers have found out the hard way that their Internet software could fail at relatively low volumes. WebSphere is the underpinning of the MerchantReach architecture and thus the basis for a retailer's success in a growing multichannel world.

ABOUT THE AUTHOR...

Louis Van De Water has earned the top sales award in IBM and was also named of one of its top sales managers. He currently has overall sales management responsibility with IBM for the retail and media industries in New York and New Jersey.

E-MAIL
lfvande@us.ibm.com