WebSphere® DEVELOPER'S JOURNAL

The World's Leading Independent WebSphere Developer Resource

WebSphereDevelopersJournal.com

MARCH VOLUME 1 ISSUE 2

SYS-CON MEDIA

PROLIFICS

WWW.PROLIFICS.COM

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
Jane Edwards, Joe Farsetta, John Goodson, Michael Maas, Jack Martin, Jim Martin, Jackie McAlexander, Morgan Smyth, Antonio Villafana

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 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
BILL DARRON
CONTENT EDITOR\ LIN GOETZ
JDJSTORE.COM ANTHONY D. SPITZER
CUSTOMER SERVICE ANTHONY D. SPITZER
CUSTOMER SERVICE LIASION PATTI DEL VECCHIO

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

Reader Opportunity Information

BY JACK MARTIN

I'd like to thank everyone for the feedback and kind words about our premier issue. I found it both complimentary and encouraging. So in this issue, number two, we're pledging to try even harder to deliver more ROI (Reader Opportunity Information) every month.

It's been my experience that business decisions have been based on the other ROI (Return on Investment) since… forever! But now we have an official race to the ROI bottom line.

The ***WebSphere Developer's Journal*** is not one to turn its back on the latest trends. We like to jump right into the mix, too. We have all sorts of opinions and commentary, facts and figures, all with the thought that developers, system administrators, ISVs, executives, and market analysts need this kind of knowledge to make sound business decisions.

I want your time spent with this magazine to reap a healthy return in knowledge. Apply it to what you do every day. We're trolling the marketplace for valuable insights pertinent to WebSphere and its supporting technologies. This way you'll get code and technical advice with a dash of business purposefully placed and conveniently contained within this publication.

In this spirit we introduce, "Why They Buy" (see page 18), a recurring column that looks at customer rationale in consummating the deal. Get your customers to the Yes/Yes stage and walk them through the decision process. This information becomes powerful knowledge to the savvy software designer who wants to understand business demands and how a company makes a technology investment.

I had the pleasure of interviewing Don Ferguson, "the Father of WebSphere," in his Somers, New York, office. He's a gentleman who understands the business use of technology better than most technical/engineering types. The first part of his interview appears this month. It was an incredible amount of information and clearly requires a continuance in future issues. Don discusses his insight into WebSphere, Portlets, Flow Composition Modeling, and more. He tells us where it may all end up two to three years out. Isn't that something you'd like to get your hands on?

From where I sit, there's no better time to be in the WebSphere business than today! I believe that the current U.S. economic condition is a fertile field for the technology sector.

There are unprecedented IT budgets for federal, state, and local governments. Whether it's for homeland defense, military initiatives, or public safety, there appears to be great opportunity as many government agencies review their ability to protect and serve our citizens. Systems need to be integrated. Architecture needs to be reassessed. Standards must be employed. Those holding the purse strings are now better educated in technology. They are also more experienced at preparing bids and understanding how a technology project is implemented.

In the private sector, companies always need to save money. In the push for faster, better, and cheaper (meaning less expensive), the answer is found in leveraging technology.

Begin with improved communications with suppliers. Orders can be placed, manufacturing tracked, and delivery confirmed through WebSphere and its ancillary products. Internally, workflows can be streamlined and peers updated on shared projects. HR can get information disseminated in record speed. Custom portals can deliver the right mix of data to the desktop. Customer service is improved by greater flexibility in comparing data, sharing records, and customizing offers based on up-to-the-moment conditions. And WebSphere is on target to deliver the goods.

IBM has charted its course and we, the developer community, have awakened to the possibilities – the full capabilities of WebSphere in our future. The opportunities are as numerous as we are. Whether it's in groundbreaking, pervasive inventions, the creation of the many portlets that will combine seamlessly to deliver customized content, or something else just hitting our imaginations, we have the tools and the standards to get the job done.

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 performance-optimized JDBC applications for DB2

Making Your WebSphere Apps Run Faster and Jump Higher

BY JOHN GOODSON

What's the best way to develop and fine-tune your WebSphere applications to run faster, jump higher…and make fewer trips to the database?

ABOUT THE AUTHOR

John Goodson, vice president of product operations for DataDirect Technologies. He leads the strategy and development efforts for the company's ODBC, JDBC, ADO, and XML data connectivity technologies. For nearly 10 years, John has worked with Sun and Microsoft on the development of database access standards, and is active in the JDBC Specification expert group and Java Rowset expert group.

E-MAIL
john.goodson@datadirect-technologies.com

When creating optimized WebSphere applications that access data on your DB2 database, what kinds of challenges do you face? Creating DB2-enabled WebSphere apps involves the Java Database Connectivity (JDBC) API, which can be a challenge in itself. This article will look at various DB2-enabled applications and offer some guidelines to help your WebSphere applications run more efficiently when they connect to a DB2 database.

Designing WebSphere Applications

The tips in this section will help you optimize system performance when designing your applications.

START BY PLANNING YOUR CONNECTIONS

Connection management is important to application performance, so optimize your application by connecting once and using multiple statement objects, instead of performing multiple connections. Avoid connecting to a data source after establishing an initial connection.

Although gathering driver information at connect time is a good practice, it's often more efficient to gather it in one step rather than two. For example, some apps establish a connection, then call a method in a separate component that reattaches and gathers information about the driver. Applications that are designed as separate entities should pass the established connection object to the data collection routine instead of establishing a second connection.

Another bad practice is to connect and disconnect several times throughout your application to perform SQL statements. Connection objects can have multiple statement objects associated with them. Statement objects, which are defined to be memory storage for information about SQL statements, can manage multiple SQL statements.

You can improve performance significantly with connection pooling, especially for applications that connect over a network or through the Web. Connection pooling lets you reuse connections. Closing connections doesn't close the physical connection to the database. When an application requests a connection, an active connection is reused, thus avoiding the network I/O needed to create a new connection.

Plan connection and statement handling before you implement the application. The time you spend thoughtfully handling connection management will lead to improved application performance and maintainability.

BE CAREFUL WITH COMMITS

Committing transactions is extremely disk I/O intensive and slow. Always turn autocommit off by setting:

```
WSConnection.setAutoCommit
(false)
```

What does a commit actually involve? The DB2 server must flush back to disk every data page that contains updated or new data. This isn't a sequential write, it's a searched write to replace existing data in the table. By default, Autocommit is on when connecting to a data source, and Autocommit mode usually impairs performance because of the significant amount of disk I/O needed to commit every operation.

Although using transactions can help application performance, don't take this too far. Leaving transactions active can reduce throughput by holding locks on rows for long times, preventing other users from accessing the rows. Commit transactions in intervals that allow maximum concurrency.

AVOID DISTRIBUTED TRANSACTIONS

DB2 supports distributed transactions – that is, transactions that span multiple connections. Distributed transactions are at least four times slower than normal transactions due to the logging and network I/O necessary to communicate between all the

components involved. Unless distributed transactions are required, avoid using them. Instead, use local transactions whenever possible.

For the best system performance on DB2, design the application to run under a single Connection object.

Retrieving Data

To retrieve data efficiently, you should return only the data you need, using the most efficient method possible. The following guidelines will help you to optimize system performance when retrieving data with JDBC applications.

AVOID RETRIEVING LONG DATA

Unless it's necessary, applications should avoid requesting long data because retrieving long data across a network is slow and resource-intensive. Most users don't want to see long data. If the user does want to see these results, then the application can query the database again, specifying only the long columns in the select list. This method allows the average user to retrieve the result set without having to pay a high performance penalty for network traffic.

Although the best method is to exclude long data from the select list, some applications don't formulate the select list before sending the query to the JDBC driver (that is, some applications select * from <table name> ...). If the select list contains long data, then some drivers must retrieve that data at fetch time even if the application doesn't bind the long data in the result set. Whenever possible, try to implement a method that doesn't retrieve all columns of the table.

Additionally, although the getClob and getBlob methods allow the application to control how long data is retrieved in the application, you should realize that in many cases, the JDBC driver emulates these methods. The driver must retrieve all of the long data across the network before exposing the getClob and getBlob methods.

Sometimes you *must* retrieve long data. In this case, remember that most users don't want to see 100 KB, or more, of text on the screen.

REDUCE THE SIZE OF DATA RETRIEVED

To reduce network traffic and improve performance, you can reduce the size of any data being retrieved to a manageable limit by calling setMaxRows, setMaxFieldSize, and the driver-specific SetFetchSize. Another method of reducing the size of the data being retrieved is to decrease the column size. If the driver allows you to define the packet size, use the smallest packet size that will still meet your needs.

In addition, be careful to return only the rows you need. If you return five columns when you only need two, performance is decreased, especially if the unnecessary rows include long data.

CHOOSE THE RIGHT DATA TYPE

Retrieving and sending certain data types can be expensive. When you design a schema, select the data type that can be processed most efficiently. For example, integer data is processed faster than floating-point data. Floating-point data is defined according to internal database-specific formats, usually in a compressed format. The data must be decompressed and converted into a different format so it can be processed by the wire protocol.

Different DB2 data types take different amounts of time to process (see Table 1). Processing time is shortest for character strings, followed by integers, which usually require some conversion or byte ordering. Processing of floating-point data and timestamps is at least twice as slow as processing integers.

Updating Data in the DB2 Database

This section provides general guidelines to help you optimize system performance when updating data in databases.

USE UPDATEXXX METHODS

Although programmatic updates don't apply to all types of applications, developers should try to use programmatic updates and deletes. Using the updateXXX methods of the ResultSet object allows you to update data without building a complex SQL statement. Instead, you simply supply the column in the result set that's to be updated and the data that's to be changed. Then, before moving the cursor from the row in the result set, call the updateRow method to update the database as well.

In the following code fragment, the value of the Age column of the Resultset object rs is retrieved using the method getInt, and the method updateInt is used to update the column with an int value of 25. The method updateRow is called to update the row in the database that contains the modified value:

```
int n = rs.getInt("Age");
// n contains value of Age
```

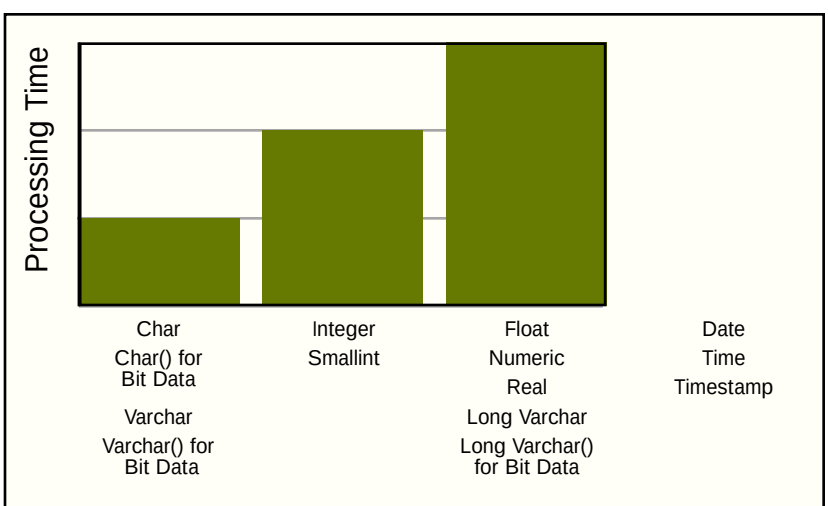


TABLE 1: Comparative processing time for various DB2 data types

```
column in the resultset rs
. . .
rs.updateInt("Age", 25);
rs.updateRow();
```

In addition to making the application more easily maintainable, programmatic updates usually improve performance. You don't need performance-expensive operations to locate the row to be changed, because the DB2 server is already positioned on the row for the Select statement in process.

Maximizing Metadata Methods

Because database metadata methods that generate Resultset objects are slow compared to other JDBC methods, using them too often can impair system performance. The guidelines in this section will help you to optimize DB2 system performance when selecting and using database metadata.

MINIMIZE THE USE OF DATABASE METADATA METHODS

Compared to other JDBC methods, database metadata methods that generate Resultset objects are relatively slow. So that you don't need multiple executions, applications should cache information returned from result sets that generate database metadata methods.

While it's almost impossible to write a JDBC application without using database metadata methods at all, you can improve system performance by using them as little as possible. To return all result column information mandated by the JDBC specification, a JDBC driver may have to perform complex queries or multiple queries to return the necessary result set for a single call to a database metadata method. These particular elements of the SQL language are performance-expensive.

Applications should cache information from database metadata methods. For example, call getTypeInfo once in the application and cache away the elements of the result set that your application depends on. It's unlikely that any application uses all elements of the result set generated by a database metadata method, so it shouldn't be difficult to maintain the cache of information.

AVOID SEARCH PATTERNS

Using null arguments or search patterns in database metadata methods generates time-consuming queries. In addition, network traffic potentially increases due to unwanted results. So always supply as many non-null arguments as possible to result sets that generate database metadata methods.

Because database metadata methods are slow, applications should invoke them as efficiently as possible. Many applications pass the fewest non-null arguments necessary for the function to return success. For example:

```
ResultSet WSrs =
WSc.getTables (null, null,
"WSTable", null);

should be :

ResultSet WSrs =
WSc.getTables ("null",
"johng", "WSTable",
"TABLE");
```

Sometimes little information is known about the object for which you're requesting information. Any information that the application can send to the driver when calling database metadata methods can result in improved performance and reliability.

DETERMINE TABLE CHARACTERISTICS WITH A DUMMY QUERY

Avoid using getColumns to determine characteristics about a table. Use a dummy query with getMetadata instead.

Let's consider an application that allows the user to choose the columns that will be selected (see Listings 1 and 2). Should the application use getColumns to return information about the columns to the user or instead prepare a dummy query and call getMetadata?

In both cases, a query is sent to the DB2 server, but in Listing 1 the query must be evaluated and form a result set that must be sent to the client. Because the Where clause of the query always evaluates to FALSE, the query generates no result rows and should execute without accessing table data. Clearly, the Listing 2 model will perform better.

Selecting JDBC Objects and Methods

The following guidelines will help you optimize system performance when selecting and using JDBC objects and methods.

USE PARAMETER MARKERS WITH STORED PROCEDURES

When calling stored procedures, instead of using literal arguments, you should always use parameter markers for the argument markers. JDBC drivers can call stored procedures on the DB2 server either by executing the procedure as any other SQL query, or by optimizing the execution by invoking a Remote Procedure Call (RPC) directly into the DB2 server. Executing the stored procedure as a SQL query results in the DB2 server parsing the statement, validating the argument types, and converting the arguments into the correct

"With thoughtful design and implementation, the performance of JDBC applications on DB2 can be improved"

DB2-ENABLED JDBC APPS

data types. Remember that SQL is always sent to the DB2 server as a character string; for example:

```
"{call getCustName (12345)}"
```

In this case, even though the application programmer might assume that the only argument to getCustName is an integer, the argument is actually passed inside a character string to the server. The DB2 server would parse the SQL query, isolate the single argument value 12345, then convert the string "12345" into an integer value.

By invoking an RPC inside the DB2 server, the overhead of using a SQL character string is avoided. The procedure is instead called by name only, with the argument values already encoded into their native data types.

CASE 1

Stored Procedure can't be optimized to use a server-side RPC. The DB2 server must parse the statement, validate the argument types, and convert the arguments into the correct data types:

```
CallableStatement cstmt =
conn.prepareCall ("call
getCustName (12345)");
ResultSet rs =
cstmt.executeQuery ();
```

CASE 2

Stored Procedure can be optimized to use a server-side RPC. Because the application calls the procedure by name and the argument values are already encoded, the load on the DB2 server is less:

```
CallableStatement cstmt –
conn.prepareCall ("Call
getCustName (?)");
cstmt.setLong (1,12345);
ResultSet rs =
cstmt.executeQuery();
```

USE PREPAREDSTATEMENT OBJECTS FOR REPEATED SQL STATEMENTS

JDBC drivers are optimized based on the perceived use of the functions being executed. Choose between the PreparedStatement object and the Statement object depending on the planned use. The Statement object is optimized for a single execution of a SQL statement. In contrast, the PreparedStatement object is optimized for SQL statements that will be executed two or more times.

The overhead for the initial execution of a PreparedStatement object is high, but the benefit comes with subsequent executions of the SQL statement.

CHOOSE THE RIGHT CURSOR

Choosing the appropriate type of cursor allows maximum application flexibility A forward-only cursor provides excellent performance for sequential reads of all of the rows in a table but it can't be used when the rows to be returned aren't sequential.

Insensitive cursors used by JDBC drivers are ideal for applications that require high levels of concurrency on the DB2 server and require the ability to scroll forwards and backwards through result sets. The first request to an insensitive cursor fetches all of the rows and stores them on the client. Thus, the first request is very slow, especially when long data is retrieved, but subsequent requests require no network traffic and are processed quickly. Because the first request is processed slowly, insensitive cursors shouldn't be used for a single request of one row. As a designer you should also avoid using insensitive cursors when long data is returned, because memory can be exhausted.

Conclusion

With thoughtful design and implementation, the performance of JDBC applications on DB2 can be improved. By using DatabaseMetaData methods appropriately, retrieving only required data, selecting functions that optimize performance, and managing connections and updates, your applications can run more efficiently and generate less network traffic on your WebSphere application server.

By following the tried-and-true approaches for JDBC in this article, you can develop and fine-tune your WebSphere applications to run faster, jump higher…and make fewer trips to the database.

LISTING 1: GETCOLUMNS METHOD

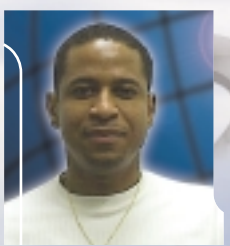
```
ResultSet WSrc = WSc.getColumns (... "UnknownTable" ...);
// This call to getColumns will generate a query to
// the system catalogs... possibly a join
// that must be prepared, executed, and produce
// a result set
. . .
WSrc.next();
string Cname = getString(4);
. . .
// user must retrieve N rows from the server
// N = # result columns of UnknownTable
// result column information has now been obtained
```

LISTING 2: GETMETADATA METHOD

```
// prepare dummy query
PreparedStatement WSps = WSc.prepareStatement
    (... "SELECT * from UnknownTable WHERE 1 = 0" ...);
// query is never executed on the server -
// only prepared
ResultSetMetaData WSsmd=wsps.getMetaData();
int numcols = WSrsmd.getColumnCount();
...
int ctype = WSrsmd.getColumnType(n)
...
// result column information has now been obtained
// Note: we also know the column ordering within the
// table!  This information cannot be
// assumed from the getColumns example.
```

Database Programming with Version 4.0

BY ANTONIO VILLAFANA



HOW TO MIGRATE TO SUPPORTED DATABASE CONNECTION APIs USING CONNECTION POOLING AND DATA ACCESS BEANS

The rich set of new features and functionalities in WebSphere Application Server 4.0 reduces some of the complexity usually inherent with database programming, especially with distributed database programming.

ABOUT THE AUTHOR

Antonio Villafana is an independent consultant with e-Mind Solutions. Previously a senior technical architect at Management Information Consulting (MIC), Antonio has over 12 years of experience in the computing industry and has worked with several leading technologies such as messaging services, XML/XSL transformations, J2EE, workload management and load balancing, application server implementation and monitoring, and RDMS development.

E-MAIL

pcclinic@mediaone.net

Before any application component attempts to retrieve data from a database it must first acquire a valid connection to the database. Most seasoned developers are well aware of the overhead incurred with getting a connection to a database. Not only must the application server set up a connection to the database, it must also maintain the open connection and perform any housecleaning operations that must be done prior to releasing the connection and freeing up system resources again.

The prospect of performing these expensive operations each time a user request is submitted to the application is indecently unacceptable. Resources would be quickly drained and expensive hardware storage solutions would have to be implemented so applications can perform reliably and scale with user growth. With connection pooling the overhead of creating and releasing a connection is dramatically reduced since connections are now cached and reused by the application.

Connection Pooling

Web-based and standalone desktop applications can now benefit from this useful concept that's been around for some time. Connection pooling mechanisms became more sophisticated with the introduction of improved middleware components such as Java servlets and Microsoft's ASP technology. With improving middleware performance, RDMS and application servers needed a way to keep abreast of new customer expectations and application scalability.

IBM first introduced a connection pooling mechanism and datasource objects, in WebSphere Application Server version 3.02x. The nonstandard method used to create datasources and obtain connections to a database in versions 3.02x looked like this:

```
import
com.ibm.db2.jdbc.app.stdext.javax.sql.*;
```

This must be replaced with:

```
import javax.sql.*;
```

in versions 3.5.x and later.

WebSphere's connection pooling provides a layer of abstraction from databases that removes some of the burden of rewriting applications to trap vendor-specific exceptions whenever a database vendor change has occurred, thereby making your code more maintainable and portable.

Connection pooling provides several key benefits to two- and three-tier applications that access relational databases using JDBC 2.0 or later:

- ***Improved application performance:*** Due to the reduced overhead of creating and destroying database connections. (Getting a new connection to a database is a very expensive operation.)
- ***Improved resource management:*** Resources are no longer consumed unnecessarily, and are available to other application server components such as caching mechanisms and concurrent service requests.
- ***Less code:*** Java classes and methods no longer need a separate mechanism for creating, obtaining, and managing database connections to relational back-end systems.

Migrating to Version 4.0

Any database access code that uses WebSphere's datasources must be modified to use the standard packages, including all J2EE objects such as servlets, session beans, and entity beans. The packages:

```
com.ibm.db2.jdbc.app.stdext.javax.sql
```

and

```
com.ibm.ejs.dbm.jdbcext
```

have been deprecated as of version 4.0. Existing applications can use already created datasources, however – but they can't create new datasource objects, since new datasource objects created using the deprecated packages can't be bound into JNDI in version 4.0. In version 4.0 all new datasources must be created using the com.ibm.websphere.advanced.cm.factory.DataSourceFactory.

WebSphere versions 3.0x ran on the JDK 1.1 level, and IBM provided the necessary transaction support features now found in the JDK 1.2.x level and later JDKs. In version 4.0 the proprietary transaction support provided by IBM is in the following package:

```
com.ibm.db2.jdbc.app.jta.javax.transac
tion.*;
```

Which must be replaced with the standard package:

```
javax.transaction.*;
```

After changing the above import statements, Java sources should be recompiled using the new JDK 1.3 package.

Before migrating your applications to the new supported transaction API, as a WebSphere developer you should be aware of several important transaction-management principles, one of which includes database connection usage and availability. If an application component obtains a connection to a database and then executes a transaction on that connection, once the transaction is flagged as completed by the database server and the transaction manager component of the transaction API, the connection is closed automatically and returned to the connection pool. A somewhat smaller and obviously overlooked setting is the transaction timeout setting. If you're involved with fine-tuning WebSphere, then this little tidbit should be of some concern to you, since improper numbers will degrade performance and scalability.

DEFAULT USERNAME AND PASSWORD

Previous versions of WebSphere's connection pooling didn't allow for default username and password properties, which meant that developers had to figure out the best method of storing this sensitive information without complicating application architecture and compromising database server security. However, in Version 4.0 connection pooling datasource object now allows developers to use the default username and password set by the administrator, using one of WebSphere's administrative tools.

Application Development

WebSphere has implemented the J2EE 1.2-compliant method of performing lookups. The syntax used by both versions 3.5 and 4.0 will work, but all new applications should be written using Version 4.0 syntax.

One of these changes involves the initial context lookup method:

- Version 3.5x
  `com.ibm.ejs.ns.jndi.CNInitialContextFactory`
  must be changed to Version 4.0
  `com.ibm.websphere.naming.WsInitialContextFactory`

Another change involves the string argument for the ctx.lookup() method:

- Version 3.5x

```
javax.sql.DataSource ds =
(javax.sql.DataSource)
ctx.lookup("jdbc/SampleDB");
```

must be changed to Version 4.0

```
javax.sql.DataSource ds =
(javax.sql.DataSource)
ctx.lookup("java:comp/env/jdbc/SampleDB");
```

Datasources and Connection Pooling

Datasources are normally created by a system administrator using one of WebSphere's GUI-based admin consoles. To obtain a datasource from the factory and then bind the datasource into JNDI is a relatively simple process. Depending on your maintainability requirements you may decide to place the username, password, and datasource name in a plaintext properties file or, if using the HttpServlet interface, retrieve them using the getInitParameters() method. Listing 1 (go to www.sys-con.com/websphere/sourcec.cfm for complete listings) illustrates the retrieval and JNDI binding process:

One property that's been introduced on the com.ibm.websphere.advanced.cm.factory.DataFactory, is the DISABLE_AUTO_CONN_CLEANUP property. This new property setting specifies whether or not the connection should be automatically released and returned to the connection pool at the end of each transaction (see Figure 1). The default value is false, which indicates that WebSphere's connection pooling should automatically close a connection at the end of a transaction – thus releasing it back to the pool (see Figure 1).

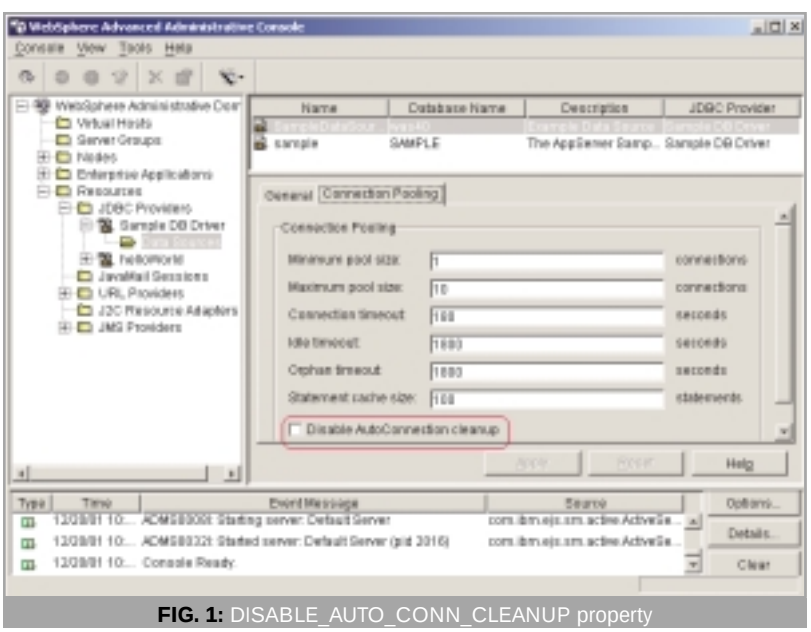


FIG. 1: DISABLE_AUTO_CONN_CLEANUP property

STATEMENT CACHE FEATURE

I often used the "roommate from hell" story to get junior developers to fully understand the purpose of performing trivial, but very important, housekeeping routines such as closing resultsets, statements, and connections. Imagine living in an apartment with someone who loves to eat, but never cleans the kitchen: after a week, the energy and time required to restore the kitchen back to its once-pristine state is very resource-intensive. The same concept applies to database programming. The rule's simple: if you're the one who opened it – then close it.

One of the really outstanding features that can certainly improve your application's performance is WebSphere's statement cache feature. WebSphere has provided a caching mechanism that caches prepared statements, thus allowing for reuse. With statement caching the overhead of creating a new PreparedStatement each time one is needed is now noticeably reduced, thus improving performance by reducing the overall response times of the application.

WEBSPHERE EXCEPTIONS

In version 3.02x the ConnectionPreemptionException was introduced, but no longer exists in version 3.5.x. ConnectionPreemptionException, in all cases, has been replaced with StaleConnectionException in versions 3.5.2 or later. com.ibm.ejs.cm.portability.StaleConnectionException has been deprecated in version 4.0, but applications using this class will still function as expected. As expected, IBM recommends writing new applications using the new com.ibm.websphere.ce.cm.StaleConnectionException class.

Connections that are no longer consistent because of momentarily disabled database servers or network failures are called stale connections. Other factors that can cause a connection to be invalid include:

1. Trying to use an orphaned connection.
2. Trying to use a connection on which close() was already called.

When this occurs, all connections in the pool must be flushed and recreated. In version 4.0 the StaleConnectionException detection and recovery mechanism has been improved. StaleConnectionException extends the ever-popular SQLException, so you may be asking "why should I even bother trapping this exception?" By explicitly handling the StaleConnectionException, you can now perform a specified number of retries, thus giving the application a chance to recover from a noncritical network or database failure.

Some SQLExceptions are now mapped to specific WebSphere exceptions. WebSphere's connection pooling mechanism monitors these exceptions, thus making it easier for developers to trap and recover from these exceptions. In version 4.0 more vendor-specific codes are mapped in the abstraction layer, thus enabling better support and recovery from noncritical database errors. Database programmers no longer have to know each data-

base-specific exception and can now write easily adaptable code without worrying about database specific SQLExceptions.

Listing 2 shows one method of handling a StaleConnectionException. If a request for a connection throws a StaleConnectionException, the application makes three additional attempts at getting a new database connection. If all attempts fail then close() is called on the java.sql.statement and java.sql.connection objects, and the application responds with a critical error.

A ConnectionWaitTimeoutException indicates that the application has requested a connection from the connection pool but that no connection has been issued in the allotted time. The allotted time is the number of seconds specified in the CONN_TIMEOUT setting. The following code sample shows one acceptable way to handle this exception in WebSphere. One of the biggest real-world causes of ConnectionWaitTimeoutException is incorrect username and/or passwords (did you read all your e-mails today). System administrators usually sets this parameter according to the application workload requirements and/or expectations.

```
…
java.sql.Connection conn = null;
javax.sql.DataSource ds = null;
…
try {
  …
  // get initial context properties here
and create initial naming context.
  …
  ds = (javax.sql.DataSource)
ctx.lookup("java:comp/env/jdbc/sample");
  conn = ds.getConnection();
} catch
(com.ibm.ejs.cm.pool.ConnectionWaitTimeoutEx
ception cwte) {
  // perform recovery operations – there's
no reason to retry getting a connection
here.
} catch (java.sql.SQLException sqle) {
  // let the world know!
}
```

Improving Application Performance

Improving application performance benefits not only the user of the system but also the creators of the system. Since developers are by nature problem solvers at some level, their constant desire to improve performance and solve other related issues such as quicker development turnaround time is quite understandable.

JNDI LOOKUPS

One very important aspect of improving J2EE application performance is the caching of JNDI lookups. Imagine what it would be like if nobody remembered your name, or worse yet, who you are – they'd be forced to ask you the same questions each time you met. Well, the same analogy applies to the process and reasoning for caching (remembering) JNDI lookups. By implementing caching of JNDI lookups you'll reduce the response times of your application(s). JNDI lookups are expensive operations, and should only be done once by the application. As a best practice lookups should be done in the servlet's init() method or, if using EJBs, from the ejbActivate() method.

FREEING UP RESOURCES

A very common mistake in database programming is waiting until the finalize{} block to close() opened resultsets, statements, and connections. The assumption here is that the resources on which the close() method is called are actually freed, and are available for reuse. This isn't the case, though: resources are merely flagged for cleanup by the JVM, and until this happens valuable resources are still being held by the application. The result is an application that performs well under normal conditions, but won't scale and is very difficult to debug. As a rule of thumb, database resources should be freed as soon as they are no longer needed.

DATA ACCESS BEANS

WebSphere programmers can benefit a great deal by using data access beans provided by IBM. These data access beans provide additional functionalities and features beyond the benefits of connection pooling. Websphere's data access beans provide a tight coupling between its connection pooling mechanism and standard JDBC best practices. To use these data access beans in your code and appreciate the advantage of doing so over doing it yourself, you must first incorporate some changes to your code.

Any application component that will use WebSphere's data access beans must:

1. Import the com.ibm.db package. The package can be found in "databeans.jar" located in the <WAS_HOME>\lib folder. As with any other package, it must be made available to your servlets, preferably in your application server classpath.
2. For servlets, declare a class-level variable for the "StatementMetaData" class that will be fully initialized once the init() method has completed. All subsequent requests serviced by the servlet will be using the same instance.

Listing 3 shows a sample implementation of some the features and portability you can obtain by using IBM's data access beans.

The init() method of the DBServlet performs a one-time initialization of the StatementMetaData (mData) object and the InitialContext lookup. The username and password credentials required to access the database, along with a datasource name, are set as init parameters and retrieved using the getInitParameter() method of the ServletConfig. A DataSource object is created using the datasource name provided to the InitialContext lookup method. This datasource object will now be available to service all subsequent requests to the servlet, without performing the expensive operation of looking up a DataSource object.

Now let's discuss some of the operations performed by the servlet's setStatementMetaData() method and by the performTask()(service) method, which makes it possible to access our data using IBM's data access beans. In the setStatementMetaData() method, a new Statement MetaData (mData) object is created. Several methods on

"A somewhat smaller and obviously overlooked setting is the transaction timeout setting. If you're involved with fine-tuning WebSphere then this little tidbit should be of some concern to you"

the mData object are called to set the database metadata information. The addParameter() method allows you to delegate the task of converting between Java datatypes and the underlying relational datatype to the mData object. Notice that the argument list of the addParameter() method includes both the regular Java datatype (Integer.class) and the relational datatype (Types.SMALLINT). Once created, the mData object becomes responsible for automatically converting the servlet's Java datatype so as to make it consistent with relational datatypes. Because our servlet will be updating the relational table, we must also include the addColumn() method, which includes the same argument list as the addParameter() method. The addTable() method is also necessary for updating the relational table.

A connection object (conn) is created for every user request being serviced by the servlet, which allows for multiple concurrent users. The link between the database access bean and the actual database connection is the DatabaseConnection (dbConn) bean object, which provides an expanded set of functionalities and features. The SelectStatement (selectStatement) and SelectResult (result) objects act as an abstraction layer between the already established mData and the user data passed to the servlet. The selectStatement object is associated with a connection object, the mData object, and any parameters required by the underlying SQL statement.

After calling the execute() method on the selectStatement object, the result of the query is captured in a SelectResult (result) object. Once the result object is created, the connection object associated with the selectStatement can be returned to the connection pool by calling the dbConn disconnect() method. This is possible since the result object caches the query results one layer closer to client, thus making it available to the rest of the application, without the application holding on to much-needed system resources.

Testing the DataAccessBean (DBServlet)

To test the DBServlet sample in WebSphere 4.0 you will need, if you haven't already done so, to complete the database configuration steps for using WebSphere sample applications. To access the configuration instructions, open your browser to http://localhost/WSsamples/index.html. The DBServlet locates all the employees for the specified department (dept) and, for demonstration purposes only, updates/increments the value of the employee's salary by the amount specified, for the first employee in the list only. To add the servlet to WebSphere's default_app application you'll need to perform the following steps:

1. Ensure that application server reloading is enabled. Otherwise the server will have to be restarted.
2. Open the dbSample.jar file and extract the DBServlet.class file to <WAS_HOME>\installedApps\Sample_Application.ear\default_app.war\WEB-INF\classes.
3. Add the following to the <WAS_HOME>\installedApps\Sample_Application.ear\default_app.war\WEB-INF\web.xml file, making sure that the five attributes (ids) highlighted in blue are unique within the current web.xml file:

```
<servlet id="Servlet_1">
      <servlet-name>DBServlet</servlet-name>
      <display-name>DBServlet</display-name>
      <servlet-class>com.mic.sample.DBServlet
      </servlet-class>
      <init-param id="InitParam_1">
       <param-name>username</param-name>
       <param-value>wsdemo</param-value>
      </init-param>
      <init-param id="InitParam_2">
       <param-name>password</param-name>
       <param-value>wsdemo1</param-value>
      </init-param>
      <init-param id="InitParam_3">
       <param-name>datasource</param-name>
       <param-value>jdbc/sample</param-
        value>
      </init-param>
      <load-on-startup>1</load-on-startup>
    </servlet>
    …
    <servlet-mapping id="ServletMapping_1">
      <servlet-name>DBServlet</servlet-name>
      <url-pattern>/dbSample</url-pattern>
    </servlet-mapping>
```

4. Open your browser to http://localhost/servlet/dbSample?dept=E01&amount=12345.

Try changing the value of "amount" in the query string, and you'll be able to see the salary increment each time the page is submitted.

Summary

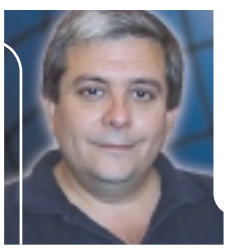
IBM's WebSphere Application Server provides an abstraction layer for better data exception handling and data access beans for working with relational data. These new features reduce some of the complexity usually inherent with database programming, especially with distributed database programming. WebSphere provides a rich set of features and functionalities that most development teams aren't familiar with or simply ignore. So, don't be a statistic – maximize your resources!

Communicating the fundamentals to the customer

The Selling of WebSphere

BY JIM MARTIN

Welcome to the inaugural column of "Why They Buy." In this space we'll explore the reasons behind why customers choose WebSphere and why WebSphere sales are made. Making sure that the customer is part of the solution team ensures effective team selling and WebSphere sales. It's not as hard to achieve as it sounds. Hopefully, this column will help you do that.



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 for the financial industry. Jim currently works for Nova Corp. in New Jersey.

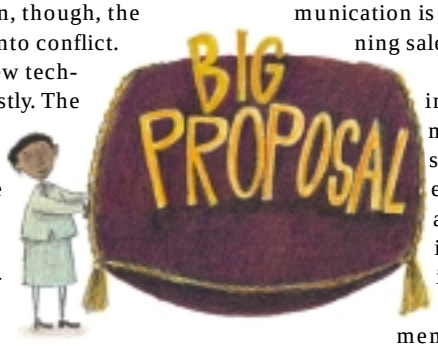
E-MAIL
jim@skc.com

We'll look at the role of the technical members of the sales teams, hear from sales representatives from IBM and its business partners, and along the way try to provide helpful tips. But most of all we're going to try to give you a competitive advantage in selling WebSphere.

In this first installment we'll look at an overview of some of the basics of selling and assisting a customer in choosing WebSphere. In upcoming columns we'll look at real-world examples in the form of successful WebSphere sales, and speak with the customers and sales teams to better understand how they work together to create winning solutions. We'll hear from customers' first hand on how they view the WebSphere solution and how they measure and realize ROI.

Begin At The Beginning

In this case, the beginning is the fundamental question: why does a customer want to improve their infrastructure? The simple answer is because they want to improve their business, to service their customers better, so the business makes a greater profit. How do they do this? They control costs to be competitive. They're responsive to their customers. They're able to adapt to the ever-changing marketplace. How do they do this? They adopt technology that helps them better share information within their own organization and with partners and with suppliers, making the business process more efficient. More efficient usually equals more profit. Often, though, the two goals come into conflict. Implementing new technology can be costly. The decision affects the entire enterprise, not just the IT department. Potential customers must participate in planning, cost analysis, and implementation, bear the hard dollar cost of the technology, plan for support, contend with some business disruptions, and plan for employee retraining.



So how do we bridge the gap between running a cost-efficient business and the cost of new technology? First, we need to make sure whatever new technology is selected works for the whole enterprise, not just for stovepipe operations. Second, it has to be scalable so, as the business grows, the same technology will continue to be the right answer. Third, it must create a better customer experience or business growth will stall, defeating the original purpose of the business improvement. And don't forget the IT holy grail question: will the technology be able to support today's business environment as well as tomorrow's? WebSphere answers all of these questions and more when the sales team and the customer have a shared vision of their business goals.

The Front Line

Before any sale can get to the win/win stage the sales team and the customer first need to get to the yes/yes stage. What's that? Yes/yes is when the sales team has a clear understanding of the customer's business goals and plans, while at the same time the customer understands the dimensions and functionality of the proposed solution. So if anyone asks, "Do you understand that?", whether the question comes from the customer or the sales team, the answer is, "Yes, I do." Effective communication is the key to a winning sales solution.

It's always vitally important that the members of the sales team be experts in their areas of responsibility to the team, but it's equally important that each member be expert in

acquiring and understanding the customer's business goals. It may seem obvious, but very often we human beings miss seeing what's right in front of us. This sometimes happens in the sales process and that's when the good deals sometimes slip away. Selling as a team works because each member brings strength to the overall effort. They also bring a unique way of seeing a situation because of the experiences they've had. The better the sales team interacts together, the more useful an understanding of the customer you'll have. The benefit to the team is realized both early on and later in the sales cycle.

What It Isn't

However, before we get to how to better understand a customer, let's make sure we know what's on our side of the equation. What's WebSphere? What can it do for a customer? How will they benefit? Why should a customer choose WebSphere instead of a competing platform? It's very important to understand what WebSphere is not.

- WebSphere isn't yesterday's or experimental technology.
- WebSphere isn't proprietary technology with limited applications or toolsets.
- WebSphere isn't cheesy.

What WebSphere Is

It's even more important to understand what WebSphere is.

- WebSphere is an open environment that allows applications to be leveraged by users regardless of their locality or access device.
- WebSphere is a foundation for building and deploying business processes.
- WebSphere is a launching pad to deliver personalized user experiences even when scaling to enterprise levels.

Let's look at what WebSphere is and how the platform fits into the customer's plans to improve their business processes. Today, many business processes are migrating to the Web because of the tremendous benefits that come from its ubiquitous access and distribution model. The same foundation technologies need to extend themselves to private networks, extra/intranets and the wireless world. This is the e-business transformation.

The e-business transformation is a process that has helped propel many companies to 21st-century success. Both for new economy companies and traditional businesses that have been around for a while, WebSphere is a launching pad for the e-business transformation. E-business is more than just horoscopes or selling stuff on the Web. It's a method of leveraging technology to create streamlined business processes to improve business cycles. It's the ability for the enterprise to reach out to its customers and suppliers. It's also the ability for customers and suppliers to be incorporated into the enterprise. In a nutshell, WebSphere is the vehicle for merging the back office and front office together to create one straight-through process.

WebSphere's use of open standards provides the platform to create applications and tools that are independent of the user's method of access. By embracing standards such as SOAP, UDDI, WSDL, XML, and J2EE 1.2, WebSphere becomes an integral part of almost every e-business technology solution. These accepted standards are the foundation for Web services, which offer reliable, e-business processes that can be accessed anywhere, anytime, every time, by everyone. Pretty neat stuff when you stop and think about it.

The ability to leverage open standards enables rapid application development and will later minimize support issues. Whether your customer or their customers or suppliers choose to develop applications in-house or through an application provider, the foundation technologies are the same. There are no compatibility or integration issues. My standard is your standard, which is also someone else's standard. It almost seems like IT world peace, scary. Open standards also bring a reduction in the costs for the development and support of future initiatives because there will be a deep talent pool to be tapped. It's very unlikely that there will be a shortage of people with the skills in these areas. A WebSphere customer won't be locked into a proprietary arrangement where the business is limited by the availability of skilled people. The global e-business transformation is already under way.

Web Services

Web services make it easy for the infrastructure to be extended within the customer organization and out to the world of their customers and suppliers. This is the e-business model at it finest. This practice retains the technology and skill sets that are most likely in place today, while opening the door to the applications and methods of tomorrow. Most organizations have by now started down the open-standards road in one way or another; WebSphere will get them to their destination. There are no ties to applications or tools that can disappear or lose popular support. However, there's the ability to leverage resources already within the enterprise. This is the first benefit of the transformation to an e-business. Web services based on open standards are what make WebSphere the preeminent environment for positioning an e business to be competitive during today's fast-changing business cycles.

As the range of access devices proliferates, so will the nature of the applications deployed to them. This flexibility makes the WebSphere infrastructure not only the machine that keeps the business humming but also the one that transforms it into an offensive business tool, allowing your customer to react to business conditions more effectively and quickly. Open standards eliminate the burden of tailoring applications to physical devices, instead linking them to what they should be linked to in the first place: users.

Making the Decision

Why do we need to understand and communicate these fundamentals to a customer? Shouldn't they know this already? Perhaps they do, but always remember that we each have a unique view of the world. We

need to make sure that our customers' views and ours come from the same perspective. No matter how effective the solution we're selling, the customer's decision will be based on a relatively small number of decision points that each carry tremendous weight. The following is a simplified list, but it offers the foundation points of the decision process.

1. Will it improve and grow the business?
2. Can the solution be implemented without significant disruption to the business?
3. Is there a measurable return on investment?

We've already explored some of the factors that address the first two points; now let's look at number three, ROI. This is the key to selling any technology solution in today's marketplace. It's a numbers game in the end, where you have to prove value. The days of implementing cutting-edge technology and then figuring out what to do with it are long past. Today we need to demonstrate that new technology investments will directly benefit the business by quickly recouping the initial investment and the after effect will be a long-term foundation to further grow the business.

The value proposition is very straightforward: the cost of maintaining the existing infrastructure versus the increased benefits of implementing WebSphere. This is where having a clear understanding of the customer becomes invaluable. It's also when the shared vision with the customer is put to the test. The cost of the current environment minus the WebSphere environment should equal increased benefit. Hey, we're all grown-ups here; go ahead and think profit instead of benefit.

If, during the sales process, the sales team has had meaningful and effective communication with the customer, then there should be an established set of measurable standards to determine the cost of the current infrastructure. There should also be a measurable set of benefits that the WebSphere solution will bring to the customer. These may be a reduction in operating costs for order processing or increased sales due to having a broader or better marketplace. It's here that a price is put on the customer's vision. If during the fact-gathering process the sales team did a good job understanding the customer's business goals, this part of the selling and closing process becomes much easier. This is why the customer should be part of the WebSphere solution team. They need to see the situation from the same perspective and through the same lens that the sales team sees the situation. If this happens, then the sale should begin to close by itself. (It's not that easy; you'll still have to work at it – sorry to lure you in like that.) Some of this may seem fundamental, but it's the little things that may separate you from success. Big things will knock you out early in the game and you will move on to the next opportunity. That's the way things are. It's having the understanding of the details and your ability to address them that will separate you from your competitors.

Closing the Deal

Very often, and unfortunately, surprises come from both sides during the closing process. The best way to prevent this is to be attentive during the early sales processes. Understand what's driving the move to a new way of doing business. Understand the decision-making process of your customer. Establish lines of communication at all levels of the organization. Put yourself in your customer's shoes. Try to see what they see. This keeps the sales team from presenting a less-than-complete or inaccurate solution while it ensures that the customer won't introduce eleventh-hour thoughts that can extend the sales cycle. The key is to truly make the customer part of the solutions team. Remember the old saying, "Many hands make for light work."

When properly engaged, the customer should internally create almost the same ROI model that the sales team creates. Why? You're all working from the same set of measurable facts. There are no "Well, the way I see it…." conversations. Those conversations can extend, or even worse, derail the sales process. A common understanding will also foster customer loyalty long term, which will mean more sales later on as they continue to leverage the WebSphere platform. In the end you're pursuing the same goals as your customer but in different ways and for different reasons. You're both searching for a better way to do business, both looking to discover a more efficient way of doing business, and in the end you both want to increase your productivity – and make a greater profit.

How do we accomplish this lofty goal? Pay attention to each stage of the sales process and be part of it. Listen to your customers. Listen to your teammates. Teams win, individuals just work very hard. Some of this may sound trite or rudimentary, but we overlook simple things every day that affect our business. We're only human; we'll make mistakes or forget something, which is why we work as a team. We support one another to make things work. That's why we should never forget to include our customers on our team. Many hands will make for light, and smart, work.

"Pay attention to each stage of the sales process and be part of it. Listen to your customers. Listen to your teammates. Teams win, individuals just work very hard."

THE NEW MATH: APPLICATION SURVIVABILITY + OPERATIONAL READINESS = BUSINESS CONTINUITY

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. He is currently working as an independent consultant in the New York metropolitan area.

E-MAIL

jjf10965@yahoo.com

Business continuity. A new and exciting catchphrase? For some, perhaps. Traditional business continuity planning involves many aspects of corporate activities, from call center rerouting and alternative raw material suppliers to policies requiring executives to fly on separate planes.

BY JOE FARSETTA

ACHIEVING THE 99.99% AVAILABILITY THRESHOLD

For the IT professional, business continuity usually equates to offsite data storage, along with contracting for data center space with recovery sites such as Sunguard. For the seasoned IT professional, it has long been the somewhat elusive prize at the end of the day. It's the delicate balance between data center heartiness and capital expenditure, between overkill and practicality, and between business needs and shrinking IT budgets. In the end, it's an exercise in risk assessment. And, as with any insurance policy, the more protection you want, the more it's going to cost.

As many businesses discovered after recent events, the data center space they thought they were guaranteed in the event of a disaster may not always be available, especially if the event has affected a number of businesses and your disaster recovery partner is oversubscribed. So, what steps can one reasonably take to help ensure IT business continuity and application survivability? The exercise requires detailed analysis. It may ultimately require dual data centers taking geographically dispersed and load-balanced traffic. As such, this article will focus on business continuity from the standpoint of bulletproofing your network and server farms to help ensure application survivability. Although I point out some WebSphere product suites that offer some added functionality in a WebSphere environment, most of the principles revealed here are truly universal and pertain to most high-availability scenarios.

Basic Concepts

Business continuity, from the IT side at least, is as much of an exercise in operational planning and readiness as it is in engineering and application development. Operational criteria is the driving force behind the need for fault tolerance and survivability in the first place. The establishment of service-level objectives is a key concept early in the planning phases. Service-level objectives will establish system availability criteria. Availability, the benchmarking threshold by which varying degrees of engineered fault tolerance are applied to the overall design, is the long-range goal. Availability thresholds will set the stage for policies and procedures addressing quality assurance, testing, application release methodologies, change control, monitoring, alerts, remote support, and, of course, the degree of fault tolerance required from the application, server, and network levels.

Availability, as a concept, differs somewhat from the notion of "downtime." Availability assures that the system, or application, is functioning and able to process desired transactions. Individual servers, network segments, or storage devices may indeed be down, without affecting the overall system or application to the point where transactions have stopped. For instance: a single server within a high-availability server cluster may have "tuned itself for maximum smoke." (I like that phrase.) That is, it's gone, finished, light a candle – it's dead. Although this server is indeed down, the application survives within the remaining clustered servers. The server failure is indeed catastrophic, but the damage is limited to hardware, rather than system or application availability. Availability is unaffected and the system survives. This assumes, of course, that the failure was properly monitored, spare parts are available to replace the damaged component in a timely fashion, everything is documented, and that a well-trained operations staff is on top of the situation before the next failure occurs.

So, what should the targeted availability threshold be? Well, that depends on the nature of your, or the client's, business. For instance, if your business truly relies upon continuous system availability, including off hours, be prepared for a 99.99% availability threshold. This equates to approximately four minutes of actual allowable system unavailability per month. Impossible, you say? Not really. I, myself, was entrusted with bringing a large financial institution onto the Internet and delivering a guaranteed 99.99% site availability SLA. In the end, we ended up actually delivering 99.999%. So don't think it can't be done. That client's system is still running, two years later. And, if your business relies on the system or application as its lifeblood, then we are talking about that all-important concept: business continuity.

Planning, Planning, Planning

Let's say that your service level objectives call for high system availability as the end goal. Management has determined that this lofty criteria is cost-justifiable, without running the numbers. You've got the green light, and are now expected to produce a near bulletproof solution. Let's begin the planning phase…

ANTICIPATED TRAFFIC

Sometimes a bit difficult to really nail down, the formula for determining anticipated traffic patterns becomes a cross between standard accounting practices, wishful thinking, and magic. In the end, though, you need to really try for an accurate projection, then add some headroom just in case. Anticipated traffic will drive network and server design. The higher the transaction rate, the more network bandwidth and processing power required. The more processing and storage required, the larger the equipment footprint. This leads to even more dependencies, so you can see the importance of this first step.

KNOW YOUR APPLICATION

Familiarity with the application is key to peak performance, scalability, and supportability. For instance, knowing the size of a typical transaction will help determine network throughput requirements. Benchmarking a typical user transaction is also key in managing server and network performance, as well as correctly setting expectations as to how an application will react and respond to the average user during peak and nonpeak periods. Familiarity with your application will also help to determine which operating system and server platforms it runs on best. This dovetails into selecting clustering software, and all that's

entailed in that endeavor. For example, WebSphere Everyplace Server has some robust capabilities for this type of high-availability environment, including fault-tolerance, load balancing at the cluster-level, and caching. The product also runs on a variety of OS platforms, including AIX and Solaris

OPERATING SYSTEMS

Key, key, key. Decide which OS you'll use, choose the most reliable and robust high-availability clustering solution, and go with it. Know how your app will run in this OS environment. WebSphere Application Server has cluster software bundled within the application suite, while a database cluster may require separate software to handle the cluster function. Also, find out if the OS supports dual network interface cards (NICs) or single NICs with dual interfaces. This is important for the purposes of dual network pathing. Spend some time and effort on this decision. It's that important.

STORAGE REQUIREMENTS

Beyond storage capacity, logistical characteristics are key when selecting a storage solution. For instance, will you be using high-capacity arrays in a SAN configuration? How far from the server farm will these arrays be? Knowing this may ultimately require you to purchase and install fiber cards and switches to link the servers to the storage boxes if distance limitations are exceeded. What about backup and restoral requirements and procedures? Will off-site storage be required? Will tape robots be deployed? How often will incremental and full backups be performed? What's the expected restoral time if lost data needs to be reloaded? These are all critical factors that must be carefully thought out.

SERVER HARDWARE

As your understanding of what you are trying to achieve begins to gel, and all the interdependencies are identified, server hardware requirements come into focus. The first thing you must determine is whether you plan to use legacy platforms or purchase new ones. Remember the application benchmarking exercise I mentioned a few paragraphs back? This is where your analysis will determine if expected application performance criteria can be achieved on the processing platform you plan to use. Whatever the case, it is of paramount importance that you benchmark application performance early on. I remember being part of one of the first real SAP R3 rollouts for an enterprise-class client. We actually benchmarked transaction size and true response time from the user experience. Based on that data, I was able to engineer the LAN, MAN, and WAN infrastructures to ensure expected application performance in peak and nonpeak traffic periods. Again, this is about managing expectations and building an information base for future troubleshooting

Certain network infrastructure requirements may influence server hardware choices. Multiple processors within the server framework are typically required in a high-availability design. RAID arrays are typically included in base server design. SAN configuration and the physical proximately on the mass storage units to the server may influence connection methodologies. If you are connecting the server to the storage unit via optical fiber, dual fiber cards (for redundant links) are also a good idea.

SERVER CONNECTIVITY

Dual NICs at the server level are a must. Separate interface cards are the optimal choice, but carry their own unique set of problems. Two interfaces on a single card are also acceptable and may ultimately be your only viable option. The main idea is to provide dual pathing of network traffic to the cluster servers (yes, I mean the individual servers within the cluster) in the event a network switch fails. Dual-pathing to all servers in the infrastructure is a requirement. Application behavior and supportability of a dual-NIC scenario is an important component of your design due diligence, so don't underestimate its importance.

SERVER CLUSTERING

An absolute requirement for most high-availability server infrastructures, clustering helps ensure application survivability by load balancing across multiple servers within a cluster, or by pooling cluster resources. Depending on the application and platform of choice, various clustering solutions are available. For instance, it's critical to understand how failover will occur, and how long the process will take. Will the cluster software support an active-active configuration? Will the cluster software configuration delay application restoral for more than five minutes? If your service-level objectives call for 99.99% application availability and this is a two-server cluster, you have just blown your number. So, you'd better be sure that your solution can meet the expected performance criteria.

The function of the servers to be clustered is also an important consideration. Certain applications are "cluster-aware." For example, WebSphere Application Server successfully manages both load balancing within the cluster, as well as failover functionality. In database clusters, things are somewhat different. In this scenario it's likely that some clustering software will work very closely with the server OS, and typically be transparent to the application. So, knowing your cluster is an important consideration. Most server manufacturers have their own version of cluster, or high-availability, software. Some software houses, such as Veritas, also have offerings. Regardless, it's also critical to be sure and have the cluster certified by the software manufacturer before putting it into a production state. This helps eliminate any finger-pointing later on.

BASE APPLICATION FUNCTIONALITY

Remember to ensure that your application will function correctly (or at all) across multiple processors in a single box, across multiple servers, and with storage in mind. If you plan to use some form of disk shadowing or mirroring, be sure that the app supports it. QA is really important; so is documentation. How the overall data flow works within the application and across the platform is key. Building, understanding, and documenting relevant and usable bug codes is also essential. If the application malfunctions, or merely hiccups, flag the user or monitoring system with a noncryptic bug code. This will go a long way toward supporting the application in the future.

NETWORK DESIGN

What good is having multiple servers and sophisticated storage solutions if your infrastructure relies on a single communications link between boxes, and that link goes away? It's kind of pathetic to imagine yourself standing in the data center staring at a dead application, lots of blinking red lights, and your sole Ethernet switch sitting dark and silent. As you hear your boss screaming from down the hall, you wonder, "How long will it take to update my resume?" All kidding aside, dual network paths are a prerequisite for any high-availability framework. This dual-pathing follows throughout the network design, including NICs, switches, routers, and firewalls

NETWORK LOAD BALANCING WITHIN THE DATA CENTER

A critical concept, network load balancing achieves several things. Aside from the obvious, load balancers (dual of course) also allow servers to be seamlessly rolled in and out of production. This is especially useful if you have servers in a hot-standby mode. Now, let's imagine your new application is geared toward some retail sales function. Seasonal traffic may crush all available processing power. Wouldn't it be nice to place your hot-standby machines into production by simply adding them to the group that the load balancers distribute traffic to? This is truly production on demand!

As an example, WebSphere Edge Server offers enhanced load balancing via NAT, content-based routing, and Edge Server Consultant for Cisco CSS Switches. Very powerful stuff! The product can also help ensure Quality of Service (QoS) by allocating computing and network resources via custom-defined policy rules.

And, since I've brought it up, remember I mentioned how application release methodology and change control are needed to ensure site availability? Well, here's how it could work…Let's say that your new application is evolving rather quickly. Six new versions are planned for the next 12 months. How can you ensure continuous application availability with little or no downtime, while providing a mechanism for easy and near instantaneous restoral of the previous system? Well, you install production and nonproduction groupings of servers. Common services that both groupings rely on, and that will typically be unaffected by an application upgrade, remain separate and in their own separate, or common utilities grouping. One of the application groupings remains in hot-standby mode, while the other is in production. Both groupings are loaded with the same application release. For illustrative purposes, let's call the current production grouping "A," and the standby grouping "B." A new release is announced and loaded into grouping B. The evening of cutover, redirect network traffic to grouping B and stop the flow or traffic to grouping A. Perform all real-time production testing and leave A with the previous release. When user traffic hits the application, carefully monitor those metrics you've previously benchmarked. Ensure that enhancements and functionality are performing to spec. If things aren't working as advertised, revert traffic back to A. If things are running as expected, upgrade A to contain the new release. Grouping B is now production, while A has now become the standby, at least until the next release.

Another benefit of this design covers you in the event of a catastrophic failure on the production side. So long as both groupings are on the same release, you can take the troubled production machines out of service and activate the hot-standby machines. This design goes a long way to ensure ultra-high application availability. Load balancers help you manage the flexibility you need to make this sort of design function correctly.

GEOGRAPHICALLY DISPERSED LOAD BALANCING ACROSS MULTIPLE DATA CENTERS

If you really want application survivability, you must plan on having it reside in more than one data center. This covers you in the event of the unthinkable – natural disasters or acts of terrorism. Grim, but nonetheless an unfortunate reality. So, how do you plan for something like this? The most surefire way, with no real downtime at all, is to have your traffic and transactions fed to two mirrored and live data centers.

Geographic load balancing isn't a new concept, but one that should be carefully considered. Most enterprise-class businesses probably have the real estate readily available to support a second data center. These sites cannot be too close to each other. Ideally, they'll reside several counties or states away from each other. A configuration of this nature is no easy task, but is the best high-availability solution available to help ensure business continuity.

Logistical challenges present themselves at every turn. Common challenges regularly faced include database storage and synchronization. Where physical distance between storage arrays is a factor, asynchronous transmission may be the only viable option. Be aware that WebSphere product suites, along with IBM's SHARK storage solution, provide some interesting features and functionality for these types of environments. Be sure to check them out.

In the end, geographic load balancing will play into operational realities and procedural changes. Again, knowing all the cause-and-effect scenarios will help to correctly manage expectations as to how the application and systems, will function as a whole. The major benefit to this design is the reality that if one data center "goes away" a refresh or reset will bring you to the other data center. You may have to restart whatever transaction, function, or query you were in the middle of, but your application has survived. There may even be a degradation in response time, but that is far superior to no response time at all.

REDUNDANT WIDE-AREA LINKS

Depending on your particular application and who will use it, you mustn't forget the importance of peering arrangements (in a Web environment), or conventional WAN links in a closed network. Lack of redundancy in these situations could lead to disaster should a link failure occur. It would be a shame to have a robust LAN design feeding your redundant servers only to have the faucet shut off at the door with no WAN connectivity.

SECURITY

With all this flexibility, redundancy, dual pathing, and multiple clusters, let's not forget the need for really tight network, server, and application security. A hacker can bring down your system faster than any lightning strike. Worse than that, viruses, Trojans, and God knows what else can leave you with a parade of pain for weeks to come. I can't stress enough how important is is to to find and close security holes. It's an ongoing effort that includes continuous intrusion detection, and an ever-vigilant attitude.

OPERATIONAL READINESS

Another key to business continuity is operational readiness. This is the policies, procedures, and support infrastructures that distinguish planned availability from actual high availability. And, with all this investment in hardware, software, and application development, be sure to examine your data center design (***WSDJ***, Vol. 1, issue 1) to ensure that it's up to high-availability standards.

Operational readiness includes everything from full documentation, to QA and troubleshooting procedures. It includes everything from on-call duties, to network operation center configuration. It addresses monitoring and alert status. It covers all service-level objectives. It includes procedures for on-site spare parts, parts replacement, and outside support contracts. It covers staffing models, SLA requirements, and reporting procedures. It also covers the all-so-important procedural elements of incident documentation, notification, and crisis management that could someday save your job. It's the mind behind the machine; the differentiator between your high-availability application and the competition's.

Conclusion

As you can see, you'll encounter many interdependencies, variables, and unexpected twists and turns when planning and implementing a high-availability design.

Remember, an undertaking of this sort requires that all bases be covered. So, you must remember to view application survivability in a holistic manner, where a single problem or flaw may affect everything. Keep your eyes and ears open. Embrace change, and remember that no one has all the answers. An undertaking of this magnitude requires a lot of work and a true team effort.

EXCLUSIVE INTERVIEW

Birth of a Platform

Interview with Don Ferguson

The Father of WebSphere

PART ONE OF A THREE-PART SERIES

Jack and Pat Martin, editors of **WebSphere Developer's Journal**, *recently sat down with Don Ferguson, "the Father of WebSphere," to talk about its history and his view of WebSphere's future.*

Editor's note: We would like to make a special note that Don is very modest about this "Father" title and, as a matter of fact, avoids claiming this title because so many people have provided leadership and insights.

WSDJ: WHAT WAS THE GENESIS OF WEBSPHERE?

DF: What is now WebSphere started as Component Broker about 6 years ago. I had been leading a research team, working with IBM Hursley, that was proposing and prototyping a CORBA/OO transaction server called OATM. OATM provided database and application integration, workload management, high availability, etc. I had presented the ideas around the company on several occasions.

One day, my manager, Nagui Halim, and another senior researcher, Mark Wegman, came to me and said, "Get in the car." We went to Somers to meet with Robert LeBlanc, who was the executive in charge of our CORBA/C++ projects. He offered me the job of lead architect for the efforts. He made me an offer I could not refuse.

At the same time, there was another Research Division project, called the Web Object Manager (WOM), that was working on dynamic, personalized Web/HTML front-ends. We had done some work in OATM on dynamic HTML generation, but WOM was more carefully thought through.

Many ideas from WOM charted a direction for Portal, Personalization, and the Web Container in WebSphere. Component Broker defined much of what is the EJB Container and WebSphere's overall approach to workload balancing, server groups, system management, etc.

Several key executives saw the opportunity and vision of what we were doing in CB and WOM, and supported us. The executives that have sponsored WebSphere include Robert LeBlanc – now GM of Tivoli; Steve Mills; John Swainson; Danny Sabbah, VP of WebSphere Development; and Ambuj Goyal, then a VP in Research. They hung tough through some really hard times and that's why WebSphere is what it is today.

WSDJ: WHAT WAS YOUR INITIAL VISION?

DF: The real origins of WebSphere go a bit farther back. Research started some work on scalable, clustered UNIX systems. I proposed a project focusing on CICS/6000 and MQ on the clustered systems. The project started out as a systems management project, and we selected C++ and CORBA to implement our ideas. The implementation we selected was an IBM product called SOM/DSOM that IBM Austin was developing.

The Research team discovered that 80% of the work we did turned out to be reusable frameworks for building distributed OO applications; only 20% of the stuff was CICS-specific or system management-specific. Some examples of functions we implemented included container management persistence for objects, an interceptor pattern that mixed object services into each of the deployed classes, and automatically generated Web front ends from meta data, kind of like JSPs.

One day a customer was coming to Watson and I was asked to think about the customer's problem and give a presentation. Irving Wladawsky-Berger, vice president of technology/strategy, gave me this customer's problem. He was hosting the customer visit. And I thought about it, and put it into what we were

doing with system management. The business problem was just a different style of application, and what we were doing was generally applicable to application integration, multi-tier technology, and multi-channel support.

After the presentation, we started working on that problem with the customer.

LeBlanc had heard about the customer work and had Nagui and Mark bring me up to Somers. He said, "Listen, you're in charge now, come back to me in a couple of weeks with a solution for what we should be doing." I said, "I obviously needed to think about it."

So I thought about it and scheduled a conference call with him to discuss specific concerns and what I thought we needed to do. I wanted to bring up some issues, "Well, you know my wife is going to have our first child in a month, and now is not the time for a major career decision." He said don't worry about it. My daughter was born during the first month of the project. Being a father was a good way to put things in perspective; there have been some rocky times on the CB/WebSphere evolution.

I find a couple of things to be amusing: the number of people who have been identified as the "Father of WebSphere," and the concept that innovation occurs as a single effort.

There were two types of people who supported CB, WOM, and WebSphere. The first was the executive sponsorship. Revenue commitments were overlooked for the vision. That's not usual business at IBM. They made a bet on the future of the company. That took guts.

Second was the technical engineering group. That included Tony Storey. He and I put together the initial proposal for OATS/OATM (Object Application Transaction System). Much of that research went into WebSphere. Rob High, now lead architect of WebSphere Application Server, made sure I didn't make any mistakes. Rob was also the driving force behind much of the Component Broker and WebSphere designs, as well as CORBA, EJB, and J2EE standards. Eric Herness was the lead architect for the enterprise extension to WebSphere and one of the people who helped invent our Managed Object Framework.

Jason McGee and Michael Fraenkel have been the lead architects and programmers for our Web front end and getting XML supported. Rob Will, Carol Jones, Martin Nally, Scott Rich, and Timo Salo drove the application development tool design and implementation. The AD tool support we have is one of the best-loved functions of WebSphere. Graham Dixon of Transarc, a company owned by IBM and Jerry Cuomo in Raleigh, also made profound technical impact. Each of these people, and Jim Rayfield from Research on OATM, has as good a claim as "Father of WebSphere" as anyone.

I've never seen anything like this team in my life. I would go online at 3 a.m. 'cause I couldn't sleep and I'd be getting e-mail about some problems we had in the test scenario or bring-up. On Sundays too! I continue to wonder what we did to get these people and hundreds of others to do this? Two thoughts come to mind.

They believed they were the best and they wanted to show everyone they are the best. And the second thing is, I would routinely go and talk to them and say that the fate of the IBM Corporation is in our hands. If WebSphere doesn't succeed IBM will not succeed. We have lots of other products; hardware, transaction processing, and e-business, but they believed, and literally just killed themselves. In spite of the fact that, in terms of the overall expenditure of the corporation, this wasn't a big project – but so much was hinging on its success.

WSDJ: WHEN DID IT HIT GERSTNER'S RADAR?

DF: I heard from him a couple of times – the first, a visit he made to Research when we were part way into Component Broker. Research management selected one project to show him. So we had a private review to explain what we were doing. I was all dressed up in a suit for the presentation. A peer razzed me about wearing a suit. I told him I only wear a suit for the chairman of IBM and Swiss bankers. Gerstner made a remark about the suit, "From now on just wear it for the Swiss bankers."

I was impressed that Gerstner was so quickly able to understand the value of the solution. Gerstner had been a customer of IBM and understood what we were doing right away. When I explained to him the business problem we were trying to solve, he understood the business case immediately

WSDJ: WHAT WAS THE BUSINESS PROBLEM YOU WERE SOLVING?

DF: The problem that the customer came to us with was what is called a stovepipe problem in large corporations. They had a bunch of disconnected systems that were vertically integrated from presentation down to data and business logic. There was no horizontal application integration, however. These stovepipes got built up through mergers and acquisitions, and individual line of business solution development. These systems didn't integrate data, business logic, or presentation. If you went into a call center, what you would actually see was a PC running terminal emulators to all these applications. The users would hot key back and forth and read their manuals. Basically, the end users were doing the application integration.

The customer had written a bunch of utilities that read the data from all these stovepipe applications. They sucked it out on tapes and brought in external sources of information so that they could get marketing information. They put it all into one big database. Then they would run analytics to clean and merge the information so they could make decisions on direct-marketing campaigns.

They would do a tradeoff of costs of running the mailing campaign to the probability of generating new business. It was done in a very ad hoc way. This, in essence, is what the information integration and business intelligence functions in the Data Management side of IBM do today. A lot of what has gone on with Content Manager and DB2 solves these problems.

The problem that this corporation had was twofold. First, the campaign was channel-specific. The direct-mail piece went to the customer with an 800 number to call back. If the customer went directly into a bank branch, they were told to go home and dial the 800 number. There was no way to handle it. The goal was to get the campaign service-

able over multiple channels. No matter how the customer contacted them, the information could be received and updated in the operational systems.

Second, the business wanted to do just-in-time application integration: the extract, clean, and merge process took several weeks or months. So by the time the person was actually contacted by the company, the world had changed quite a bit. The person might have already opened a portfolio account or the balances may have changed. What the company originally wanted to do for them might also have changed.

The business wanted the customer contact to bootstrap up a new version of the person, to pull together all the informations' current values. There may also have been some business-policy changes. So, the business wanted to pull the new information and policy together, and at contact time do a last-minute optimization. So the customer could be offered say, a gold-status account based on the current account information, as opposed to what was in the original mailing.

Jack Martin, editor-in-chief of ***WebSphere Developer's Journal*** spoke with Don Ferguson at IBM's office in Somers, NY.

WSDJ: SO, IT WAS DONE TO SUPPORT DIRECT MARKETING?

DF: It was more. That was the initial business problem that the enterprise had. It also was call center support, branch manager support, etc. It's not that surprising because when you would talk about the technology and show people this stuff and what it would look like, they would say "YES, if we could rebuild our infrastructure, this is the way we would do it."

The problem is you need to motivate that kind of investment based on long-term cost reduction. It's a hard thing to do in an enterprise. We used to call what we were doing "incremental business process in engineering." We'd identify 5 or 10 business problems that they couldn't solve. So we'd incrementally reengineer the business infrastructure by solving the business problems one at a time. That way we could justify the investment based on delta increase in revenues rather than delta decrease in cost. This is one of the scenarios that we had.

WSDJ: WHAT ABOUT FLOW COMPOSITION MODELING (FCM)?

DF: FCM, Flow Composition Modeling, is included in WebSphere 4.2 and our tools. It's the ability to assemble new applications from existing ones and new Web services from existing Web services. We also use it to build application adaptors.

If you wanted to build an operation that checks someone's net worth, you would go to an interface to check the balance of a savings account and then a checking account and write an algorithm. That's part of what integration/adaptation is. We now have a visual builder that allows people with flow chart skills to build these kinds of applications using the Flow Composition Model.

WSDJ: DO THEY REQUIRE TECHNICAL SKILLS?

DF: They probably require more technical skills than they should but a lot less than in previous solutions. We are working on making the tooling much more intuitive and simple.

WSDJ: DO THEY HAVE TO KNOW HOW TO PROGRAM?

DF: No. The external name for this is Web Services Flow Language. We've been working quietly with some people in the industry to make it another one of the Web services standards. Part of this is just a homogenization and convergence of some things like JCA (Java Connector Architecture) microflows, which just shipped.

FCM is one theme about how the message-orientated middleware and the application server-orientated middleware are coming together. The thing that is really powerful about this initiative is that this is another case where a lot of the people I work with have shown real leadership. Java 2 Connector Architecture is pretty much based on something we in IBM called Common Connector Framework. If you looked at the JCA spec and then looked at the Common Connector Framework, you could say they were the same thing.

One of the real insights we had when we started to do WebSphere Business Integrator was the concept of a service-oriented architecture. In WebSphere Business Integrator, if you took a set of interfaces to back-end systems that you built in JCA, you would materialize it as a Web service and so do a WSDL version of its interface. You would use the FCM tools to build the individual adapters, as well as assemble new, higher level services that implemented new functions by calling the adapters. We supported calling the Web services over distributed EJBs, MQ, and HTTP.

What the FCM and WSFL do is allow you to build new services from existing ones so you can basically build a new "Web service" that uses a WSDL interface to a business partner, WSDL interface to perhaps a CICS application, and a WSDL interface to some EJBs.

The first product we are shipping allows you to do that. And so it's a really powerful concept. In addition to current application integration, it allows you to build new components visually from existing ones. It's really got two modes of operation. Mode number one is to build the application that accesses and talks to SAP; mode number two is if you've got a bunch of components already built – you can have some new ones, all it really requires is flow-charting skill.

WSDJ: ARE YOU THE LEAD PERSON ON THAT PROJECT?

DF: Well, I set some of the high-level direction, but most of the insight and innovation was done by Mike Starkey, Mike Beisiegel, Frank Leymann, and Marc-Thomas Schmidt, Terry Borden, Tim Holloway, and Vish Narayan. In terms of significant new work, this is probably one of the biggest things we are doing.

WSDJ: WHAT IS YOUR VIEW OF VISUAL ASSEMBLY MECHANISM?

DF: Visual Assembly Mechanism – when you think about it there are two core tasks people want to perform. The first is that if I have existing components, I want to build new ones from them. The second one is if someone gave me a component, how do I customize it? How do I change its behavior? Today, the way this works is through programming. Developers write new Java code to build new components from existing ones, and modify existing code to tailor a compo-

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
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 ABRAMATIC
SENIOR VP R&D, FORMER CHAIRMAN, W3C • ILOG

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
201-802-3067
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
WIRE
XML.ORG
jCert
wrox
BASIS
SD Times
JAVA
wireless
XML
WebLogic
WebServices
WebSphere
ColdFusion

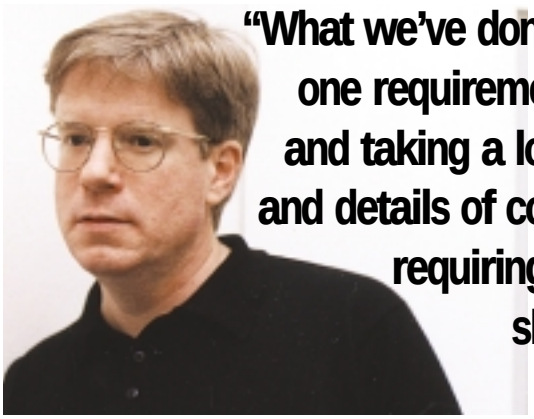
OWNED AND PRODUCED BY

SYS-CON MEDIA
SYS-CON EVENTS

nent's behavior. The FCM visual tools are a much simpler and more rapid approach to building new services/components from existing ones. Our next big innovation is support for Business Rule Beans for tailoring the behavior of existing services and components without modifying their "source."

WSDJ: CAN YOU EXPLAIN BUSINESS RULE BEANS?

DF: Business Rule Beans (BRBeans) allow people to document a service's or component's "trigger points" or "points of variability." Instead of putting policies in the component's implementation code, you document the trigger points at which policy decisions occur. The trigger points are visually connected to BRBeans that satisfy the trigger points and implement the policy. Without modifying the component's business logic, you can perform deployment and runtime customization of policy decisions through visual tools.



"What we've done is eliminated one requirement for going in and taking a look at the code and details of components and requiring programming skill for tailoring applications"

If you have a Web service, or today an EJB, an example might be that you have a method that "adds a line item" to a purchase order. If you have a purchase order component, and add a line item, the implementation is usually a method with embedded business logic that determines whether or not the new line item is valid.

With BRBeans, we tell people to do this method differently. The only thing that is in the add line item method is the mechanics of adding it in. Are you actually adding a PO to a database? It's the physics of setting the line item to point to the purchase order and vice versa.

Before performing the data changes for adding the line item, the method calls a trigger point to see if this is a valid thing to do. Is the line item valid? Is the resulting state of the PO valid? So programmers and software vendors will ship an application, and they'll ship the default policy and logic through BRBeans configured at trigger points. Then the business professionals and system integrators can use visual tools that answer questions like, "Please show me the rules that are associated with this trigger point." The tools also allow the integrators and business professionals to remove, add, and modify the BRBeans associated with a trigger point. You can also have different rule sets for different periods of time, for example when a marketing campaign is in progress.

If you look at the way this type of rules-based logic works, it has the tendency to follow the template instance pattern. My favorite example, and again this BRBeans stuff comes out of a piece of work we did in research as part of the OATS project, is the following: you could have a policy that a person can't be a manager if their IQ is above 50. Instead of embedding that in the code in the set methods for IQ, you put a trigger point on the set "IQ" and the set "manager" methods. The thing that is powerful about that is that you can have the same component in two separate places. So you may say, "Well, actually in this department that rule doesn't apply." So instead of having to have a special version of that component for that department, it's just configuration-based – what does the rule say? You just make a new instance of a BRBean with different values for the constraints and apply it to different components in different parts of the application.

This is an example of intra-entity, crossfield validation. It was on the employee component and constrains two fields: "IQ" and "is manager". The business analyst or integrator sees the set of trigger points through a developer tool. The tool also shows templates for rules and tools for building new rule types and instances. What the analyst wants is a constraint between two fields. So he clicks on "build me new constraint" and what he gets is a new instance of the template and just fills in the fields and values and associates with the trigger point. It's unlike scripting; it's more and simpler. At a particular point in time, the professional wants to take this rule, this rule, and this rule, make new instances of them, fill in the actual values and fields, and associate them with trigger points. It's a template-based thing.

What we've done is eliminated one requirement for going in and taking a look at the code and details of components and requiring programming skill for tailoring applications.

This was first shipped with WebSphere Personalization Server. Part of what we're doing now is completing this because people now have a lot of power to do campaign management with personalization on the Web front end. How do they customize and personalize the business logic within the campaign? Well, now they have rule capabilities. This also extends that capability of rule programming and complements what we have done in the past with Versata.

WSDJ: WHAT DO YOU SEE HAPPENING FOR THE NEAR FUTURE?

DF: When I think in terms of the two biggest things that are coming over the next year, I think of two things. The first is Visual Assembly for simplified component and service development. So the WebSphere support for graphical flows and a standard Web service flow language. This, combined with BRBeans will provide simple and powerful application development and customization functions.

The second big thing is multi-channel support. Customers come in on multiple channels. The problem people have had with the stovepipe problem was, in essence, that the application was vertically integrated. It was presentation, business logic, and data. The channel wasn't represented in the application.

There were quite a few things early in WebSphere that enabled multi-channel support. Obviously the Web front end was put in there, but most people don't realize that from the very beginning WebSphere was agnostic to the actual markup language that you used.

So, our Servlet and JSP front-end and tools supported various markups, including PDA and cell phone markup languages as well as voice markup for voice channels. We had support for thin clients; we had an ActiveX-VB bridge; we had a CORBA client. In essence, the middle-tier server supported the business logic in the channel. We documented that as a style of programming. We think of the Web as one channel. You can code your business logic independent of a channel. You can add other channels as needed. There's been a fair amount of stuff we've done to improve that. We've got the multilanguage markup support, Transcoding Publisher, WebSphere Everyplace for mobile devices, and WebSphere Portal Server.

Next month, the interview continues when Don Ferguson discusses how business partners will collaborate in the future using WebSphere and some of the deficiencies in WSDL that need to be standardized to make the dream come true.

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 relevant solutions to i-technology professionals and senior IT/IS strategy decision makers.

The Java East Picks on June 24, a Java Roswell CEO Keynote Panel, and comprehensive conference sessions will focus you on Java every 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 ARCHITECT ATHRA

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

BARRY MORRIS
CEO
IONA

GREGG O'CONNOR
PRESIDENT
SONIC SOFTWARE

RICK ROSS
FOUNDER
JAVA LOBBY

JAMES DUNCAN DAVIDSON
FATHER OF ANT

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 SkyHigh
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

DECLARATIVE LOGIC

Automating business logic for WebSphere

Transaction Rules Rule

BY JACKIE McALEXANDER

One of the primary uses of Enterprise JavaBeans (EJBs) is to free developers from routine infrastructure issues such as distribution, security, and persistence so they can concentrate on business logic. This recognizes that the truly valuable part of an application is the set of data and control flows that implement its business functions.

The WebSphere Application Server (WAS) has supported EJBs since version 3.0. Placing the business logic of a WebSphere application into the EJB layer should allow the logic to be centralized, understood, reused, and maintained more easily. In theory at least, leveraging EJB container services should simplify development and greatly improve programmer productivity.

The reality is usually very different. Many organizations have discovered that delivering systems with this layered architecture is a challenge. In their report, "Putting J2EE to Work," Forrester Research concludes that most organizations will face a bumpy adoption road that won't be fixed by advances in the J2EE specification. For EJBs, the difficulties include a steep learning curve, the complexity of designing for both flexibility and performance, and the relatively low-level Java language used to implement business functions.

All this points to the need for another approach that specifically addresses the development and management of business logic in WebSphere applications. According to Forrester, this approach would automate away complexity, so developers "never have to know or care that EJBs require six different deployment files."

An approach using declarative logic, also known as rules, to specify transaction business logic is delivering a high degree of EJB automation for J2EE application servers such as WebSphere.

Automated Business Rules

Within the software industry, the term "rule" is defined quite broadly. IBM's MQSeries uses routing rules to direct messages; WebSphere Personalization uses parameter-driven rules to control content delivery; finally, an expert system uses inference rules to make complex decisions.

To clarify our discussion, the rules used to automate EJBs are transaction-logic rules. Rules expert Barbara von Halle explains that such rules are data-change–oriented, which means that they execute when the data to which they are attached is touched.

Specifying business logic through rules has many benefits. First, because a single rule can control the behavior of a whole set of objects, rules are much more productive than procedural or even object-oriented languages. Next, because they are centralized and easy to understand, rules can be located and changed quickly. Finally, because they are unambiguous and self-documenting, rules become a tool for communication between users, analysts, and developers.

Rule-based specifications, of course, only begin the business logic automation process; for the specifications to be useful, they must be executable. That is, they must be turned into a set of business objects whose sophisticated interactions can be driven by changes to data values. This is done through a transaction logic "server" or "engine." The engine is typically installed in an EJB container within the WebSphere application server.

In addition to executing rule events, a transaction logic engine may also provide other services to improve the performance and manageability of rule-based logic. Ideally, these services would include:

- Optimizations for chained and intra-object logic
- A prebuilt framework for fast client access to sets of business objects
- Tools to trace rule execution and monitor logic statistics

AN APPLICATION EXAMPLE

The IBM Redbook, *Application Development Using Versata Business Logic Automation for WebSphere*, shows how transaction logic rules can be used to implement the business logic in a well-known application, IBM's Trade 2 application performance benchmark.

Trade 2 has a sophisticated Model-View-Controller architecture that simulates an online stock brokerage. Its functionality includes buying and selling stock "holdings." By default, the business logic in Trade 2 is implemented in EJBs.

Using Versata, the Trade 2 EJBs are automated using rules. In addition, the Trade 2 business logic is extended to include additional functions, such as controlling margin selling, selling partial holdings, and charging personalized commission rates.

The model of the rules-extended Trade 2 application has five core business objects, described as:

ABOUT THE AUTHOR

Jackie McAlexander, is a solutions architect and senior technical marketing manager at Versata, a leading vendor of business logic development and management software. She is lead author of IBM's Redbook, *Application Development Using Versata Business Logic Automation for WebSphere*.

EMAIL

jackie_mcalexander@versata.com

- ***Account:*** The user ID, cash balance, and number of transactions
- ***Profile:*** Descriptive information about the account
- ***Quote:*** Stock symbols and prices
- ***Holding:*** A block of stock purchased by an account
- ***Transaction:*** Individual buy and sell transactions against a holding

The next section uses the Trade 2 object model for its examples.

Classification of Rules

Over the years, there's been considerable analysis of the types of rules needed to specify application business logic. The classification below is largely based on the 1995 GUIDE Business Rule Project (see Fig.1).

- ***Relationship rules*** specify associations between entities. Rather than controlling transaction behavior itself, they enable other rules to be designed and enforced.
- ***Derivation rules*** are algorithms that derive attributes from other attributes. Typically, derived attributes can be persisted or nonpersisted (in which case they represent runtime-only values.)
- ***Inference rules*** use one or more truths to arrive at a new truth. The new truth usually derives an attribute.
- ***Constraint rules*** specify the policies of a business. They govern under what conditions operations can proceed.
- ***Action rules*** initiate another business event, message, or activity based on some condition.

RELATIONSHIP EXAMPLE:

Relationships typically imply a parent-child association between entities, although they can be used for more complex models. An example of a simple relationship is:

"An Account Has Holdings."

RULE TYPES	RELATIONSHIP DERIVATION INFERENCE CONSTRAINT ACTION

FIG. 1: Business Rule Types

Relationships are the foundation of many rules, and transaction logic servers can automate many operations between related entities. For instance, through the relationship described above an Account can automatically count and sum all of its related Holdings. Holdings can automatically check for sufficient funds in the related Account. Furthermore, as they are entered, Holdings can automatically check to ensure they are associated with valid Accounts.

DERIVATION EXAMPLE

Derivations are computational, which means that a formula determines the value of an attribute. An example is a rule for a Holding that says

"QuantityOnHand = QuantityPurchased – QuantitySold"

In addition, computational derivations can use the attribute values from several objects in computations. An example of this is a rule for selling a stock that says

"The Transaction Amount is the Transaction Quantity multiplied by the Price (from the Quote object) navigated to through the related Holding object"

Leveraging its relationship rules, the transaction logic server can automatically navigate to other objects to retrieve attribute values. A logic server will typically have a GUI for building rules.

INFERENCE EXAMPLE:

Inference rules can be thought of as a specialized type of derivation, where an attribute is derived from a truth. An example is a rule that says, for an Account,

"AccountType is wholesale if the Balance is greater than $500K."

Like computational derivation, inference derivations can chain together evaluations and calculations from several entities. It's possible, for instance, for a rule to say for an Account,

"An AccountType is wholesale if the Account Balance is greater than $500K (Account object) AND the number of ActiveHoldings is greater than 20 (Holding object) AND the average TransactionSize is > $10K (Transaction object)."

CONSTRAINT EXAMPLE

Constraints define legal states in the system. For transactional systems they define legal values of data that are allowed to exist. A rule to restrict (constrain) margin selling defines one such state. It says, for Account

"Balance can't be less than zero"

Although it sounds simple, this constraint shows the power behind some logic servers. These servers can compile high-level, unordered rules into automatically sequenced chains of object interactions.

For instance, in the Trade 2 example, the following "Buy" rule-chain is constrained by the margin-selling rule above. When buying a stock:

1. The Logic Server begins to create a new Holding for this Symbol, Date, and Quantity.
2. It gets the stock's price from the associated Quote.
3. It begins to insert an initial "Buy" Transaction object .
4. It calculates the cost of the stock (multiply Price by Quantity).
5. It determines the Commission (by checking the Account entity to see whether the AccountType is a wholesale or retail account, checking the current CommissionRate for that Account Type and multiplying the CommissionRate by the Transaction Amount).
6. It calculates the total Transaction Amount.
7. It begins to update the Transaction Count for this Holding.
8. It begins to debit the Account Balance with the Transaction Amount.
9. Finally, it encounters the constraint.
10. If the constraint is violated (if the new Balance would be less than zero)
11. It unwinds the entire operation (rolls back changes to the Holding, Transaction, and Account objects).

In addition, to optimizing performance, a logic server may be able to bring all the objects needed for a transaction chain into a transaction cache, so the entire chain can be processed with a single database I/O.

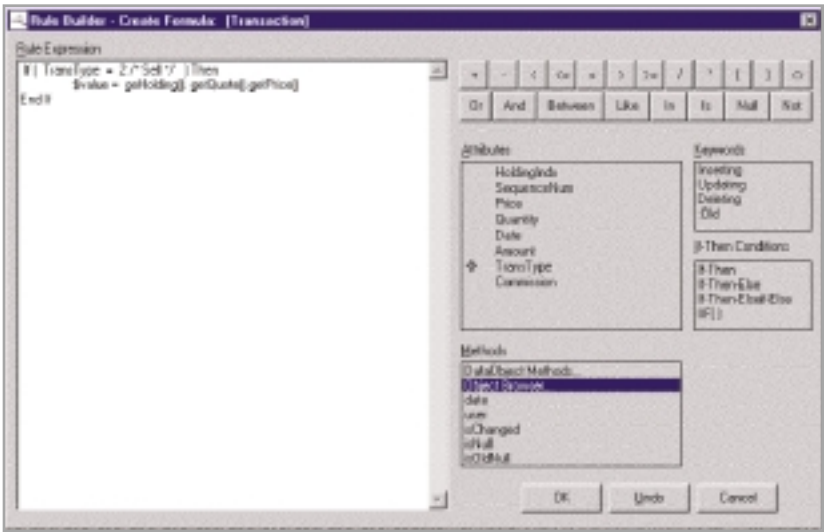


FIG. 2: A Derivation rule in the Versata Rule Builder

ACTION EXAMPLE

The final type of rule is an action rule, more completely called an Event/Condition/Action. The Event is the operation and entity being watched by the logic server. The Condition must be met in order to proceed. The Action is the result that should occur if a condition is met: An example of this is a rule that says,

"When adding a new Transaction, if the TransactionType is 'sell,' credit the related Holding's related Account Balance."

Typically, a transaction logic server will implement this with a rule "listening service." The logic server will receive the event when all other rules leading up to this event/condition have evaluated "true."

Extending Rules with Java

The Event/Condition/Action example above demonstrates a key requirement for a rules system – extensibility. The rule has been extended with a custom Java method "credit".

While transaction logic rules are estimated to cover more than 90% of application business logic, any automation approach must address the remaining requirements. Extensibility can be provided in many ways: for instance, the rule system can:

- Surface its underlying classes so that the Java developer can subclass and extend them
- Provide predefined extensibility points, for instance through predefined events, where custom Java code can be inserted
- Allow rules to call methods from custom Java components to control rule behavior

Combined, these extensibility options should allow the developer to use a rule-based approach for almost any transactional application.

Application Use of Rules

Another way to look at rules is by application function. Here are just some of the common uses of rules that can have a big payoff for WebSphere applications.

TO AUTOMATICALLY SEQUENCE AND OPTIMIZE TRANSACTIONS

One of the most common patterns coded into EJB logic is the getting and setting of attributes in multiple objects within the scope of transactions.

The Trade 2 application "Buy" function is a simple example. Given a UserID, stock Symbol, and Quantity, the Buy operation:

- Finds the related Quote using the stock's Symbol
- Checks that the Quote is valid and gets its Price
- Finds the related Account for that UserID
- Checks that the Account is valid and gets its Balance

RESOURCES

GUIDE Business Rule Project: www.businessrulesgroup.org/first_paper/

von Halle, Barbara, *Building Business Rule Systems*, John Wiley & Sons, 2001.

WebSphere Performance Benchmark Sample (Trade 2 Application): www.ibm.com/software/webservers/appserv/wpbs_download.html

Application Development Using Versata Business Logic Automation for WebSphere: www.redbooks.ibm.com

Koetzle Laura, "Putting J2EE to Work," Forrester Research, July, 2001.

- Creates a Holding and checks to see that this succeeds
- Debits the Account Balance

A 32-page session bean implements this (and other) logic in the original Trade 2 application. The automated logic demonstrated by Versata replaces the hand-coded Java with four rules.

FOR DATA VALIDATION

Validation is frequently overlooked when estimating the amount of business logic in an application; its design and development consumes costly programming resources. Moreover, validation (and related error handling) code is usually sprinkled throughout various client-tier and logic-tier components. This makes logic difficult to reuse and maintain.

Rules allow data validations to be attached to objects directly, as part of their metadata. Using rules to specify attribute validations has several benefits. First, validations, like all rules, will be applied consistently across all applications. This increases system integrity.

In addition, validation metadata, like other rule-based metadata, can be accessed through object methods. This makes it possible for client components to derive their behavior from business entity metadata. This extends the use of the Model in the Model-View-Controller architecture and could allow client components to adapt automatically to changes in business logic.

FOR EJB REFERENTIAL INTEGRITY

Referential integrity addresses the need to maintain consistency between one business object and all the other objects that refer to it.

In applications where all data is kept in the same relational database, enforcing referential integrity can be left to the database management system. If, however, there is the potential for data to come from more than one source, integrity must be assured through application code. In J2EE applications, the place for this code is in EJBs. (Whether it should be placed in the parent entity EJB, the child entity EJB, or in a third, session EBJ, is frequently debated.)

Rules provide an easy and consistent way to enforce relationships and integrity, even between objects in different databases or legacy applications.

FOR AD HOC OBJECT QUERIES

Since rule chains frequently must find other objects by attribute name or value, the transaction logic server automatically implements a complete set of "finder" methods for objects and relationships. These methods may be available to developers, either directly through EJB remote interfaces or through APIs.

For instance, the Versata client library provides a JDBC-like API to submit ad hoc queries against objects, including objects that are joined from multiple entities. The libraries deliver optimized ResultSets of serialized "rows" of objects with client navigation (first, next, last) as well as buffered reads and saves. This simplifies client development and automatically provides the performance found in J2EE best-practice patterns such as value objects and value-object lists.

TO INITIATE ASYNCHRONOUS EVENTS

Events/Action rules integrate the automated system with external applications. Events initiate exception processing. Although Trade 2 is an entirely synchronous application (transactions are committed or rolled back immediately), most enterprise systems have some asynchronous operations, such as putting a message on a message queue.

Rules watch data changes and can initiate synchronous transactions or asynchronous events based on those changes.

Beyond Transactions to Processes

Beyond transactions, which are typically short-lived interactions with applications or services, lie processes. Processes typically involve a whole series of interactions that span systems, people, and time. Processes can be automated by their own set of specialized rules and engines.

A future article will examine the natural evolution of rule-based WebSphere applications into sophisticated process-driven systems.

WEBSPHERE USER GROUPS...

www.websphere.org offers Web space to user groups worldwide for posting contact and meeting information. The site's LISTSERV offers peer-to-peer e-mail Q & A and has hundreds of subscribed members from all over the world. The site's current instant poll ("Share Your Views") enables members to vote for appropriate meeting topics. For site sponsorship and general information contact info@websphere.org.

Users are eager to share and increase knowledge

WebSphere Users Group Fills Knowledge Gap

BY MORGAN SMYTH

The quest for knowledge and the opportunity to share experiences with WebSphere is evident with the successful launch last year of Canada's first WebSphere Users Group.

ABOUT THE AUTHOR

Morgan Smyth is president of Toronto-based Braegen Group Inc., which has architected and implemented approximately 30 WebSphere sites throughout North America, mainly for commercial institutions. For more information about Braegen, visit www.braegen.com.

E-MAIL

morgan@braegen.com

Established with just 30 members, the Southern Ontario WebSphere Users Group has grown by leaps and bounds to 180 members ranging across all industries and organization sizes.

Brought together by the common goals of building a network and increasing their knowledge base, the WebSphere Users Group will further expand by merging with a local MQSeries Users Group. Membership of the merged group will top 300 and be composed of developers, architects, designers, managers, and administrators.

The aim of the joint group is to promote knowledge and technical understanding of the IBM WebSphere family of products and to focus on common issues and experiences. More importantly, the users group provides an opportunity to learn first-hand from the experiences of others in implementing WebSphere.

The Southern Ontario WebSphere Users Group has an executive committee of five members which meets regularly throughout the year. From our experience, it's important to have executive members from across all industries who are users themselves.

Having an IBM representative, as well as a third-party vendor, sitting on the executive committee, is also helpful.

The executive committee plans three meetings yearly where members can hear presentations on proven innovative WebSphere solutions, from both suppliers and user organizations. Plenty of opportunity exists for members to network before, during, and after the meetings. Soliciting input and feedback from members following each meeting ensures a successful meeting next time around and ensures all voices are heard in the process.

The WebSphere Users Group has taken the concept of team-based, shared learning to a larger community. The meetings provide the setting for a large interactive classroom from which members can gain valuable knowledge, advance career opportunities, and learn more about new and innovative applications of WebSphere.

The user group also benefits from its close proximity to IBM facilities. Our last meeting, for instance, was held in December 2001 at the IBM Toronto Software Development Lab. Members not only had an opportunity to peek inside a state-of-the-art facility devoted to software development, but also heard presentations from key members of the IBM WebSphere Software Commerce Suite development team. Invaluable insight was gained from this team, as well as an insider's view of IBM's WebSphere plans for 2002 – an exciting treat for those in attendance.

A special guest and keynote speaker for the meeting was Hershel Harris, vice president of WebSphere Server Development and director of the Toronto Lab IBM Software Group. Harris told the audience that the WebSphere family of products is one of the most rapidly growing e-business platforms, with 35,000 global users, and that WebSphere ranks highly with CIOs.

With its growing popularity and market share, it's important that users have a forum from which they can learn, share ideas, and learn new applications of WebSphere. WebSphere technology is key to the implementation of effective e-business strategies that result in increased efficiencies, productivity, and ultimately, ROI for users.

Participation and attendance has grown exponentially with each successive users group meeting. The turnout is testimony to the fact that WebSphere users are eager to increase their knowledge, share knowledge, and increase the benefits of networking. It also demonstrates their eagerness to capitalize on this innovative e-business software and apply it to their own enterprises in creative ways.

As a founding member of the group, I highly encourage this forum for a growing and vibrant community. It's an excellent avenue to share experiences, gain valuable knowledge and keep abreast of an innovative technology.

For more information on the users group and upcoming activities, visit www.websphere-users.ca.

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

NORBERT MIKULA
XML CHAIR
BOARD OF DIRECTORS, OASIS
INDUSTRY EDITOR
DR. DOBB'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, FORMER CHAIRMAN, W3C • ILOG

TYLER JEWELL
PRINCIPAL TECH. EVANGELIST • BEA

DAVID LITWACK
CEO
SILVERSTREAM

ANNE THOMAS MANES
CTO
SYSTINET

BARRY MORRIS
CEO
IONA

RICK ROSS
FOUNDER
JAVA LOBBY

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™
ADDS
bea
TogetherSoft
SilverStream
Actional
MERANT
Altaworks
PolarLake
ALTOVA

MEDIA SPONSORS

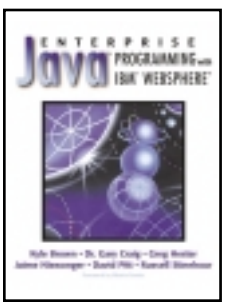
Federal Computer Week
Webservices.org
XML-JOURNAL XML TIMES.com
Java Skyline
CF Advisor
WebServicesMall
XML.ORG
WIRE
iCert
wrox
OASIS
XML.ORG
SD Times
JAVA
wireless
XML
WebLogic
WebServices
WebSphere
ColdFusion

OWNED AND PRODUCED BY

SYS-CON MEDIA
SYS-CON EVENTS

Chapter 6

Using the IBM WebSphere Application Server, Advanced Edition



In previous chapters, we learned how to use the IBM WebSphere Test Environment (WTE) inside VisualAge for Java to create and to test servlets. Now we need to understand how to use our servlets outside a test environment. This can be accomplished by deploying our servlets to the IBM WebSphere Application Server, Advanced Edition (WASAE). The WTE is used at development time, whereas the WASAE is used for staging or production purposes.[1]

ABOUT THE BOOK

Enterprise Java™ Programming with IBM WebSphere (with CD-ROM)

By Kyle Brown, Gary Craig, Greg Hester, Jaime Niswonger, David Pitt, Russell Stinehour

Addison Wesley Professional
ISBN 0201616173
$44.95

You may purchase this book with a 10% discount at: www.informit.com/promo/wsdj

The IBM WebSphere Application Server, Advanced Edition, provides the necessary tools to deliver J2EE-based applications. The IBM WASAE allows a machine—more commonly called a node—to host multiple application servers. Within each application server, both an EJB container and a servlet engine—known in J2EE parlance as a Web container—can exist. Within a servlet engine, multiple Web applications can be defined. Specifically, this chapter covers

- WASAE architecture
- Testing the WASAE installation
- Using the administration console
- Using the XMLConfig tool
- Configuration issues

WASAE is available on Windows, Novell, UNIX, and OS400 platforms. For this chapter, we demonstrate installation only under Windows NT. However, we point out where configuration steps differ in other environments, such as UNIX.

WASAE Architecture

IBM WASAE, is based on the J2EE specification. Most J2EE products provide a container for each application component type: application client container, applet container, Web component container, and Enterprise Bean container. WASAE provides out-of-the box support for all the containers except the applet container. However, WASAE provides this support, using the IBM Java 1.2.2 Java runtime environment (JRE) with enhancements. We discuss applet deployment issues in a later chapter. Figure 6.1 shows the J2EE architecture.

WASAE is designed for a multinode—multiple-machine—environment. WASAE defines an additional container called a WebSphere domain, which consists of a collection of nodes that can be configured and administered together. All the configuration data for all the nodes contained in a WebSphere domain is kept in a single, shared—but, ideally, replicated—repository. This repository, known as the WAS database, is kept in an RDBMS.[2] Using a single database for all configuration data provides a single point of communication for all nodes and enhances the possible solutions for workload management and balancing, replicated services, and maintenance. Additionally, using a single database for all configuration data greatly decreases maintenance. If each node had its own set of configuration files, you would have to edit the files on each node individually in order to administer it. In WebSphere's model, a shared repository contains all the configuration data, so you can manage the configuration remotely from a desktop because WASAE implements its configuration objects as EJBs, using its own EJB support. Therefore, access to the configuration data by multiple nodes is controlled through the use of the transactional and persistent capabilities of the EJB Entity Beans. This access is done through RMI over IIOP, which allows the configuration to be managed securely over the Internet.

Within each node, WASAE allows multiple application servers to be configured. Each application server may contain a servlet engine—a J2EE Web container—and an EJB container. Both the servlet engine and the EJB container within a particular application server operate within a single shared Java Virtual Machine (JVM). Application servlets and various other Web-based services are defined within the context of the servlet engine. EJBs are deployed within the EJB container. A sample WebSphere domain is shown in Figure 6.2. Note that this domain con-

tains three nodes and that you are shown an exploded view of node 1. Also note that only node 3 contains a Web server.[3]

Because WASAE was designed to work in a multinode environment, you must understand the requirements for each node. Minimally, each node in a WebSphere domain must contain an instance of the WebSphere admin server. The admin server performs many functions, including starting, stopping, and monitoring all configured application servers and providing a location service daemon (LSD), a persistent name server (PNS), and a security server. In addition, the so-called nanny process, a watchdog process, restarts the admin server in case of failure.

In Figure 6.2, node 3 contains a Web server. As is shown, it is not necessary to have the Web server on all nodes. If an HTTP request is sent to node 3, the Web server uses the WebSphere plug-in, possibly with an additional servlet redirector to route the request to the appropriate local or remote application server. The local node then services the request, if possible, or routes it to the proper node, based on the configuration data contained in the EJB Entity Beans using the database.

Some Definitions of WebSphere Components

Now that you have seen the overall architecture of WebSphere's process architecture, it's a good time to learn some of the more detailed definitions of the WebSphere components you will use and how they are configured in WebSphere, Advanced Edition. You will learn about the topology view of the WebSphere administrative console, a view arranged as a tree of configuration objects. But before you begin looking at the view, let's examine the following objects you will see:

- ***Domain:*** the root of the WebSphere configuration tree. A domain is a set of nodes that all share a single administration database: a single shared WAS database.
- ***Node:*** a server machine. Each node is uniquely identified within the domain by its network name, such as its TCP/IP host name. Server processes—called *managed servers* in WebSphere—run within a node. WebSphere defines two types of managed servers: an application server, a Java process; and a generic server, or any process, such as a CORBA server or an RMI server. All the processes managed on a node by WebSphere may be started or stopped by a single command in the administrative interface.
- ***Application server:*** a JVM that is managed by WebSphere. Each application server may contain a servlet engine and an EJB container.
- ***Servlet engine:*** a Web container as defined in J2EE. An application server that contains a servlet engine thus has the infrastructure and the classes to execute servlets. A servlet engine contains a set of Web applications, which are isolated logical groups of servlets.
- ***Web application:*** a number of common objects brought together into a unified representation of an application. A Web application may contain a group of logically connected servlets and also defines a class path that the servlet class files and associated files are taken from. The Web application also defines a JSP processor that indicates how JSPs will be interpreted, such as using the JSP 0.91 or the JSP 1.0 specification. In addition, the Web application defines a root URI, which will be a part of the URI of all the servlets it contains, indicating the logical grouping within the URL.

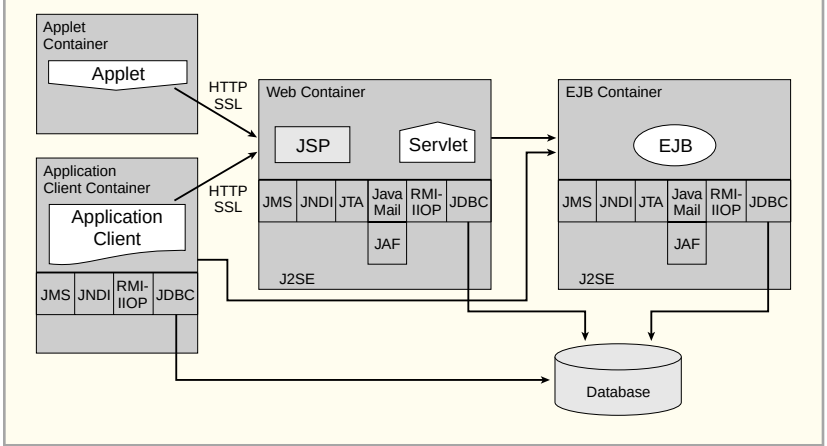


FIG. 6.1: J2EE Architectrue

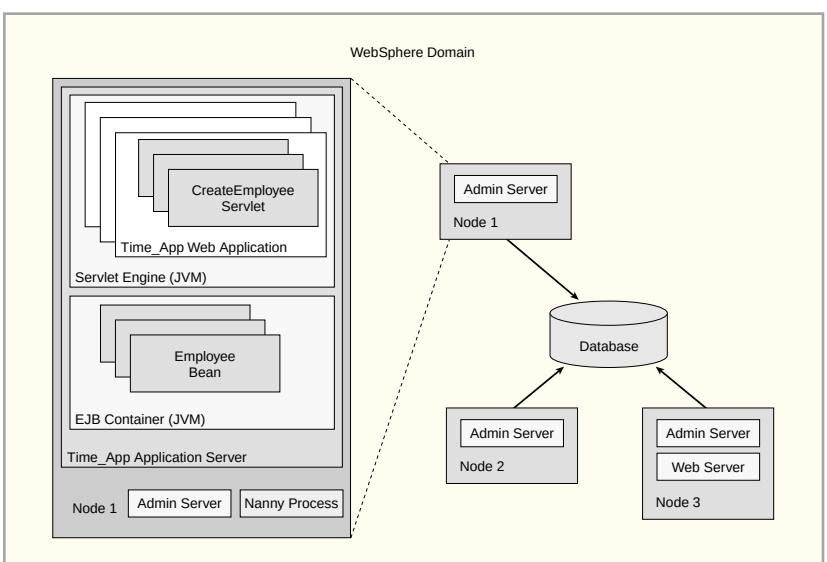


FIG. 6.2: Example of a WebSphere Domain

1. Note that whereas servlets and JSPs can be deployed into WebSphere Application Server, Standard Edition, we discuss the capabilities only of WASAE in this chapter.

2. In WASAE version 3.5, the WAS database can be installed in IBM DB/2, Oracle, or Sybase for standalone or networked use or in the Instant DB database for standalone use.

3. WebSphere supports multiple options for separating the Web server from the application server. The configuration and setup of these options is beyond the scope of this book. See [Ueno] for details.

Finally, the Web application defines a document root path to locate the JSPs used by this Web application.

- ***Servlet:*** the servlet at the bottom of this particular branch of the topology tree. Servlets can be accessed in a Web application in one of two ways. The most common, and most maintainable, way is to individually define servlets by specifying the servlet class name and servlet name, as well as a set of URLs that the servlet will be identified by. Servlets may also be defined within a Web application by including a special servlet, called the "Invoker" servlet, which allows servlets to be invoked with a URL containing their fully qualified class name.

Testing the Installation

Once WASAE is installed, you can then test the installation to ensure that everything is functioning correctly. In this section, you will test the installation of the WASAE by using the default server, which can automatically be installed for this purpose.

BEFORE YOU START THE ADMIN SERVER

During the installation of an admin server on a node, the WAS database must be identified. Specifying a JDBC URL and a JDBC driver class identifies the database. This information is stored in the admin.conf file, which can be found in the [ROOT_DIR]\WebSphere\AppServer\bin directory, where [ROOT_DIR] is the root directory where the admin server is installed.[4] For learning purposes, it is best to start with one node before creating a multinode environment. Here, we will assume that only one node exists in the WebSphere domain and that the WAS database is located on the node itself. If the WAS database is implemented in DB2, using the default database name, you will find the following lines in the admin.conf file:

```
com.ibm.ejs.sm.adminServer.dbDriver=COM.ibm.db2.jdbc.app.DB2Driver
com.ibm.ejs.sm.adminServer.dbUrl=jdbc:db2:was
com.ibm.ejs.sm.adminServer.dbUser=userid
com.ibm.ejs.sm.adminServer.dbPassword=password
```

If you want to force the admin server to create the default server application server when it starts, ensure that the install.initial.config property is set to true, as in the following line:

```
install.initial.config=true
```

If you have problems because the database becomes corrupted, you may have to drop and recreate the database. To do this, open a DB2 Command window and use the following DB2 commands.

```
REM ensure DB2 has the correct computer name
setdb2n.exe %computername%
REM Drop the existing WAS database
DB2 DROP Database WAS
REM create a new WAS database
DB2 CREATE Database WAS
REM increase the application heap size to 256
DB2 UPDATE DB CFG FOR WAS USING APPLHEAPSZ 256
```

At this point, the admin server has been reinitialized. If you ever need to reinitialize the server again, you can perform these same steps. However, you may want to use the XMLConfig tool, described later, to obtain an export of the configuration data so that it can be reimported into the new configuration database.

STARTING THE ADMIN SERVER

Most commonly, the admin server is started from NT services in Windows NT. The service is named IBM WS Admin Server. A shortcut to this service is in the Start menu of Windows NT under IBM WebSphere.Application Server V3.5>Start Admin Server. If you need to debug the admin server, you can also start it from a command line window. You will find a batch file that can be used to start the admin server in [ROOT_DIR]\WebSphere\AppServer\bin\debug\adminserver.bat, where [ROOT_DIR] is the root directory where the admin server was installed.

OPENING THE ADMINISTRATOR'S CONSOLE

The primary tool used to view and to change the configuration of a WebSphere domain is the WebSphere administrator's console, a Java-based application that acts as a client to the configuration EJBs that are running in the admin server. In WebSphere 3.5, an administrator's console can connect to any available admin server in the domain. In this chapter, we assume that the default configura-

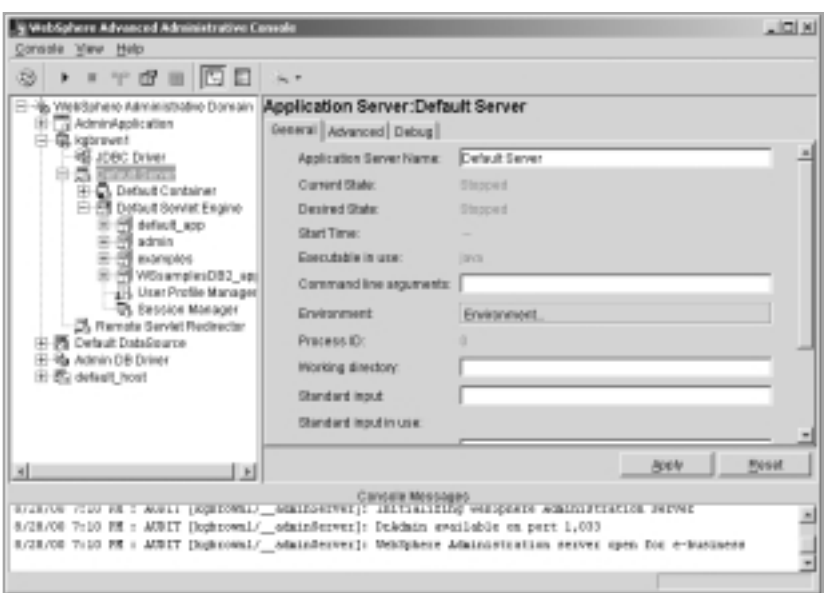


FIG. 6.3: The Administrative Console

4. [ROOT_DIR] is usually a drive letter under Windows NT or a directory mount point under UNIX.

5. If you forget to launch the admin server before opening the administrator's console, you will see a dialog box informing you "The Admin Client failed to connect to the admin server." If this happens, first start the admin server and then try opening the administrator's console.

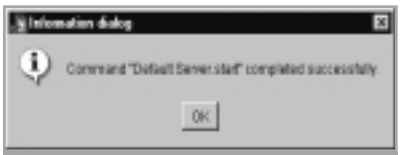


FIG. 6.4: The Start Dialog

tion, in which the administrator's console connects to an admin server on the same node, will be followed. To start the administrator's console under Windows NT, choose IBM WebSphere.Application Server V3.5.Administrator's Console from the Windows NT Start menu.[5]

STARTING THE DEFAULT SERVER

When the admin server is started, it automatically starts all the application servers that were in the "started" state before the admin server was last shut down. If this is the first time that you have started the admin server, the default server application server will not automatically start. Instead, you must ensure that it is started. In order to start a server, you must open the administrator's console as described earlier. The WebSphere Advanced Administrative Console is shown in Figure 6.3.

Select the Default Server. To start this application server, click the Start button. When the server has started, the dialog shown in Figure 6.4 will be displayed.

STARTING THE WEB SERVER

The last thing that has to be done is to start the Web server. WebSphere supports a number of Web servers, including IBM's HTTP server, which is supplied with WASAE, and Web servers from Microsoft, Netscape, and the Apache consortium. The Web server will listen, by default, on port 80 for HTTP requests and will forward any requests for servlets through the WebSphere plug-in to the applicable application server. From the NT Services panel, select IBM HTTP Server, and then start it. If you are using a Web server other than the IBM HTTP server, start it by following the instructions provided by the Web server manufacturer.

TEST USING THE "SNOOP" SERVLET

The "Snoop" servlet echoes back to the Web browser the details of the request, the requested server, and the servlet context. This servlet is part of the initial configuration that comes with the default server. In order to invoke "Snoop" servlet, open a Web browser and type http://localhost/servlet/snoop on the URL line of the Web browser. If WebSphere is installed correctly, the "Snoop" servlet will display a response as shown in Figure 6.5

Creating the "EmployeeSys" Application Server

Now that you have verified that the WASAE product is installed correctly, you need to create a new application server for the book's case study. The case study defines an application server named "EmployeeSys" that contains both an EJB container and a servlet engine (Web container). Additionally, the servlet engine defines a Web application with a root URI of /TimeApp. The steps that follow will help you to export the Java classes from VisualAge for Java and to configure the "EmployeeSys" application server in WASAE.

1. From the Projects tab in the VisualAge for Java Workbench, select the CreateEmployeeServlet class. This class is contained in the WS Case Study project and the com.wsbook.casestudy.servlet package. (Remember that you created this project, package, and servlet in the previous chapter.) Using the pop-up menu from the right mouse button, select the Export option, as shown in Figure 6.6.
2. When the dialog appears, select the Directory radio button and click the Next button. When the next dialog appears, type the export directory:

```
[ROOT_DIR]\WebSphere\AppServer\hosts\default_host\TimeAppWebApp\servlets
```

as the export directory. Click the Finish button to start the export. (See Figure 6.7.) At this point, a dialog will appear informing you that this directory has not been created. Click OK to let

BOOK EXCERPT

WebSphere create the directory.

3. Ensure that the IBM WS Admin Server NT service is still running, and open the WebSphere administrator's console.
4. Inside the administrator's console, you will now use the Start Wizard pull-down menu to start a wizard that will help you create a new application server.

   Select Create Application Server from the pull-down menu. The wizard that begins should resemble Figure 6.8.
5. The first panel that is displayed allows you to include either an EJB container or a Web container or both containers. Select both check boxes so that your new application server can service EJBs and Web applications. Click the Next button.
6. Next, you will specify the application server properties: the application server name, command line arguments, working directory, and standard output and standard error. Type EmployeeSys for the application server name. Change the name of the standard output to empsys_stdout.txt, and change the name of the standard error to empsys_stderr.txt. Click the Next button.
7. Specify that the server not be automatically started after creating it. Click Next.
8. Select your node from the list of nodes and click Next.
9. The next panel allows you to specify Enterprise Beans to be installed on this server. For now, do not make any changes to this panel. Click Next.
10. On the General tab, accept the EJB container name EmployeeSys Container. Click Next.
11. On the Select Virtual Host page, select the default_host virtual host, and click Next. We will not be defining new virtual hosts for this node, so we will use the default host (which refers to the default host name defined for this node in TCP/IP).
12. On the General tab of the Servlet Engine Properties page, accept the servlet engine name EmployeeSysServletEngine. Click Next.
13. On the General tab of the Web Application Properties page, do not accept the default Web application name; instead, type the Web application name TimeAppWebApp. Select default_host as the virtual host. Type /TimeAppWebApp as the Web application Web path. This path will become the root URI for servlets in this Web application. Select the Advanced tab.
14. On this tab, you can change both the document root and the servlet class path. Accept the default document root [ROOT_DIR]\WebSphere\AppServer\hosts\def

6. Note that it is a best practice not to use the "Invoker" servlet. We will describe these and more best practices later. Nonetheless, we will use this servlet in our example.

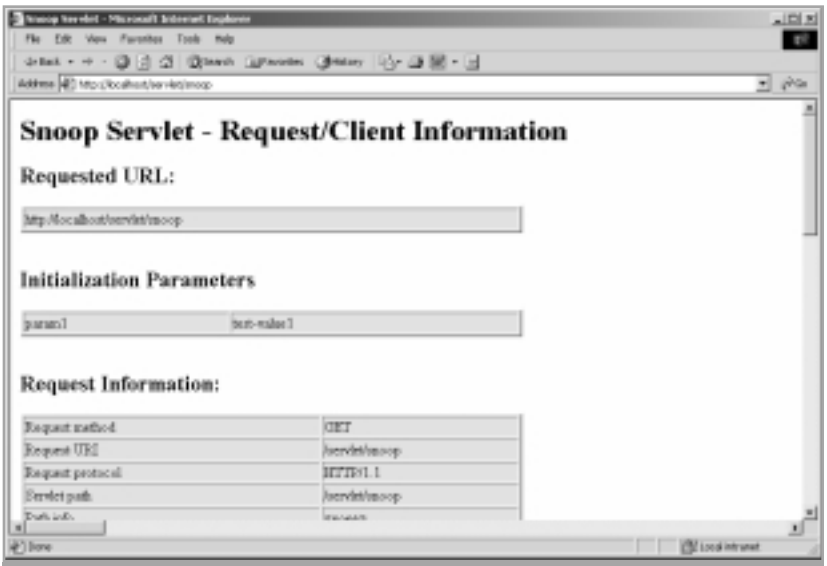


FIG. 6.5: Results of the "Snoop" Servlet

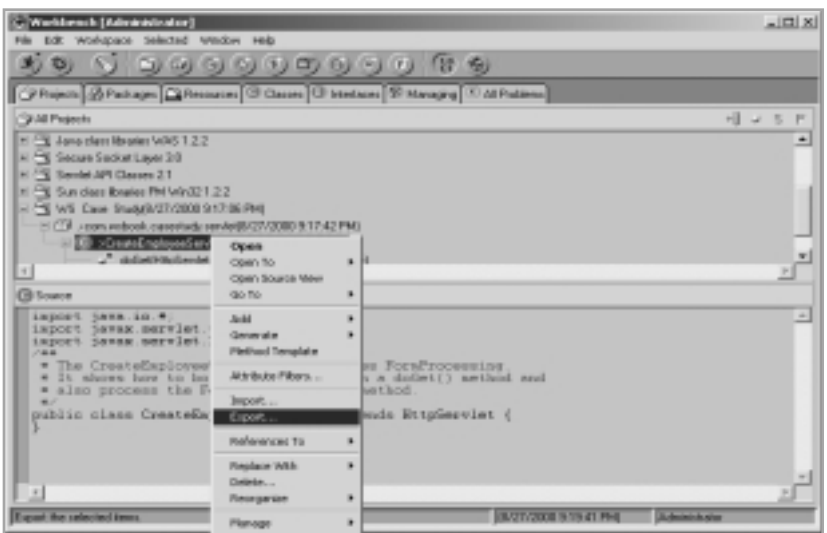


FIG. 6.6: Exporting CreateEmployeeServlet

FIG. 6.7: Exporting "CreateEmployeeServlet" to WASAE

ault_host\TimeAppWebApp\web. Accept the servlet class path as [ROOT_DIR]\WebSphere\AppServer\hosts\default_host\TimeAppWebApp\servlets. Click Next.

15. This last page of the wizard allows you to specify system servlets that will be part of the Web application. Uncheck the Enable File Servlet check box. Check the Enable Serving Servlets By Classname check box.[6] Select Enable JSP 1.0 so that the JSP 1.0 page compile servlet will be used by the Web application. Click the Finish button to create the application server.
16. When the process has completed, drill down to the "EmployeeSys" application server to show the configuration as in Figure 6.9.
17. Select the Web application TimeAppWebApp in the administrator's console. Then, using the pop-up menu by clicking the right mouse button, select the Create-.Servlet option, as shown in Figure 6.10.
18. When the Create Servlet dialog appears, type the servlet name CreateEmployee. Type a Web application name of TimeAppWebApp and a description of Create an employee. Type the class name com.wsbook.casestudy.servlet.CreateEmployeeServlet, as shown in Figure 6.11.
19. Finally, add a servlet Web path. Click the Add button to display the Add Web Path to Servlet dialog. Type a path URI of /TimeAppWebApp/CreateEmployee, as shown in Figure 6.12. Click the OK button to close the dialog.
20. Click OK on the Create Servlet dialog to finish the addition of the servlet to the "TimeAppWebApp."
21. Having configured all the items, start the application server by selecting the application server "EmployeeSys" and then click the Start button.
22. To test the deployed servlet, ensure that the Web server has been started, and then open a Web browser on http://localhost/TimeApp/ CreateEmployee, as shown in Figure 6.13.

Using XMLConfig

In addition to using the administration console to create and to configure application servers, WebSphere offers an XML configuration management tool called XMLConfig, which allows you to export a complete or a partial configuration from a node or to import configuration into a node. This tool can be used to create and to restore a configuration backup or to copy a configuration between WebSphere domains. If you have staged your servlets and EJBs on one node and now want to move them to another node, the XMLConfig tool is ideal for helping speed this process. You can use XMLConfig to export the entire configuration from a node to an XML file. Change directories to the [ROOT]\WebSphere\AppServer\bin directory and enter the following command to perform the export on the node named my_node.

```
xmlconfig -export
myNodeConfig.xml -
adminNodeName my_node
```

You must then edit the exported file to remove any items that you do not want on the new node and to change most of the tags with action5 "update" on them to action5 "create." In order to import an XML file containing a WAS configuration, type the following command:

```
xmlconfig -import
myNodeConfig.xml -
adminNodeName my_node
```

The XML schema that the XML configuration utility uses is quite simple. Its structure mirrors that of the topology view in the administration console. For instance, consider the following minimal XML configuration file template:

```
<websphere-sa-config>
<node name="nodename"
action={"create" | "locate"
| "update}>
<application-server name =
"name"
action =
{"create"|"locate"|"update"}
```

```
>
...insert the parts of the
application server here...
</application-server>
</node>
</websphere-sa-config>
```

If you wanted to update an application server with a new value—for instance, to change its trace-output file—you would simply insert the appropriate tags into the minimal file. To change the trace-output file, you would insert the tag <trace-output>(ROOT)\mytrace.trc</trace-output> between the <application-server...> and the </application-server> tags. If you wanted to change the definition of a servlet, you would need to also include the tags defining the servlet engine and Web application, and so on.

The CD-ROM that accompanies this book contains a sample XML file (employee_sys.xml) that can be used to configure the "EmployeeSys" application server in the same way that we configured it using the administrator's console. You would configure the new application server by changing the node name in the XML file to match your node name and then running the xmlconfig import as shown previously. You can learn more about how to use the XML configuration tool by reading the product documentation and by browsing through this example output file.

Configuration Issues

Earlier, we went through the steps of setting up an application server with an EJB container, a Web application, and a servlet. In this section, we will present some best practices for configuring your application servers and Web applications.

APPLICATION SERVER VERSUS WEB APPLICATION

To understand the best practices for using application servers and Web applications, you need to consider a few more facts about Web applications. Each application server contains a single JVM, whereas a Web application is contained within an application server and shares the JVM with other Web applications. However, each Web application has its own class path and document root because a specialized class-loader class enforces isolation between the Web applications. As such, servlets within one Web application cannot directly invoke or directly utilize public static variables from a servlet within another Web application. In effect, having multiple Web applications divides up a JVM such that each Web application acts as if it has its own JVM. The only way servlets in different Web applications can communicate with one another is through the ServletContext object. You might recall that each Web application has its own ServletContext associated with it. The Servlet API allows a servlet to "look up" other servlet contexts,[7] thus allowing communication with servlets in other Web applications.

Earlier, you learned that a Web application is a logical grouping of servlets that share a common root URI. Each Web application also defines a class path; what has not been discussed so far is the fact that each Web application within an application server can be independently "restarted," or made to reload its servlets from the files on the class path. This can happen in two ways. First, a feature called autoreload can be set on each Web application. If autoreload is enabled, whenever a file changes on the class path for that Web application, the application server will dump all the current servlet instances and any other objects that were loaded with the servlets and reload the servlets from the new files on the class path. Second, an administrator can force a reload of a Web application by choosing the Restart Web App menu option from the menu in the topology tree.

This ability to force a reload of a particular Web application makes it possible to change the files related to a particular Web application while leaving running and unaffected all the servlets in the other Web applications within the application server. Thus, you can administer servlets and their associated classes, such as controllers and domain classes, on an individual Web application basis. The only limitation to this administration ability is that you cannot add or remove named servlets from the Web application

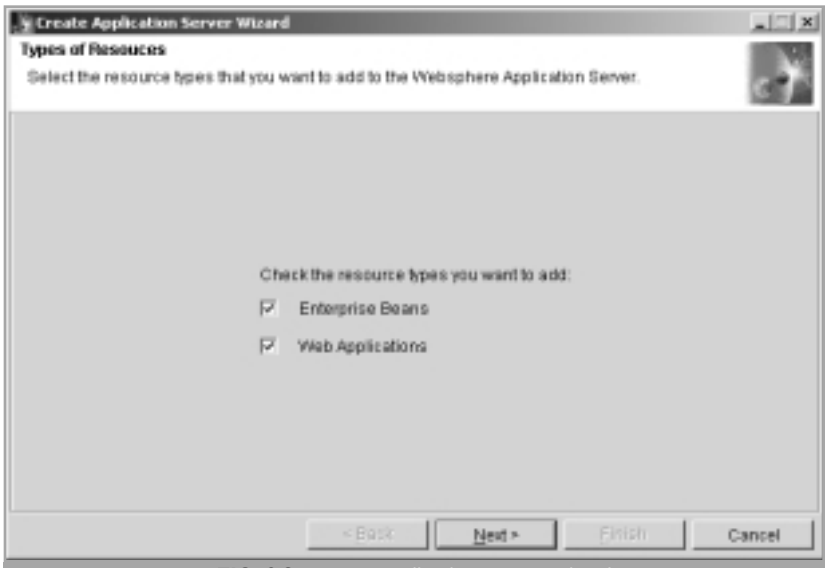


FIG. 6.8: Create Application Server Wizard

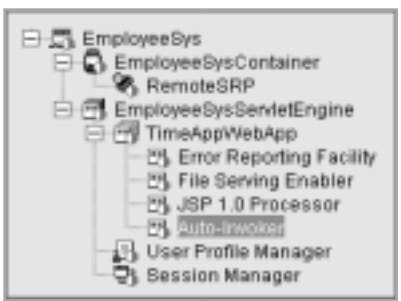


FIG. 6.9: Expanded View of the "EmployeeSys" Application Server

7. This lookup is through the "ServletContext.getContext(String uri)" method, which returns a Servlet Context for a particular root URI.

unless you shut down the entire application server. These facts are key to making the decision between choosing to create a Web application or an application server.

For example, let's say that we have two logically independent sets of servlets that run within the same intranet, although the same notion applies to different applications running within an Internet portal or application service provider (ASP). We will assume that one Web application contains our time card application—"TimeApp," which was built for the human resources department of our fictional company. We can also assume that another set of servlets was developed for the sales department representing an application for setting and monitoring sales quotas: "QuotaApp." Now let's say that both applications have relatively low utilization rates—no more than five concurrent users at a time total—well within the processor utilization capabilities of a single-processor Windows NT or UNIX server.

If we create two application servers, one for each set of servlets, we will need to either deploy the two application servers on different nodes or equip our server with a substantial amount of RAM to run the two servers simultaneously. The reason is that each application server has a substantial memory overhead. An empty application server occupies between 30MB and 60MB of RAM—just the overhead to create the JVM and to load the base WebSphere classes. If you add more application classes, this figure will grow. However, adding a new Web application to an existing application server will not substantially increase its base memory size. So, to avoid the overhead of a new JVM, you should add a new Web application instead of creating a new application server. Your applications can still remain largely independent of one another through the Web application class path feature, but you will not have the unnecessary extra overhead.

Finally, you need to understand the autoreload feature. Autoreload is a computationally intensive process; WebSphere has a thread for each Web application that will "wake up" every few seconds—set in the reload interval—and scan the class path for changes. Although this is fine for servers with low utilization or that are in a test or quality assurance (QA) environment, it becomes an unacceptable amount of "wasted cycles" when in a higher transaction-volume environment. Therefore, we recom-

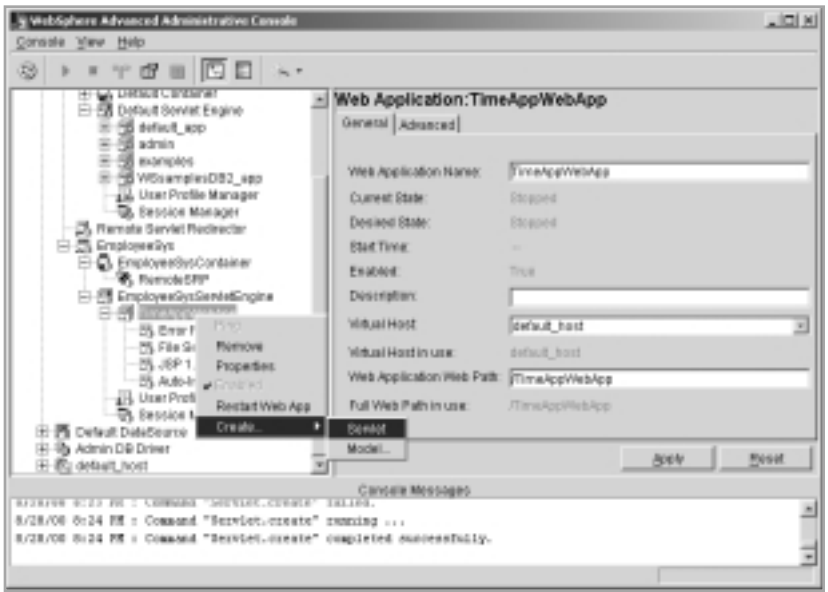


FIG. 6.10: Adding a New Servlet

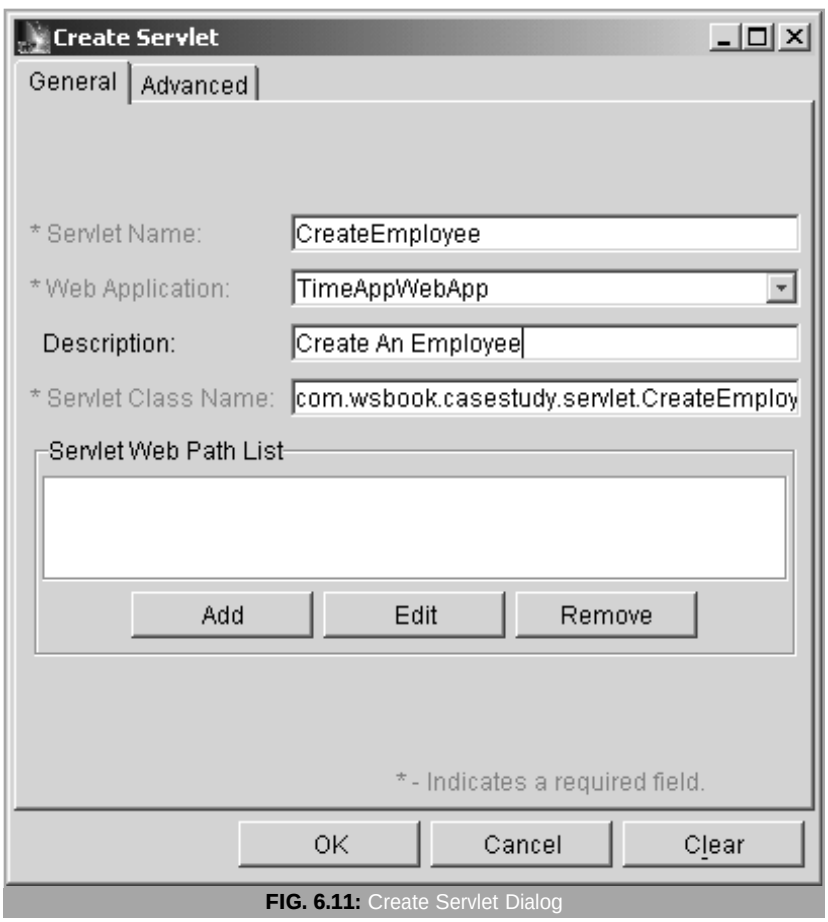


FIG. 6.11: Create Servlet Dialog

BOOK EXCERPT

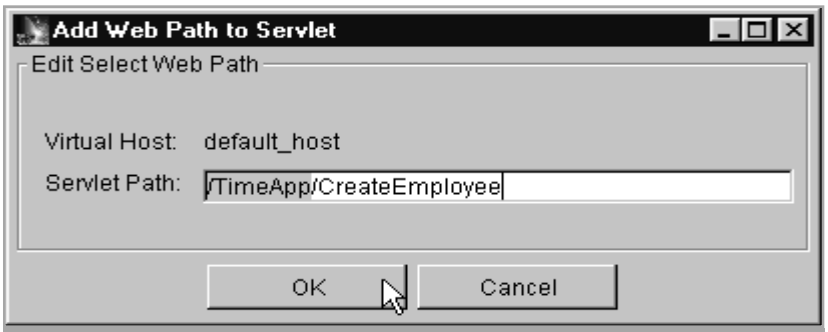


FIG. 6.12: Add Web Path to Servlet

FIG. 6.13: Testing the "CreateEmployee" Servlet

mend that you turn off the autoreload feature for production WebSphere servers in all but the least-active server environments.

Some advantages of using a Web application are as follows.

- Servlets within the same application server can use the RequestDispatcher object to forward (transfer control to) or to include the output of other servlets in the same application server.
- A running Web application can be restarted apart from restarting the application server so as to pick up configuration changes.
- Less system resources are needed, as only one JVM is needed for several Web applications.

Some disadvantages of using Web applications are as follows.

- In order to add servlets or to delete servlets from the Web application, the entire application server must be restarted.
- Less isolation between "applications" exists.

When should you use one rather than the other? The general rule of thumb is to use one application server per set of related applications and to apply Web applications as a way to segment the application into subapplications within a larger application. We would caution you to not group several unrelated projects with different delivery schedules into one application server, as you will inevitably force the other Web applications to have to restart when new servlets are added.

"INVOKER" SERVLET

As discussed earlier, the "Invoker" servlet can find other servlets by their Java class names and then load and/or invoke them on request. Generally, this servlet should be used only during testing and staging. Using "Invoker" can open up security holes. When "Invoker" is used, the HTML contains the class name of the servlet. As such, the user has more information than is necessary for calling a servlet. This small bit of information might lead an unscrupulous user to attempt to break into your system by using that information. Using "Invoker" can also decrease performance. In order for it to find a servlet by its class name, "Invoker" must use the Java Reflection API to find and to load the servlet. This API is generally slower than using a direct reference in Java. Finally, using "Invoker" limits how you can pass parameters to your servlet. You can only supply initialization parameters to a servlet that has been defined in the administrator's console.

Summary and Preview

In this chapter, you have seen how the WASAE can be used to deploy both servlets and EJBs. You learned about how WASAE implements the containers in the J2EE architecture, how to verify a WASAE installation, how to configure an application server in WASAE by using the administrator's console and the XMLConfig tool, and some best practices for configuration. In the next chapter, we look at WebSphere Studio and how it can be used to manage and deploy a Web application's resources.

SIMPLEX KNOWLEDGE COMPANY

WWW.SKC.COM

Microsoft "Missing the Point," Says IBM Program Director

IBM Responds to Microsoft's Jabs at WebSphere 4.0

In an exclusive to the *WebSphere Developer's Journal* News Desk, Stefan Van Overtveldt, program director, WebSphere Technical Marketing, IBM, commented on the IBM vs Microsoft debate that's been raging over the superiority of their respective platforms for creating Web services. He holds that Microsoft's original white paper belittling WebSphere 4.0 was fatally flawed from the start due to its premise, which, in his words, "is missing the point." Before Van Overtveldt's complete response, let's look at highlights of the verbal battle:

Microsoft launched the initial salvo with a white paper that compared the creation of Web services (using the PetStore.com scenario) using Visual Studio.NET versus IBM WebSphere v4.0. To support their claim that .NET has a significant advantage over WebSphere, Microsoft hired an independent consulting firm to develop a Web service with each platform. According to Microsoft, the results of this benchmarking exercise proved .NET the winner in developing Web services.

IBM responded with its WebSphere competitive review, (see www-3.ibm.com/software/info1/websphere/news/ibmnews/compreview4.jsp?S_TACT=101CMW13&S_CMP=campaign) calling Microsoft's white paper "misleading," and firmly stating that "there is no doubt that WebSphere is the superior platform for developing Web services." IBM pointed out that Microsoft's study used the IBM Web Services Toolkit, when the Web service should have more appropriately been built using the new WebSphere Studio Application Developer tool. IBM also said Microsoft overstated by nearly six hours the amount of time needed to create the service with IBM WebSphere. Finally, IBM said that .NET needed 106 lines of handcrafted code compared to 1 line in WebSphere Studio, and that the total cost for constructing the Web service was lower with IBM.

Microsoft's "Response to IBM" (at www.gotdotnet.com/team/compare/ibmrespond.aspx) minced no words: "IBM is attempting to mislead customers," it stated in a point-by-point comparison of IBM's claims and Microsoft's positions on issues including cost of deploying the service, the number of lines of code needed to build and consume the Web service, the requirement of BizTalk server, the standards adhered to, and .NET's ability to work in mixed environments. Microsoft did concede that IBM's new WebSphere Studio Application Developer tool improves IBM's support for building Web services, and Microsoft as a result issued a completely updated white paper comparing its product with that version (see www.gotdotnet.com/team/compare/webservicecompare.aspx). Microsoft has more background on its claims for the superiority of Visual .NET at http://msdn.microsoft.com/net/compare/default.asp.

For IBM's counter-reply, our News Desk spoke with Stefan Van Overtveldt...

Stefan Van Overtveldt: The premise of starting off with saying, "Let me show you how much more productive our development tools are by recreating the PetStore.com application in C# and with Visual Studio .NET," is beside the point. PetStore.com is an application that's been written to allow J2EE application vendors to test if all of the J2EE APIs are actually present and functioning well in their J2EE application server. That's the only objective that PetStore.com ever had. This being said, because it does go out and test all of those different APIs, this is an application that is not well written at all. It's not well written for performance. It's not well written for security. It is just a test. Taking this application and pretending that it's a real live application, and that the customers can draw conclusions with regards to productivity, performance, etc., is missing the point about what this application was intended to be.

When you look at trying to enable the application as a Web service not just from a development perspective but also from a deployment perspective (because, by the way, customers do want to deploy these Web servic-

es), we are still confident that we have tremendous productivity gains over any other application developer in the industry. If you look at what Web services are really all about, they're not just about just taking an application and making it available through an XML SOAP interface or to a WSDL wrapper, etc. You have to look into what is this going to do towards your entire infrastructure, what's the impact of opening up an application to the outside world, which is basically what you're doing. The impact is that you need to put in stronger security mechanisms. You need to be sure you can handle the workload. You need to be sure that you can quickly leverage these Web services as part of existing applications or expose existing applications as Web services. It's a much bigger picture than just taking an application and publishing it out as Web services.

You can debate different ways of doing this, but there's only one company right now that offers a complete Web services infrastructure, and that's IBM. We have Web services supported in our application servers. We have it in our development tools. We have a Web services infrastructure with regards to private UDDI gateways, for example, or private UDDI registries and UDDI gateway functionality. We have ways of managing Web services in a secure environment with a product like Tivoli Manager for Web services. We even have a number of technologies on the table that customers can use to take existing applications, not just stuff that was newly developed in C# or J2EE, but existing applications, applications that they've been using for years. Things like their SAP ERP systems, database applications, transactional applications running on IBM mainframe platforms, etc., and make those available very rapidly as Web services, again in the same managed and controlled environment. That is something no other company can offer.

ND: The use of the PetStore setup did seem problematic.

SVO: We can argue about which company or which tool is more productive in this scenario for years to come. Truth is, it's beside the point. It's not a representative application. It does not make a lot of sense to go out and recreate an application in another language and see if it performs better. If I were to rewrite this application, and believe me I'm not a good programmer, it would probably perform better.

ND: One issue aside from performance that Microsoft was slamming IBM on was cost. IBM's first response to Microsoft said a 4-server deployment using Microsoft .NET would cost about $399,996. Microsoft's response to that was IBM "is just wrong. IBM is attempting to mislead customers." Microsoft went on to say that WebSphere 4.0 costs over 15 times more that Microsoft .NET for this typical clustered scenario.

SVO: Well, if nobody is accessing the cluster, they're probably right. The WebSphere licensing is based on buying an application server, and we have a per-processor licensing scheme, and you can put those applications, it's $12,000 list price per processor, in a cluster. So, if you have a 4-node cluster, for example, that would cost you again at list price, $48,000. Throw in 10 developer seats around and you're looking at a total solution of around $80,000. But that is it. Whether that solution is serving one concurrent user or it's serving 100,000 concurrent users, the price remains the same. We don't have the notion of access licensing. If you count how the developer license works, if you count on actually putting those applications in production, not just showing a demo of it, you have hundreds, thousands of other applications out on the Internet accessing that Web service, then the Microsoft solution is a lot more expensive. It's just because it's a completely different way of counting licenses.

ND: That's what's accounting for the cost differential: how they're approaching their cost basis?

SVO: This is the same thing that goes back to Pet Store. Pet Store is a lab type of application, something you would never put in production. And that has a number of consequences. If you set up this type of environment, in a lab environment, to calculate what it costs you, maybe Microsoft has a point. Maybe they don't. But if you put it into a production environment where you have to start accounting for a number of concurrent accesses to these servers, we are very sure that our solution is much more cost-effective.

ND: The last point Microsoft emphasized in their most recent response had been regarding code. Microsoft said that to create a PetStore Web service, IBM WebSphere Studio Application Developer required 82 handcrafted lines of code and Visual Studio .NET required 48, about half; and to create simple client to consume Pet Store Web service, IBM required 49 lines of handcrafted code; Visual Studio .NET, 24. Now, is that still a function of Pet Store's being atypical of a real-life application code-writing situation?

SVO: Again, I did not personally run those tests. The one point I can make is that most of the ability we have in Web Studio Application Developer is to take an existing J2EE application and render it as Web services fully automatic. Now is there some hand coding involved and is it 48/24 lines of code, double in our scenario? I don't know. What I do see is that taking this application and making it available as Web services is, again, only part of what you need to do. Because you need to link this application to a security environment, which Pet Store does not do, by the way. You need to link this application to a management environment, which Pet Store does not do. With WebSphere Studio Application Developer we can do all of those things pretty much in an automated fashion. You may need to write some line of code here and there, but that is a much more realistic scenario to look at. I am absolutely convinced that if you look at what are the real overall requirements that you actually to put a Web service in production our development tools are much more effective and much more productive than any other tool out there.

The larger question is, which platform are customers choosing to deploy Web services? Giga Information Group just issued a report saying that J2EE platforms are the big winners overall among early adopters of Web services technology and the most important to Web services strategy. When it comes to companies that are actually evaluating and deploying this, Giga says IBM WebSphere is the clear favorite over Microsoft. In the end, it is the market that is deciding, and according to Giga, customers favor WebSphere over .NET.

Microsoft Replies to IBM...

Gregory Leake, Group Product Manager, Microsoft Corporation, offered this response:

I am writing about your article titled "IBM Responds to Latest Microsoft Jabs at WebSphere 4.0." In this article, you interview IBM's Stefan Van Overtveldt, program director, WebSphere Technical Marketing.

In the interest of facts and truth, I would like to correct Mr. Van Overtveldt on a couple of points, and present a Microsoft response to his interview and the article.

First, I want to point out that the .NET Pet Shop comparison to the Java Pet Store is separate from the Web Services comparison between .NET and IBM WebSphere 4.0. Both comparisons can be found at www.gotdotnet.com/team/compare. The article and the interview discuss both comparisons as if they are the same.

So let's separate them and look at the bulk of what Mr. Van Overtveldt says about each.

1. With regard to the .NET Pet Shop comparison, he says that:

"PetStore.com is an application that has been written to allow J2EE application vendors to test if all of the J2EE APIs are actually present and functioning well in their J2EE application server. That's the only objective that PetStore.com ever had…. this is an application that is not well written at all. It's not well written for performance. It's not well written for security…. Taking this application and pretending that it's a real live application, and that the customers can draw conclusions with regards to productivity, performance, etc., is missing the point about what this application was intended to be."

This is wrong. The Java Pet Store (Petstore.com) is held up by Sun Microsystems as a primary blueprint application for J2EE that illustrates "best practice architecture" and "best coding practices…for enterprise applications." It has an entire Sun blueprint Web site dedicated to it as a sample application for customers to follow. Sun has even published a Sun Blueprint-series book all about the Pet Store as a design pattern for enterprise applications, and has been telling enterprise developers to follow it for building scalable, reliable applications since Java One in May 2001. Furthermore, IBM appeared on stage with Sun and other J2EE vendors to endorse the Pet Store application at Java One in May 2001, and even demonstrated it running in WebSphere 4.0 in front of thousands of developers as a best practice enterprise application. And furthermore, they ship it in the IBM WebSphere 4.0 product as a sample application for developers to follow. So it is very disingenuous for IBM to come out now and say it is a bad application and should not be used to compare with .NET. They are all of a sudden singing a very different tune based on the release of the .NET Pet Shop. Sun has also now come out and said that the design pattern is valid, but it is not designed to be high performance. In response, we find it a highly questionable customer practice to publish a "best practice enterprise design pattern" and not ensure it will result in high performance applications. In short, if this is the case, shame on Sun for publishing it and promoting it as an enterprise design pattern to begin with. They should either rewrite it, pull it from the Web and their blueprint series published books, or stand behind it. As for the benchmark of the Pet Store application, I would like to point out that the benchmark was originally conducted by Oracle as part of their Java Performance Challenge. So in this way a major

In response, Microsoft claimed IBM's spokesman "is misleading customers" about .NET pricing and calls for an "independent shootout" to resolve .NET vs WebSphere performance comparisons

Microsoft group product manager Greg Leake asserted that his opposite number at IBM was "misinformed" and that IBM was "misrepresenting" the licensing costs applied by Microsoft when claiming that the client access licenses required for Windows 2000 Server drive up the cost for .NET as more users connect. "IBM WebSphere 4.0 costs 15 times more than Microsoft .NET in the example analyzed," maintains Leake.

He also refuted the position taken by IBM's Stefan Van Overtveldt, program director, WebSphere Technical Marketing, that IBM is missing the point when using the Petstore.com application to make comparisons between J2EE and .NET.

On the contrary, says Leake, "The Java Pet Store (Petstore.com) is held up by Sun Microsystems as a primary blueprint application for J2EE that illustrates 'best practice architecture' and 'best coding practices…for enterprise applications'… Furthermore, IBM appeared on stage with Sun and other J2EE vendors to endorse the Pet Store application at Java One in May 2001, and even demonstrated it running in WebSphere 4.0 in front of thousands of developers as a best practice enterprise application."

It's "very disingenuous," Leake insisted, "for IBM to come out now and say it is a bad application and should not be used to compare with .NET." "We are very happy to meet IBM, Oracle, or other J2EE vendor in an independent shootout for verification of the performance results," concludes Leake.

J2EE vendor invited the comparison. In the end, I think the comparison is quite valid, and the competition is healthy for customers and the various vendors involved. Right now, the .NET Pet Shop remains uncontested as far as published versions of the Pet Store go, with one quarter the amount of code required to build vs. the Java version, and offering 28 times better performance and over 8 times better scalability. We are very happy to meet IBM, Oracle, or other J2EE vendors in an independent shootout for verification of the performance results.

2. With regard to our .NET vs IBM Web Services comparison, the bulk of the interview seems to focus on the license cost comparison. Here again, Mr. Van Overtveldt is misleading customers, since he claims that our cost comparison does not take into account client access fees associated with the deployment. I would like to point out that we fully take into account client access licenses in the cost comparison, and that he is simply misinformed on this topic. Mr. Van Overtveldt claims a 4-server WebSphere deployment would cost a fixed amount no matter how many clients access the WebSphere server. He states that the MS cost, however, may be lower out of the gate, but would be higher and grow depending on the number of clients accessing the site. This, he claims, is because the client access licenses required for Windows 2000 Server drive up the cost for .NET as more users connect. This is wrong. In fact, we have included a Windows 2000 Server Internet Connector License in our cost calculation, which includes *unlimited* authenticated client access to the .NET Web Service in question. So we fully stand behind our cost comparison, no matter how many users are connected and using the Web Service. For a 4-server deployment with 8 CPUs per server, IBM WebSphere 4.0 would cost $12,000 per CPU or a total of $384,000. Microsoft .NET would cost $3,999 (W2K Advanced Server) + $1,999 (Internet Connector License) = $5,998 per server for a total of $23,992, no matter how many clients use the W2K site. Mr. Van Overtveldt should make a point of understanding the MS licensing cost before misrepresenting it, and IBM should correct their public document because it is wrong. IBM WebSphere 4.0 costs 15 times more than Microsoft .NET in the example analyzed, which is very typical of a real customer deployment configuration in the enterprise.

IBM's Van Overtveldt Comments on WebSphere vs .NET Debate... "Let the Market Decide"

On January 15, 2002, responding to the open letter from Microsoft"s Greg Leake, IBM's Stefan Van Overtveldt commented:

IBM and Microsoft can fight about benchmarks forever, but what matters most is what customers and developers are doing. .NET only supports Windows and other Microsoft technologies, while IBM offers tools like Eclipse and WebSphere Studio that are truly cross-platform and open standards-based. At the end of the day, it is obviously going to be hard to prove in a discussion like this whether one tool or another is more productive. The people that will make that call are the developers out there writing the code, and developers would rather have a choice of platform and vendors.

We believe that it's all about real-world scenarios, not about just one application like petstore.com. Developers and organizations will pick the environment and tools that provide the lowest total cost of ownership, while matching their particular systems requirements. Most companies don't just rely on Windows but have to work with heterogeneous systems like Unix, mainframes, etc. - which is why IBM's tools are ideal. Let the market decide.

EYT Opens WebSphere Innovation Center

(Parsippany, NJ) – EYT, formerly Ernst & Young Technologies, an IBM Business Partner, and LInXLogix, formerly EYT eSolutions Center, a certified IBM solution provider, have announced the opening of the newest IBM WebSphere Innovation Center (WIC). This facility, housed in EYT's Parsippany, NJ, facility, will be jointly managed by EYT and LInXLogix.

This WIC is designed to provide a number of important services, including WebSphere software-based "test drives" on a variety of hardware platforms and operating environments, and proof-of-concept support and technical briefings. WebSphere e-business infrastructure software – known as middleware – enables companies to develop, deploy, and integrate next-generation e-business applications. A key focus of this WIC will be highly secure customer support, deal brokering, and communications solutions, all delivered by LinXLogix. This enables the document sender to control data security during electronic transmissions. www.eyt.com, www.linxlogix.com

IBM Presents Braegen Group with Customer Satisfaction Award

(Toronto) – IBM Canada's president, Ed Kilroy, presented Braegen Group, Inc., with its fifth consecutive customer satisfaction award. The prestigious Thomas J. Watson Gold Award, named after the founder of IBM, was presented to Braegen, a leading provider of e-business solutions, for its outstanding commitment to customer service and satisfaction.The award recognizes business partners who have made a superior commitment toward integrating business excellence and customer satisfaction into their planning, operations, and processes. www.braegen.com

Mellanox Demos IBM DB2 and WebSphere on InfiniBand Server Blade Cluster

(San Jose, CA & Yokneam, Israel) – Mellanox Technologies, Ltd., the leading supplier of InfiniBand semiconductors, has announced the first demonstration of an InfiniBand server blade cluster running IBM DB2 Universal Database and WebSphere Application Server. The demonstration of true three-tier data center applications includes DB2, WebSphere, and Ethernet clients connected using standard sockets interfaces. The data center platform is based on Mellanox's Nitro InfiniBand server blade reference chassis and Voltaire's nVigor InfiniBand switch/router.

The high-performance DB2 Universal Database and WebSphere Application Server clusters run on InfiniBand server blades entirely housed in a single 4U chassis integrating InfiniBand switching and up to 16 server or I/O blades. The sockets-level connections provide a standard software interface to applications so no modifications are required to the IBM DB2 or WebSphere packages. Providing transparent client connectivity between InfiniBand clusters and Ethernet clients allows data center managers to leverage the performance benefits of InfiniBand technology without having to abandon current technology investments.
www.mellanox.com

Virtio Announces Virtual Platform Support for WebSphere Everyplace

(Campbell, CA) – Virtio Corporation, the creator of virtual platforms, has announced support for IBM's open standards–based WebSphere Everyplace Embedded Edition running on Virtio's VPXS Virtual Platform technology. VPXS is a software model of the Intel PXA250 Applications Processor Development Board and supports a variety of operating systems and tool chains.

IBM's WebSphere Everyplace Embedded is an integrated device software platform for both native and embedded Java applications targeted at extending e-business to a wide range of devices. It's packaged with a POSIX-compliant Real Time Operating System (RTOS), an IBM-embedded Java Virtual Machine (JVM), and the OSGi-compliant Service Management Framework, and includes an Eclipse-based Integrated Development Environment. www.virtio.com

NEON Systems and Idera Ally to Accelerate Java Application Development

(Sugar Land, TX) – NEON Systems Inc., a provider of software to "Liberate the Mainframe," has formed an alliance with Idera, a developer of data management, migration, and integration tools to provide increased productivity and reduced development costs for application developers working with mainframe data sources.

The combination of NEON Systems' ShadowConnect technology with Idera's JSync Solutions Toolbox provides application developers with a single tool for visibility into both legacy and distributed systems data sources, resulting in shorter development time for J2EE-compliant enterprise applications that run on major application servers like IBM WebSphere. In addition, the combined ShadowConnect/JSync solution allows developers to migrate or convert data between most non-relational and relational data sources.
www.idera.com, www.neonsys.com

Versata and ILOG Collaborate on Business Logic Initiative

(Oakland, CA) – Versata Inc., a provider of software and services that automate the business logic and processes that power enterprise applications, has announced an agreement with ILOG, a provider of software components, to collaborate to offer the industry's first end-to-end business rules solution.

Under the terms of the agreement, Versata will develop a connector between the Versata Logic Server and ILOG JRules, and in turn, ILOG will develop a connector between their ILOG JViews for Workflow product and the Versata Logic Server.

The connector between ILOG JRules and the Versata Logic Server will provide customers with a direct link to execute decision rules managed by ILOG JRules from the Versata Logic Server. From a running transaction or process, the Versata Logic Server will call out to the ILOG Rule Engine, receive a response, and then use this response in the transaction or process. This blend of business logic will enable customers to build highly transactional, performance-critical systems that integrate decision or deduction rules that can be changed dynamically by business users. www.versata.com, www.ilog.com

eclipsing the Competition

IBM's advantages are too great to overcome

BY JANE EDWARDS

Recently, several companies announced upgrades to their developer sites. Let's look at IBM's ongoing efforts in this area.

IBM has a development community of 1.6 million registered users. In the two years since its inception, developerWorks has offered multiple online and offline resources, achieved 85,000 downloads of Web services technologies from IBM alphaWorks, and distributed over 55,000 WebSphere CDs.

IBM is an established technology leader and innovator, already pushing into areas such as grid computing and Web services. Since November, IBM has been shipping WebSphere Studio, a set of easy-to-use visual tools for building Web services applications. Customers are building Web services applications with WebSphere Studio and have the additional advantage of support for the independent open-source community, eclipse (www.eclipse.org).

eclipse

eclipse is the next step in IBM's open-source strategy for the future of Web software development and Web services. It provides an environment that can address the entire application development life cycle with best-of-breed products. IBM has donated $40 million of software to the eclipse open-source community. eclipse technology, supported by some 150 ISVs around the world, enables developers to easily integrate and use a variety of different tools together as if they were one, greatly enhancing developer productivity. It solves the problem of integration within the IT industry today, as most development tools are incompatible with each other.

eclipse is also the newest phase of IBM's open approach to software for e-business and Web services. IBM has already successfully executed on several previous steps, including opening its servers with its $1 billion support of Linux; opening its middleware by building WebSphere on Apache; and opening its integration software by supporting it with the broadest range of Web services standards (XML, UDDI, SOAP, etc.).

Gimme the Works

With over 130 WebSphere developers and user groups in 40 countries, IBM's WebSphere Developer Domain has a widespread community. There are now more than 600,000 active WebSphere developers worldwide, nearly three times the nearest competitor.

The WebSphere Developer Domain includes articles, tutorials, frequently asked questions, and best practices that are refreshed constantly. It also features a monthly online technical journal highlighting WebSphere Application Developer, tools, application servers, specialty products, WebSphere Studio tooling, portals, load balancing, and programming to the WebSphere Application Server.

In addition, WebSphere Developer Domain grants immediate access to product updates and self-support for WebSphere, as well as an avenue to request new features in future versions of WebSphere products.

1.6 Million Developers Can't Be Wrong

developerWorks is IBM's developer resource for open, standards-based development. Committed to providing relevant and accurate technical information by tapping into IBM and other industry leaders, developerWorks content is invaluable to developers regardless of their application development tool of choice. It offers technology and product resources related to Linux, XML, Java, Web services, and emerging technologies.

developerWorks has a vast development community with 1.6 million registered developers – over twice as many WebSphere developers as some of its competitors. IBM offers developer resources that no other company can match, including 150,000 technical professionals, 3,000 researchers in eight labs worldwide, 4,000 Java professionals, and over 110 Java products. IBM has also made a billion-dollar investment in Linux and has 1,500 Linux professionals and over 1,000 technologists dedicated to Linux. It presents a mix of content from authors within IBM and also from leading external experts.

The developerWorks team has developed an immense collection of resources, including 140 online tutorials, 250 emerging technologies, and 1,500 technical articles. Resources include custom collections of local native-language content and resources in several countries outside the U.S. – in addition to China, Japan, and Taiwan. developerWorks visitors have downloaded more than 7 million resources, including Java tools, Linux tutorials, bundles of code, and sample Web services. The weekly newsletter attracted more than 200,000 subscribers in its first year alone, and has generated reader compliments such as "it's become the default 'must read' per week for me" and "excellent, relevant content."

In the midst of an industry-wide slowdown, developerWorks has seen growth in unique visitors and site usage, and the links to developerWorks from external sites have nearly doubled in the past year. Internationally, traffic to developerWorks international sites has jumped 1,000 percent in the past 18 months, with just over half the visitors living outside North America. It's not all online either; developerWorks offers support to user groups and has a conference dedicated to developers – developerWorks Live, next running in May 2002.

The newest addition to developerWorks, developerWorks China, was launched a year ago for developers whose primary language is simplified Chinese. Since then it has grown to be the largest developer site in China, with traffic increasing 30% each month. The site now includes over 500 articles translated from developerWorks' global site and nearly 300 articles from local developers.

Too Little Too Late

IBM is delivering on its billions-of-dollars investment and years of solid experience in providing developers with the resources they need. This is why IBM's programs, Web sites, and other resources are rated as some of the best in the industry.

ABOUT THE AUTHOR...
Jane Edwards is a freelance writer based in New Jersey.

E-MAIL
websphere@sys-con.com

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

— ***An Interview with Don Ferguson***
Talking with the Father of WebSphere

— ***Creating Performance-Optimized JDBC Applications for DB2***
Thoughtful design and implementation improve performance

— ***Automating Business Logic for WebSphere Using Transaction Rules***
Freeing developers from routine infrastructure issues

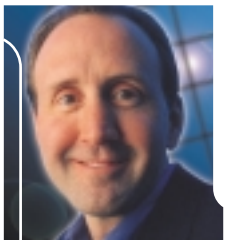
— ***Why They Buy:***
The How and Why of Selling WebSphere

— ***Automating Business Logic for WebSphere Using Transaction Rules***
Integrating the automated system with external applications

WebSphere Developer's Journal | Advertiser Index...

Real-time Business

BY MICHAEL MAAS



Time. The most precious commodity on the planet – every year there seems to be less of it available. Less time for hobbies, less time to play with the kids, less time to sleep. Many long for a simpler age, when there were fewer pressures, fewer complexities, and more leisure time. Most likely that perception stems from personal aging, not generational. In fact, studies show we now have more discretionary time than 30 years ago. Yet it doesn't matter; perception is reality. Ask almost any professional colleague what they want more of and typically the answer is time.

In business, time, or rather its inverse measurement, speed, is just as important. Hence the old adage "time is money." Many primary metrics for a corporation have an element of time embedded in them – productivity looks at how much work occurs in a given time period, factory cycle-time measures the time it takes to build a widget, accounts receivable captures the time it takes to get paid, and so on. In business, time matters.

It's important to realize that businesses operate fundamentally on repeatable processes. Developers are driven by a product-development process that (hopefully) turns market requirements into a marketable solution. Sales teams utilize a structured sales process that helps find customer opportunities and turn them into revenue. Finance teams operate processes that measure and report on the business to management and shareholders. Process, done right and without unneeded bureaucracy, is vital. Without process there'd be no business.

Time and process. Together they form the essence of improving a business. Regardless of what product is being built, what service is being provided, or what market the business serves, decreasing the time it takes to execute various processes will improve the business. This is intuitive – faster development time means first advantage and reduced costs; faster sales cycle means more sales per quarter and lower expenses; more frequent internal reports mean better decision making. It's fundamental – focus on speed of the business and good things will follow.

Real-time Business

The biggest opportunity to speed up business processes lies in decreasing the latency of information flowing into management systems and the delays that occur when information is unavailable to people. During the day field workers capture information that sits on clipboards until they return to the office to type into computers. Salesmen make unnecessary additional customer calls because on a previous call they didn't have current specs and pricing. Factories slow as just-in-time supply arrives just-too-late. Decisions are made using last month's facts, while financial systems struggle to aggregate current data. Yet we can imagine a state of nirvana for business, where every bit of data and information relevant to the business is instantly turned into knowledge for whomever needs it. That's the goal – real-time business.

The stage has been set over the past decade as e-business has been deployed in leading companies. Many firms have sophisticated back-end systems that integrate corporate applications, are based on open standards, and offer dynamic query to mere mortal users. These are often top-notch applications, but like any system, they can only be as good as the data. And more often than not, the data is much older than management would want, and will be even older by the time a report is in the hands of decision-makers. This limitation is critically important – once information gets created, it takes far too long to enter systems, be transformed into knowledge, and become available to the right people to take action.

The Impact of Wireless

It need not be that way. New technologies are now widely available, and economical for mass adoption, which promise to dramatically improve the instant availability of data. Broad deployment of wireless data terminals, location-based services, RF-tagged inventory, e-payment systems, smartcards, voice recognition, and other mobile computing technologies holds the key to eliminating the latencies associated with data input to corporate e-business systems. They also offer the promise of employees being truly able to access any information, anytime, from anywhere – which means they can act quickly on new insights being created in a real-time environment. By combining immediate data input with wherever-whenever data access, e-business will be extended to the next level and real-time business will be achieved. In the process, all aspects of a company will improve dramatically. Sales managers won't extrapolate forecasts from month-old data, foremen won't speculate on where the truckload of critical inventory is, and controllers won't need to accrue reserves for work lost in a paper shuffle and never billed. Immediate, right-then-and-there data input and access can improve quality, productivity, and profits.

Of course, significant change that touches the core of corporate process is not a trivial undertaking. Like the massive reengineering projects of the '90s, long-term commitments and clear strategies are required. But in this case the necessary infrastructure is typically already in place; what's needed is merely an extension to existing systems.

The key is middleware like IBM's Websphere product family. The extensible architecture and inherent scalability provide the basis for adding extensions to support mobile data input and output, seamlessly integrated with an existing environment. In fact, Websphere Everyplace Server already has support for location-based services, voice services, and translation for mobile devices. It's the logical place to start the journey toward making applications real-time ready.

And start soon – the future is upon us and the benefits can be had now. Field maintenance workers, mobile sales forces, and traveling executives are all critically dependent on the existing e-business infrastructure, and the success of the firm is critically linked to how well they're connected to the rest of the enterprise. Don't settle for a "batch-mode" approach. Make your corporation run interactively; make it a real-time business.

ABOUT THE AUTHOR...
Michael Maas, director of marketing for IBM's Global Wireless e-business unit,is responsible for the marketing of wireless data solutions to both enterprises and service providers.

E-MAIL
www.ibm.com/solutions/wireless