MOBILE • CACHING • MODERNIZATION

SYS-CON

www.SYS-CON.com/WebSphere

WebSphere® JOURNAL

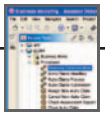
The World's Leading Independent WebSphere Resource

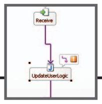
JULY 2005 VOLUME 4 ISSUE 7

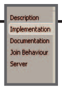
Inter-Portlet Communications

Using IBM Rational Application Developer V6.0

BY ASIM **SADDAL** PAGE 42







Parasoft SOAPtest®

Keep Your Skills Ahead of the Crowd

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

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

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

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

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

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

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

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

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

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

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

Features

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

Benefits

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

Platforms

- Windows 2000/XP
- Linux
- Solaris

Protocol Support

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

Contact Info:

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

www.parasoft.com

Movin' On Up

time to migrate to the next generation of WebSphere

Ask the Experts

Prolifics, a premier-level IBM Business Partner and 1 of 3 WebSphere Service Providers retained by IBM, specializes in WebSphere migrations.

Having conducted numerous migrations, both from earlier versions of WebSphere and from other J2EE application servers, such as BEA WebLogic, Oracle AS or Sun iPlanet, Prolifics has migration tools and a methodology that avoids unnecessary risk, time, effort and cost.

From J2EE App Servers to WebSphere

Prolifics conducts Migrations from other J2EE Application Server environments such as BEA WebLogic, Oracle AS, Sun iPlanet, etc.

From WebSphere v3.5/ v4 to v5

Prolifics migrates from earlier versions of WebSphere to IBM's latest release.

Taking You Step by Step

Prolifics offers MIGRATION ASSESSMENTS, PROOF-OF-CONCEPTS, and WEBPSHERE MENTORING to help you validate, plan, and execute a successful transition.

Prolifics

Winner of 2003 IBM BUSINESS PARTNER LEADERSHIP AWARD

116 John Street
New York, New York 10038
www.prolifics.com
crm@prolifics.com
800.675.5419

Prolifics®

IBM Business Partner

company, product and service names maybe trademarks or service marks of others.

No Theme, Just Lots of Stuff

BY ROGER **STRUKHOFF**

The idea of a "pure" application development continues to lose meaning in an enterprise IT world that continues to try to integrate legacy systems with server networks, local-area networks, increasing varieties of wireless networks and devices, and the latest new kids on the block who want to join the game.

Thus, the idea of an "open" environment seems intuitively obvious, as it has for many years. Yet "open" to a lot of people seems to mean "free," as if all cool things should be available to anyone who wants them, gratis, because…well, just because.

Large technology companies are not immune to trying to have it both ways. Most major IT vendors tout their commitment to an open-source approach, yet wish to do so under their own strict, proprietary way of doing so.

The old saying that "standards are wonderful, that's why everyone wants one," has a corollary today in that "the only correct approach is the open approach, and here's how our open approach works."

But this column is not going to be a tedious rehash of the various political arrows being slung around the enterprise IT world today. Rather, it's going to be an exciting rehash of what you will find in this issue of *WebSphere Journal*. The issue has a truly eclectic variety of articles this month, reflecting the inherent tension between the use of a rock-solid, highly sophisticated environment such as WebSphere and the more messy, sometimes bewildering demands of real-world application development.

The continuing evolution of Service-Oriented Architectures is tackled by Praveen K. Chhangani, for example.

The reality of WAS is given three different, specific looks this month. Pooja Gupta handles the complex task of outlining Memory Caching in WAS. Ade Rixon finishes off his series on Putting WAS on UNIX. And Linfeng Yu continues his series on z/OS with an article about WAS for z/OS. (Look for more z/OS articles from him in the future.)

Meanwhile, we've also addressed WebSphere Commerce Design Patterns with an examination of the topic by Bhadri Madapusi.

Two emerging topics, Mobile to Multi-Modal and Inter-Portlet Communications, are covered by Les Wilson and Asim Saddal, respectively. And we go out on a limb a bit to describe not only WebSphere, but potentially alternative approaches to legacy integration with an article by Mike Marlowe.

Do you want to read about BPEL while you're at it? You can, by viewing BJ Grau's article. And the general topic of IBM Software Integration is handled by Reid Gerson in this issue.

From the specific to the general, from "pure" WebSphere to integrated approaches to alternative approaches. It's all here, put together by a truly international cast of characters. There is no unifying theme in this issue of *WebSphere Journal*, and that's the point we are making. Political arguments tend to oversimplify matters, especially in this sound-bite age. They tend to latch onto a single aspect of an overall approach or philosophy, cast that aspect as 100% right and anyone who varies from it, even slightly, as 100% wrong.

We all know the world is a more complex place than that. And we know that the world of enterprise IT application development and management is hardly a simple world. There are no absolutes, except the absolute necessity to get too much work done in too little time under too strict parameters. Isn't life wonderful?

Happy reading, and please send your praise and other comments to me at roger@sys-con.com. We value your input even more than our own!

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

WebSphere® JOURNAL

ADVISORY BOARD

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

EDITOR-IN-CHIEF	ROGER STRUKHOFF
CONTRIBUTING EDITOR	PATTI MARTIN
PRODUCT REVIEW EDITOR	JAY JOHNSON
EDITOR	SETA PAPAZIAN
EDITOR	NANCY VALENTINE

WRITERS IN THIS ISSUE

Praveen K. Chhangani, Reid Gerson, BJ Grau, Pooja Gupta, Sherman Lee, Bhadri Madapusi, Mike Marlowe, Ade Rixon, Asim Saddal, Roger Strukhoff, Les Wilson, Linfeng Yu

SUBSCRIPTIONS

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

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

EDITORIAL OFFICES

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

WORLDWIDE NEWSSTAND DISTRIBUTION
CURTIS CIRCULATION COMPANY, NEW MILFORD, NJ

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

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

SYS-CON MEDIA

Application: Operations Monitor

WebSphere Commerce Design Patterns – Part 2

Lending a uniform structure to design patterns

BY BHADRI MADAPUSI

Bhadri Madapusi is a software developer working in the IBM Toronto Lab's Electronic Commerce Division. He earned his MSc in information systems from BITS, India, and his MSc in computer science from Queens University, Canada. Bhadri's interests include software modeling, design patterns, and data management. bomadapusi75@yahoo.com

IBM's WebSphere Commerce product is a platform for developing and deploying value chain solutions from a consumer-centric online sales channel to a completely integrated, multi-tier demand chain. A variety of design patterns were used to develop the WebSphere Commerce framework. By understanding and using these high-level design patterns, you can write WebSphere Commerce apps that adhere to the WebSphere Commerce framework. Furthermore, these patterns will help you customize the out-of-the-box capabilities of WebSphere Commerce.

WebSphere Commerce design patterns fall into two main categories: command design patterns and display design patterns. In the first category are the Controller Command pattern, Task Command pattern, and Data Bean Command pattern. In the second are the Smart Data Bean pattern and Command Data Bean pattern.

In this two-part article, I discuss the design patterns in terms of the template described in the book *Design Patterns – Elements of Reusable Object-Oriented Software* by Erich Gamma et al. This template lends a uniform structure to design patterns, making them easier to learn, compare, and use. In the first part, published in the June issue, I covered the controller command and task command. In this part, I'll cover Data Bean command and Display pattern.

Data Bean Command

INTENT: Avoid coupling a data object with the logic used to process and populate the data into the data object by moving this logic into a separate object. Modify the data fetching, processing, and population tasks independent of the actual data object.

MOTIVATION: WebSphere Commerce recommends using data beans (see the display pattern category) to display the data to the client. These data beans are data objects with properties, their getters and setters (essentially Java beans). The logic to fetch, process, and populate data must not be embedded in the data objects. Embedding this logic will strongly couple the implementation details with the data object. If your business needs change over time, this strong coupling will restrict you from modifying how your implementation fetches, processes, and populates data. If you have strong coupling and want to modify the implementation, you may have to rewrite your data object and change the client to this new data object.

Searching a product catalog is an important commerce feature. To display the results of your catalog search, you may use a data object that whose properties are defined for search criteria and search output. If your search imple-

"By using high-level design patterns, you can write WebSphere Commerce apps that adhere to the WebSphere Commerce framework and help customize the out-of-the-box capabilities of WebSphere Commerce"

mentation is coupled with the data object, you'll be forced to modify the data object or create a new data object when your search algorithm changes. This can be avoided by having the implementation in a separate object that can be varied dynamically with your business environment.

Suppose SearchDataBean is the data object that contains the search criteria and search output. SearchDataBean won't implement the search logic; instead the logic will be implemented by SearchDataBeanCmdImpl. By changing the implementation class, we can modify the search logic. SearchDataBean maintains a reference to the interface that will be implemented by the implementation class. DataBeanManager gets this interface reference from SearchDataBean and instantiates an appropriate implementation with the help of the CommandFactory. Furthermore, DataBeanManager will pass in a reference of the data bean to the implementation class so the implementer can get the input criteria and set the result data in the bean.

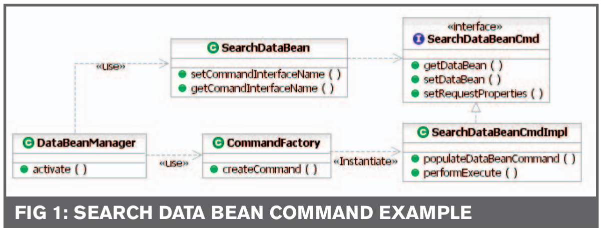


FIG 1: SEARCH DATA BEAN COMMAND EXAMPLE

APPLICABILITY: Use this pattern when all of the following conditions apply:

a. You want to avoid coupling the data object and the algorithm used to fetch, process, and populate the data object.

b. You anticipate a change to the population business logic.

c. The data population business logic may vary with store type.

STRUCTURE:

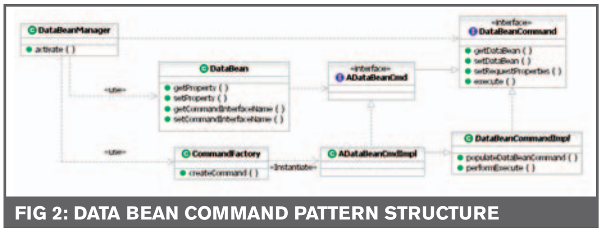
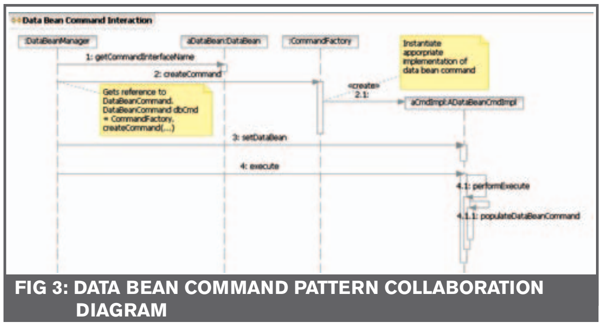


FIG 2: DATA BEAN COMMAND PATTERN STRUCTURE


Data Bean Command Interaction
:DataBeanManager
aDataBean:DataBean
:CommandFactory
Instantiate apporpriate implementation of data bean command
1: getCommandInterfaceName
2: createCommand
Gets reference to DataBeanCommand. DataBeanCommand dbCmd = CommandFactory. createCommand(....)
«create»
2.1:
aCmdImpl:ADataBeanCmdImpl
3: setDataBean
4: execute
4.1: performExecute
4.1.1: populateDataBeanCommand
FIG 3: DATA BEAN COMMAND PATTERN COLLABORATION DIAGRAM

PARTICIPANTS:

DataBeanCommand: A Java interface that declares the methods for the following tasks: to execute the logic that populates data beans, to set input properties that will be used by population logic, and to set/get the data bean reference. This interface is provided by the WebSphere Commerce Server.

DataBeanCommandImpl: An abstract class that provides the default implementation to the methods declared in DataBeanCommand. This interface is provided by the WebSphere Commerce Server.

ADataBeanCmd: A Java interface that declares additional methods required in fetch, process and populate operation. It also provides the name of the default implementation class that implements the methods that are defined by this interface and the business task logic. The CommandFactory instantiates the default class if it fails to find an appropriate implementation class in the command registry.

ADataBeanCmdImpl: Provides implementation for data fetching, processing, and population.

CommandFactory: Instantiates an appropriate implementation of ADataBeanCmd based on store type. The appropriate implementation class is defined in the command registry.

DataBean: The data object that's populated using the Data Bean Command pattern. This contains a reference to the command interface that's used to populate its properties.

DataBeanManager: The client class that executes the appropriate Data Bean Command implementation for a data bean.

Figure 3 illustrates the collaboration between participants in the Data Bean Command pattern.

CONSEQUENCES: The consequences of using the Data Bean Command pattern are as follows:

a. The Data Bean Command interface and implementation are first-class objects that can be manipulated and extended to change existing data fetching, processing, and population logic to provide new logic.

b. Data fetching, processing, and population logic can be changed without affecting the data objects.

c. Data objects must be aware of the (Data Bean Command) interface name whose implementation will be used to populate it. This interface name can be set explicitly by the client at runtime.

d. Each store type can have different fetching, processing, and population logic.

IMPLEMENTATION:

a. To create a new Data Bean Command in WebSphere Commerce you must create a new Java interface that extends DataBeanCommand and provides an implementation class for this extended Java interface. The implementation class must extend the default Data Bean Command implementation, DataBeanCommandImpl class.

b. The data population logic must be in the performExecute method. The caller (DataBeanManager) is programmed to the Data Bean Command interface. So all the input properties must be defined in the data bean or passed using setRequestProperties.

c. Always instantiate a Data Bean Command using the CommandFactory and call the execute method to execute the business process. Never call the implementation class's performExecute method directly.

d. A data bean instance must be explicitly passed into the Data Bean Command implementation class.

e. As a best practice, always use the populateDataBeanCommand method to populate the input data from the data bean to the Data Bean Command properties.

f. New population logic and extensions to existing population can be performed either by creating a new implementation class for the existing Data Bean Command interface or extending the existing Data Bean Command implementation class. When you make extensions, make sure you call the parent class's performExecute method.

g. You can modify the population logic by setting a different interface name in the data bean with setCommandInterfaceName.

EXAMPLE:

```
public interface SearchCmd extends
  DataBeanCommand{
public static final Java.lang.String default
  CommandClassName = “com.ibm.commerce.search.
  commands.SearchCmdImpl”;
public static final Java.lang.String NAME =
  “.ibm.commerce.search.commands.SearchCmd”;
}
public class SearchCmdImpl extends
  DataBeanCommandImpl
  implements SearchCmd {
public void populateDataBeanCommand(InputDataBean
  aSearchDataBean) {
//get relevant input properties from data bean
  and set to
//command properties.
}
public void performExecute() throws ECException {
  //do your population logic
}
}
public class SearchDataBean implements
CommandDataBean{
  private String commandInterfaceName=null;
public String getCommandInterfaceName(){
  if (commandInterfaceName != null){
              return commandInterfaceName;
        } else{
return SearchCmd.Name;
}
  }
}
```

RELATED PATTERNS:

A *Factory* (CommandFactory) pattern is used by DataBeanManager to instantiate an appropriate implementation of the Data Bean Command.

This pattern must be used to populate data beans that are created using the *Command Data Bean* pattern.

Smart Data Bean

INTENT: Encapsulate an entity's data attributes thereby providing a simple representation of WebSphere Commerce entities for display purposes, and efficiently transfer remote fine-grained entity data as a coarse-grained data object in one network transfer.

MOTIVATION: You may want to display dynamic data such as product information in your commerce site. This information can be stored in your database and represented as an entity bean on the server. While working on the client side, it would be easy to work with a simple representation of these entities. Furthermore, we don't want to go to the server every time we need a product attribute. Instead, we'd like to get a simple coarse-grained product information object on the client side and get the attributes when we need them.

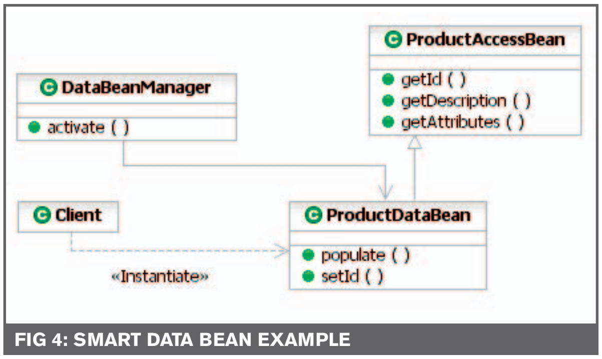


FIG 4: SMART DATA BEAN EXAMPLE

We can achieve this by encapsulating the entity's data in a simple Java bean. In our product information display example, we can create a simple Java bean ProductDataBean that encapsulates the product information by sub-classing the ProductAccessBean. The client (JSP page) sets the unique identifier of the product that it needs to display in the data bean and activates the data fetch using DataBeanManager. When activated, the ProductDataBean backed up by the ProductAccessBean refreshes itself with the current data. The client will then have a coarse-grained ProductDataBean object that can be used to display the product information.

APPLICABILITY: Use this pattern when you want to encapsulate all of an entity bean's attributes and transfer the data in one network transfer.

STRUCTURE:

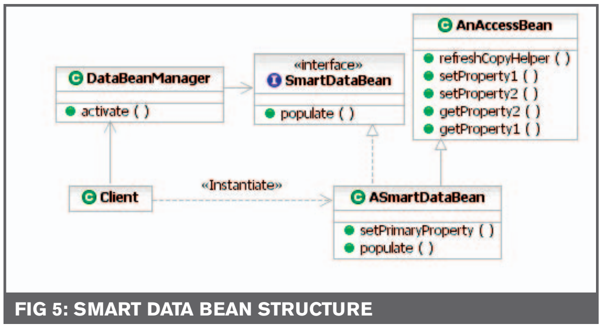


FIG 5: SMART DATA BEAN STRUCTURE

PARTICIPANTS:

SmartDataBean: A Java interface that declares a method to trigger data population.

ASmarDataBean: A Java bean class that is a sub-class of a corresponding entity's access bean and implements the populate method declared in SmartDataBean (to populate the bean with recent entity data). This class contains data needed by the display client.

AnAccessBean: This is an access bean that corresponds to an entity bean. The refreshCopyHelper method is used by its sub-class ASmartDataBean to refresh the access bean with recent data.

DataBeanManager: The manager class that triggers the data bean population.

Client: The client that requires the entity's attributes. The client creates the Smart Data Bean and sets the primary key of the business entity whose attributes it requires.

Figure 6 illustrates the collaboration between participants in the Smart Data Bean pattern.

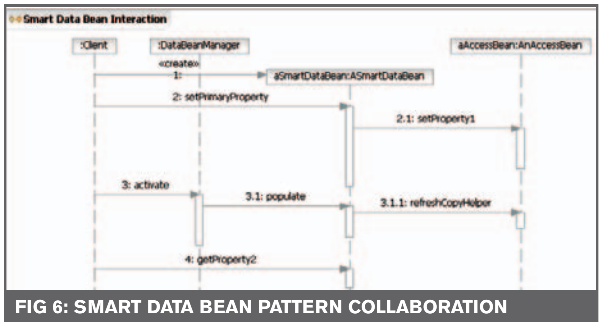


FIG 6: SMART DATA BEAN PATTERN COLLABORATION

CONSEQUENCES: The consequences of using the Smart Data Bean pattern are as follows:

a. The Smart Data Bean pattern provides a simplified view of an entity to the client.

b. Network traffic is reduced by transferring a coarse-grained object that represents the entire entity.

c. If the entity has many attributes, this can result in a large data object being transferred across the network.

d. When a new property is added to the entity, the Smart Data Bean picks it up automatically. But the access bean of that entity must be regenerated.

e. Entities can have related entities. If these related entities are fetched when the main entity is retrieved, the resulting Smart Data Bean object will become bulky. Instead, try to use a lazy fetch strategy to fetch related entities.

IMPLEMENTATION:

a. To create a new Smart Data Bean in WebSphere Commerce, you have to create a new Java class that implements a SmartDataBean interface and extends the access bean of the entity for which the data bean is being created.

b. The data bean is created by the client (for example, a JSP). The client must also set the primary key of the entity for which the attributes are to be retrieved before activating the data bean manager.

c. Always call the refreshCopyHelper of the access bean in the populate method to refresh the access bean with recent data.

d. When an entity contains related entities, try to use a lazy fetch strategy to fetch these related entities. The related entities must have their Smart Data Beans. Make a call to the related entity's Smart Data Bean only when the client needs the properties from the related entity (rather than making a call when the populate method is called by the client).

EXAMPLE:

```
public class ProductDataBean extends
ProductAccessBean implements SmartDataBean{
public void setProductID(String astrProductId) {
  super. setInitKey_productID(astrProductId)
}
public void populate() throws Java.lang.Exception
{
super.refreshCopyHelper();
}
public Long getCalculatedPrice () {
//Use lazy fetch mechanism
//Create PriceDataBean and retrieve the price for
  this product
PriceDataBean ibnPrice = new PriceDataBean(...)
return ibnPrice.getPrice();
}
}
```

RELATED PATTERNS: None

Command Data Bean

INTENT: Create a dynamic data object for display whose property value can vary with business needs and the WebSphere Commerce store type.

MOTIVATION: You may want to display dynamic data in your commerce site that may not be readily available in your database and may require some complex algorithm. The algorithm used to process this data can change with business needs and must not be exposed to the client, whose primary responsibility is to display the data.

Product comparison displays are a common feature in commerce sites. They can involve complex processing logic and we would not want to have this logic in our data object. We can solve this problem by separating the processing logic from the data object by creating a new object that will process and populate the data object data when the processing is done. We can make use of the Data Bean Command pattern to implement the processing logic to vary the processing logic according to business needs.

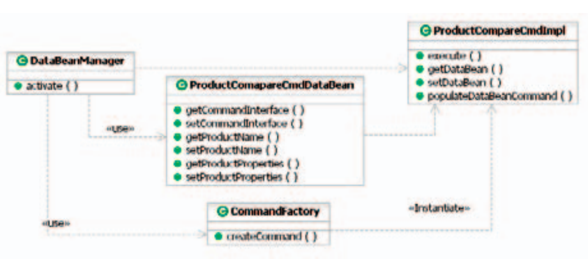


FIG 7: COMMAND DATA BEAN PATTERN EXAMPLE

In our product comparison example, the processing logic will be in ProductCompareCmdImpl, which follows the Data Bean Command pattern. The display client will instantiate the Java bean ProductCompareCm dDataBean and make use of DataBeanManager (which in turn will use CommandFactory) to instantiate an appropriate processing logic object (which in our product comparison example is ProductCompareCmdImpl). The DataBeanManager will also set the data bean object in ProductCompareCmdImpl so that when processing is over, the command object can set the data in the data bean object. When business needs change, the client can modify the processing logic by using the setCommandInterface method in the data bean or the command registry, which will dynamically modify the implementation of the Data Bean Command.

APPLICABILITY: Use this pattern when you want to have a Java bean data object for display, when the data in the data bean is a result of a complex business process, and when this process can change with business needs.

STRUCTURE:

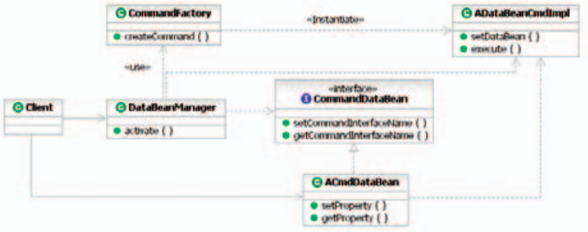


FIG 8: COMMAND DATA BEAN PATTERN STRUCTURE

PARTICIPANTS:

CommandDataBean: A Java interface that declares methods to set and get the Data Bean Command name that populates the data bean.

ACmdDataBean: A Java bean that contains data needed by the display client and implements the methods declared in CommandDataBean.

DataBeanManager: The manager class that makes use of the CommandFactory to create an appropriate Data Bean Command implementation and execute the command.

CommandFactory: Instantiates an appropriate Data Bean Command.

ADataBeanCmdImpl: A Data Bean Command implementation class that does the processing and populates the data into ACmdDataBean.

Client: Instantiates ACmdDataBean, activates DataBeanManager and retrieves data from ACmdDataBean to display.

Figure 9 illustrates the collaboration between participants in Command Data Bean pattern.

CONSEQUENCES:

The consequences of using the Command Data Bean pattern are as follows:

a. The Command Data Bean pattern relies on the Data Bean Command pattern to retrieve its data.
b. The data bean contains all attributes populated by the command whether the display client requires them or not. If the display client doesn't need all the attributes, this data bean can become costly in terms of performance.

IMPLEMENTATION:

a. To create a new Command Data Bean in WebSphere Commerce you have to create a new Java class that implements the Command Data Bean interface.
b. The data bean is created by the client (for example, a JSP). The client can set input parameters that will be used by the command implementation through the data bean.
c. By default, the DataBeanManager will execute the command that's returned by the getCommandInterfaceName method in the data bean. If the client needs DataBeanManager to execute a different implementation of the command then the client must set the interface of the command in the data bean class using setCommandInterfaceName method. An alternative implementation can also be provided by changing the command registry.

EXAMPLE:

```
public class ProductCompareDataBean implements
  CommandDataBean{
  private String commandInterfaceName=null;
public String getCommandInterfaceName(){
  if (commandInterfaceName != null){
                return commandInterfaceName;
        } else{
```

```
return ProductCompareCmd.Name;
}
  }
public void setCommandInterfaceName(String ifname)
{
  commandInterfaceName = ifname;
}
//declare properties, define its setters and
  getters
}
```

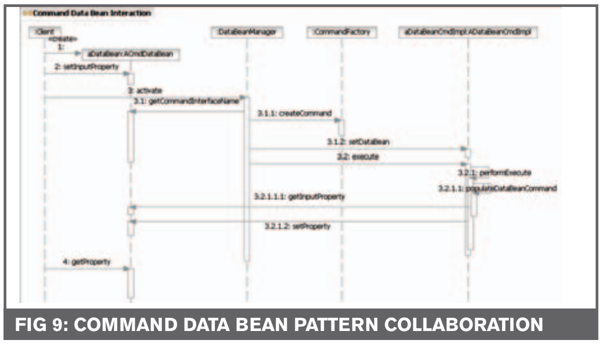


FIG 9: COMMAND DATA BEAN PATTERN COLLABORATION

RELATED PATTERN:

This pattern shown in Figure 9 uses the *Data Bean Command* pattern for populating data into a data bean.

The Big Picture

Figure 10 illustrates the relationship between the patterns described in this article. WebSphere Commerce follows a Model-View-Controller architecture. The figure illustrates where these patterns fit into this architecture. Refer to the WebSphere Commerce Developers' Guide for more details about the runtime architecture.

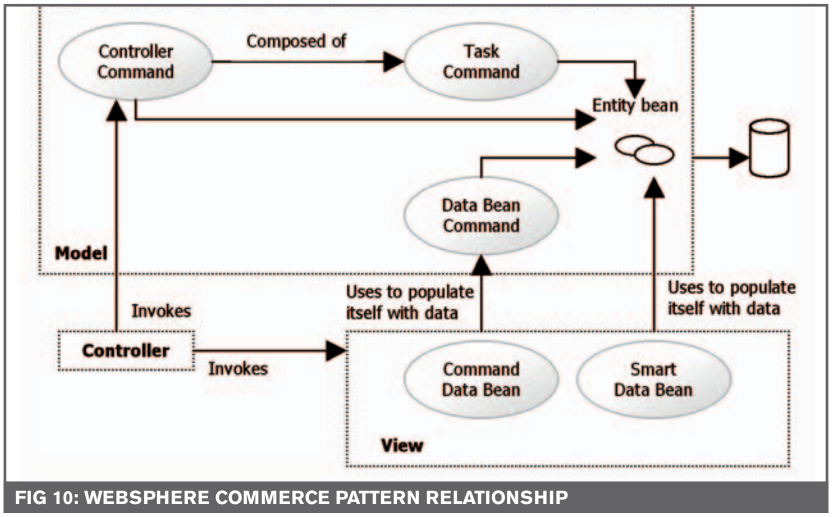


FIG 10: WEBSPHERE COMMERCE PATTERN RELATIONSHIP

And how it's all tied up with Web Services

SOA and Its Impact on EAI and On-Demand

BY: PRAVEEN K. CHHANGANI

As a certified IBM WebSphere MQ Workflow Specialist, Praveen K. Chhangani is part of Prolifics? specialized team of WebSphere consultants whom IBM calls upon to service its most challenging customer requirements by providing training, customization, administration and configuring, architecture design, development, and deployment of distributed architectures.

cpraveenchhangani@yahoo.com

Architectures aren't what they used to be, and our business models and drivers aren't either. As our business processes and drivers grow more and more complex so does the need to improve the underlying architectures that support the models and drivers. In fact, it's important to realize that well-built, robust, and scale-able architectures are no longer a luxury, but a requirement just like the building blocks of a new house.

Advances in technology and the ubiquity of the Internet have realigned the way in which business functions. Just look at the host of e-business avenues such as Internet sales portals and information portals that have been created.

This is good news for consumers because it means more competition and more competition means more competitive products at more competitive prices.

Consumer Expectations

Consumers want to get the most for their money and their expectations usually demand quick turnaround time and quality service. Their expectations constantly challenge product makers to build more and better products that sell for cheaper prices using more efficient and effective means of production.

What Can Be Done?

To succeed in such an environment of expectation and competition, organizations need a powerful IT setup with a robust infrastructure such as an on-demand operating environment that can recognize, respond to, and adapt to day-to-day fluctuations in volume demands from customers or retailers and security threats and risks more quickly and effectively.

Such an environment lets businesses improve their operations by giving them access to the latest technology advances that ultimately serve the customer. And it promotes a faster return on investment (ROI) by decreasing IT maintenance, and the time and cost that goes with it, and improving employer and employee productivity by bringing down the cost of business and the total cost of ownership (TCO).

Common Practices

Of course every business organization is different, but over the past couple of decades most have created applications and architectures to solve their business problems.

However, that means that we now have to deal with an increasing number of point-to-point applications, each with its own complex architecture, as well as a rising number of maintenance tasks and the need to update existing – but hard-to-reuse – applications with newer business models, all of which creates more redundancy in our infrastructure. In other words, we are solving business problems in a way that involves more time, more money, and more resources and detracts us from the real business model – and how it can be improved so the business is more competitive.

A Service Oriented Architecture will help us with these challenges and give us a base on which to establish a solid on-demand operating environment that lets IT devise a strategy for building newer, more efficient applications while reusing existing assets.

"An on-demand operating environment can be defined as a set of integration and infrastructure management capabilities you can use to become an on-demand business," according to Fredrik Carlegren and Tim McCrimmon.

SOAs: In Demand or On-Demand?

A Service Oriented Architecture is the basis of an on-demand operating environment. It has a tremendously powerful impact on the deliverables of the on-demand operating environment and provides for more effective Enterprise Application Integration (EAI).

SOAs are the latest buzzword. Everyone is talking about them, and while the concept isn't exactly new, it's finally getting to be better known and accepted.

The question is why now?

If the first Service Oriented Architecture used Object Request Brokers (ORBs) or DCOM – which have been around for ages – why are SOAs

NEW!

CodePro AnalytiX™

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

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

Sound like a fantasy?

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

Your money back.

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

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

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

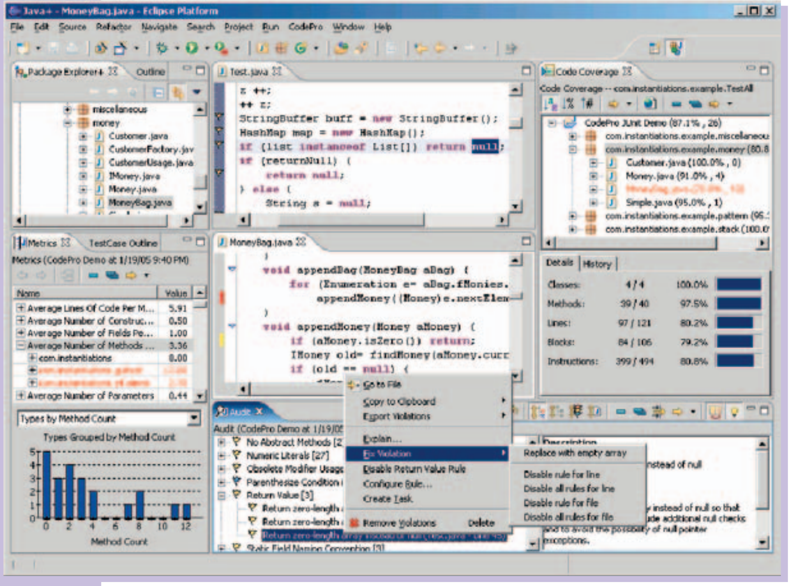
AnalytiX also allows you to define, distribute, and enforce corporate coding standards and quality measures across your development teams...no matter where they're located.

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

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

Put our experience to work for your team.

www.instantiations.com
1-800-808-3737



```
z ++;
++ z;
StringBuffer buff = new StringBuffer();
HashMap map = new HashMap();
if (list instanceof List[]) return null;
if (returnNull) {
    return null;
} else {
    String s = null;
```

```
    }
    void appendBag(MoneyBag aBag) {
        for (Enumeration e= aBag.fMonies.
            appendMoney((Money)e.nextElem
    }
    void appendMoney(Money aMoney) {
        if (aMoney.isZero()) return;
        IMoney old= findMoney(aMoney.curr
        if (old == null) {
```

Key Product Capabilities

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

Risk Free Trial Copy

www.instantiations.com/analytix

eclipse READY IBM Business Partner eclipse FOUNDATION MEMBER

instantiations

only gaining wide acceptance now as the next level in software design, development, and management?

One might think it has something to do with the maturity of the technology, but it's really the emergence of technologies such as Web Services and what SOAs can do for them and vice versa.

See, a Service Oriented Architecture is based on intelligent services that are well defined, loosely coupled, and self-contained. Web Services fit that description quite well. SOAs based on Web Services can dramatically speed up the application development process. SOAs are not just in demand, but truly on-demand.

Making the Decision – Are SOAs for Me?

Before deciding to use an SOA, it's critical to invest the time needed to evaluate all the relevant factors and educate yourself as an organization. This can be done by attending seminars, listening to webcasts, and bringing in specialized consultants to educate your staff and personnel as well as to help you turn on the "SOA switch."

EAI entails a lot of different varieties and flavors of working models, applications, and prototypes. So it's important to thoroughly understand what works best for your business before making a decision.

Businesses need to be put in the picture. I was once invited to an architecture meeting where, when I mentioned SOA, a top-level executive and his business architect thought I was referring to some fancy form of ketchup! No kidding.

Some organizations aren't interested in an on-demand operating environment and a service oriented architecture, others have the desire but not the background or commitment. This can lead to jumpstarting an SOA project too early and then aborting it due to unforeseen factors, or getting discouraged early on because of the investment cost, the time spent on training, or resource availability.

So I re-emphasize my point about doing your homework first since it really means gaining a whole new perspective on IT infrastructure and application management.

If you're looking to create a truly on-demand IT infrastructure, an SOA is the first step and the base foundation of an on-demand business application. According to Kishore Channabasavaiah and Kerrie Holley, "SOA can be both an architecture and a programming model, a way of thinking about building software."

It is equally certain that, as Praveen Chhangani puts it, "With the power of a streamlined environment, you can align and integrate your organization's resources and capabilities with your business strategies, accelerating the traditionally long-running process flows – thereby cutting overall costs and eliminating errors."

IBM Offerings: SOA and Web Services Development

BEA and Microsoft provide SOA-driven development models but for purposes of this article, I only intend to touch on IBM's offerings.

IBM understands that companies across the board are looking for ways to respond more quickly and effectively to changing market conditions and IBM promotes SOAs as a path to that flexibility. IBM categorizes its offerings by role and function. The following is a list of its key products for SOA and Web Services development.

- ***Business Analysts:*** (WebSphere Business Integration Modeler) – Software tools for modeling, simulating, and analyzing complex business processes quickly and effectively.
- ***Software Architects & Model-Driven Developers:*** (IBM Rational Software Architect) – An integrated design and construction tool for creating service-oriented applications that leverages model-driven development with the UML and unifies all aspects of software application architecture.
- ***Java and Web Developers:*** (IBM Rational Application Developer) – A comprehensive IDE for quickly designing, constructing, analyzing, testing, profiling, and deploying service-oriented applications.
- ***Integration Specialists:*** (WebSphere Studio Application Developer IE) – An Eclipse-based IDE for integration specialists to create composite applications that deploy to WebSphere Integration Server Foundation.

> **"SOA is the basis of an on-demand operating environment and provides for more effective Enterprise Application Integration"**

Essential Attributes of Well-Engineered Software

Whether your business organization is in the process of migrating to an SOA or plans to incorporate an SOA to achieve on-demand operating environment deliverables, the need for well-engineered software and design is crucial. All well-engineered software and related components have certain characteristics. According to Ian Sommerville, well-engineered software has four essential attributes as shown in Table 1.

PRODUCT CHARACTERISTIC	DESCRIPTION
Maintainability	It should be possible to evolve software to meet changing customer needs.
Dependability	Software dependability includes a range of characteristics including reliability, security, and safety. Dependable software won't cause physical or economic damage if a system fails.
Efficiency	Software shouldn't waste system resources such as memory and processor cycles.
Usability	Software should have an appropriate user interface and adequate documentation.

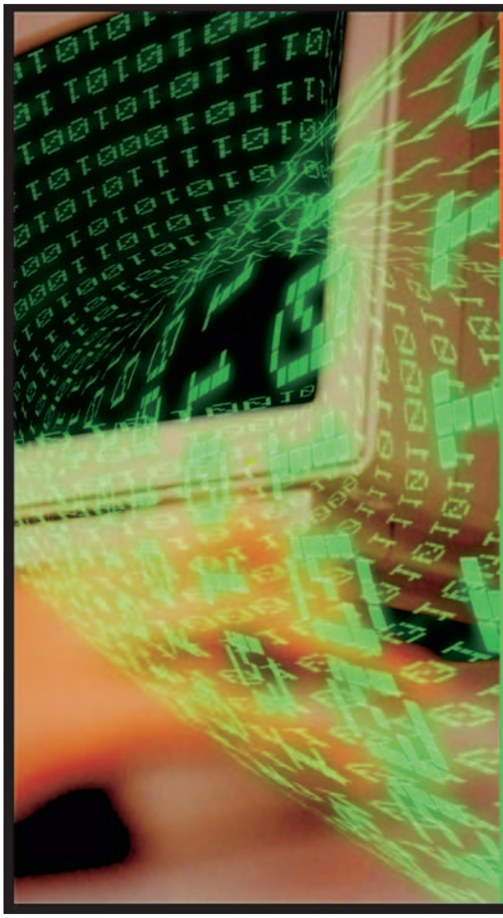
TABLE 1

What Now?

Just as every end is a new beginning, every beginning has an end. In other words, there will always be numerous avenues made available to us as IT develops. Constantly improving on existing IT assets and building newer and more effective applications will play an important role in serving the focal points of our businesses organizations, and ultimately create more structure, stability, and solidarity. My advice is to take your time, do your homework, understand the value, and then make it real and applicable to your business.

References

- Praveen Chhangani. "Creating a Healthy, Optimized Workflow Environment." WebSphere Developers Journal, Volume 3, Issue 10.
- Fredrik Carlegren, Angel Luis Diaz, and Tim McCrimmon. "Operating Essentials for an On-Demand Breakthrough." IBM DeveloperWorks, 2004.
- Kishore Channabasavaiah and Kerrie Holley. "Migrating to a Service Oriented Architecture." IBM Global Services, and Edward M.Tuggle, Jr., IBM Software Group White Paper, 2004.
- Ian Sommerville. "Software Engineering." 5th Edition, 1998.



WebSphere Business Integration Modeler, Rational Software Architect, and RequisitePro

IBM Software Integration

There is often a gap between what the business needs are and what IT delivers. Modeling the business, and the directly related IT systems to it, helps define the most effective opportunities for automation. Business modeling also helps the development team get a head start by helping to identify system use cases and finding requirements.

BY: REID GERSON

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

Business modeling can provide significant value for software development and helps stakeholders have a permanent understanding of how the business works that doesn't disappear when staff leaves. You can understand current problems in the business and identify improvement potentials, derive the business requirements needed to justify improvements, and ensure software teams pay more than superficial attention to business needs.

Business modeling can be worth the effort for software development because it helps find requirements. In the "Exploring Process Automation" workflow detail one activity is "Define Automation Requirements." It results in the beginnings of the system use case model, the supplementary specification, and the analysis model. This activity results in use cases, nonfunctional requirements, and even a start at the analysis model.

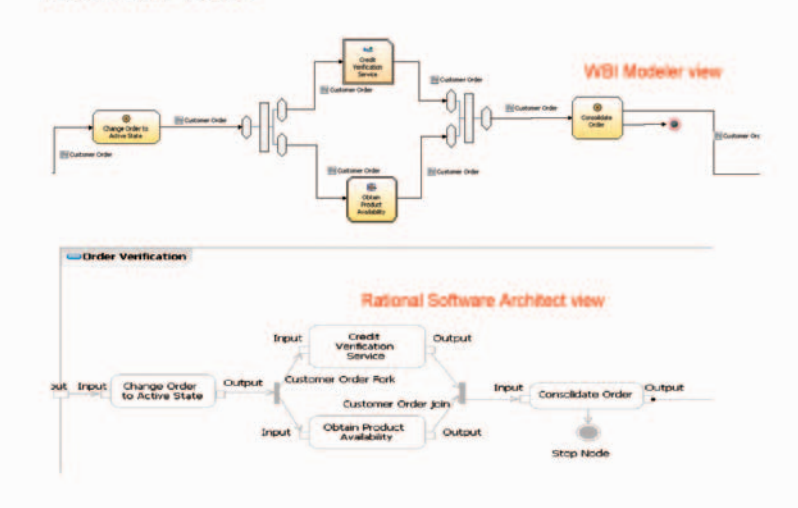


FIG 1: PROCESSES VIEWS

Requirements are central to any mature software development process. Business modeling helps ensure that software requirements and use cases reflect business needs. Well-written use cases provide the foundation for Architecture, Analysis, UI design, and Testing. The result is a system that meets the requirements and has business value.

In February 2005, IBM announced that the integration between IBM WebSphere Business Integration Modeler and IBM Rational Software Architect provides a way to migrate business models to software models that will automate business activities. The integration automatically maps business notation (BPEL) to software notation (UML), thereby relieving designers from that conversion burden. In addition, there is perspective in Rational Software Architect for IBM Rational RequisitePro, an easy-to-use requirements management tool.

The WebSphere Business Integration Modeler, Rational Software Architect, and Rational RequisitePro Integration:

- Opens WBI models in Rational Software Architect to view as UML
- Uses the WBI model as specification
- Creates a conforming Rational Software Architect model

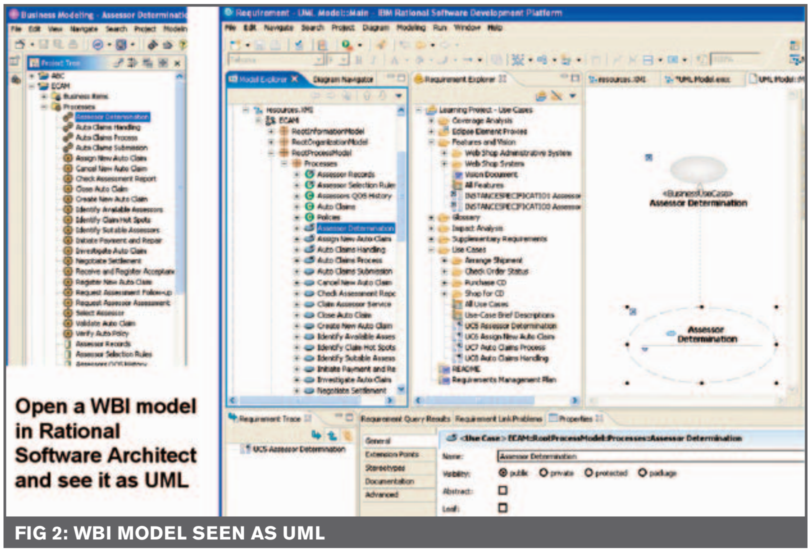

Open a WBI model in Rational Software Architect and see it as UML
Assessor Determination

FIG 2: WBI MODEL SEEN AS UML

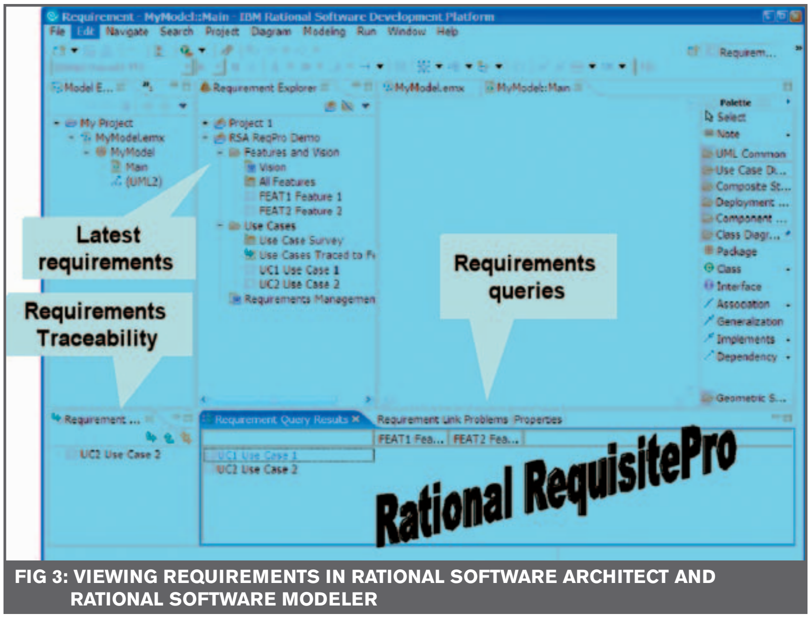


FIG 3: VIEWING REQUIREMENTS IN RATIONAL SOFTWARE ARCHITECT AND RATIONAL SOFTWARE MODELER

- BPEL from WBI can then call the Rational Software Architect implementation
- Cross-model references to the UML view of WBI model
- Requirements associations to the UML view of WBI models
- Allows you to author and share requirements into RequisitePro from Rational Software Architect and leverage capabilities such as requirements traceability and impact analysis

Business processes are represented in WBI Modeler using the business-preferred notation, and in Rational Software Architect with the standard UML notation used by software designers.

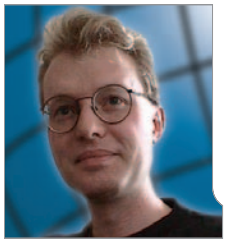
The integrity of the business model is maintained and the integration allows software teams to jumpstart their work rather than having to re-create their own version of the business model.

Using Eclipse-based Rational Software Architect or Rational Software Modeler (no direct integration in Eclipse), developers can have up-to-date access to the latest state of requirements stored in RequisitePro.

Part three of a three-part article

Putting WAS on Unix

BY ADE RIXON



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

In the last part of this three part series, we're going to look at some advanced topics and further considerations for running WebSphere effectively under UNIX, including monitoring, security and resilience.

Monitoring WAS

The immediate status of all WAS servers can be obtained using the command:

```
#serverStatus.sh —all
```

WAS processes can be listed using the command (Solaris):

```
$pgrep -f -l 'WebSphere.*java'
```

Current WebSphere process IDs are also written to files in the application server log directories (e.g., MyAppServer.pid).

The most effective way to monitor WebSphere and its applications is to request one or more known stable URLs directly from the application server. For example, either curl or the check_http plugin in the Nagios monitoring package can do this. You can optionally verify the contents of the response to be sure that the application is working correctly.

It's also useful to monitor the load (uptime) and memory usage on each system, since faulty application server processes often consume all the CPU time and/or virtual memory.

General Tips for Running WAS

- Run backupConfig.sh at regular intervals via cron on each installed WAS host. Ideally, copy each backup archive file to a remote system. Note that these files can be quite large, since they also contain a copy of every installed application.
- Review the section on WAS Security below and check the issues raised against your local security policy.
- Install the Tivoli Performance Viewer and Log Analyzer components from WebSphere on a desktop system, such as your PC (you may need the Windows version of the software for this).
- If you're developing shell scripts to manage WebSphere, it's worth sourcing the WAS setup script at the start since it provides a number of useful environment settings. For example, deployed application files can be found under ${WAS_HOME}/installedApps/${WAS_CELL}/ .
- WebSphere should be properly tuned for good performance under intensive loads. See the bibliography for suitable references.

Advanced Topics

This section contains a brief overview of other WebSphere features and issues, particularly those that are relevant to enterprise use of WAS.

WAS Scripting

Besides the administration console, WAS operations can be scripted using Java implementations of TCL and Python called JACL and Jython. These scripts are run using the *wsadmin* utility (wsadmin.sh). Scripts configure and control WAS via various method calls on different Java Management eXtension (JMX) objects known as *Managed Beans* (MBeans).

Unfortunately, there's no comprehensive documentation available on all the possible MBeans (their number and types vary depending on the WAS topology and installation). The syntax and use of MBeans is rather verbose and unfriendly. The best – in fact, the only – way to learn how to perform tasks via scripting is by example. (While wsadmin has a help facility, its contents are rather terse.) There are a number of sample scripts in the Infocenter, the Redbooks, and other online articles from IBM.

It's also difficult to debug faulty scripts except by running wsadmin interactively and entering the individual commands, since script error messages don't include line numbers. Despite these difficulties, scripting is very useful if you wish to configure and control a number of separate, identical WAS cells, or do the same operations repeatedly on a consistent basis. Scripting also offers a path towards automated WebSphere deployments and recovery procedures.

It may be easier to tackle the tasks that are simplest to script, better documented, and more frequently done first (for example, installing EAR files), and gradually develop more complex scripts as you gain experience.

Working with the wsadmin interface is easier if you have command line editing and history facilities. Although the program doesn't support this itself, the Open Source rlwrap utility can be used to add these features.

WebSphere Application Server Network Deployment

The base edition of WebSphere only supports a single node in a cell. WebSphere Application Server Network Deployment (WAS ND) is used to support and manage several nodes in a distributed cell. WAS ND is a separate product that can be installed either alongside the base edition or on a dedicated system. It provides a component called the *WebSphere Deployment Manager* (DMgr), which is a specialised application server process. The DMgr has a Web-based administration console very similar to the base edition, but with extra features for managing a cell containing more than one node. Distributed cells provide workload balancing, scalability, and resilience improvements via the use of *clusters* (groups of application servers running the same application). For example, user sessions can be migrated from a failed application server to another one in the cluster using the *session persistence* feature.

Existing node base installations are *federated* or added into a DMgr cell. During this process, their own adminconsole application is removed and they are henceforth configured from the DMgr. To accomplish this, a separate, dedicated server called the *Node Agent* is created on the remote node. New application servers can be created or cloned from existing ones on any node in the cell. However, there's no facility for making identical changes to several application servers simultaneously.

When configuring WebSphere variables or JDBC providers, it's best to make these changes in the cell scope so that they're identical on every node and only configured in one place. Be sure to delete any duplicate entries at lower levels, such as the node scope.

If you're running critical applications in production environments, you should use at least two servers for resilience and performance reasons. In this case, you'll need to identify a server suitable for the Deployment Manager. This can either be one of your existing WebSphere servers, a lightly used backend system (in a dual role), or even a dedicated system.

For maximum availability, the use of multiple identical cells is recommended. Each cell requires its own deployment manager and group of nodes, and must be managed separately. This allows individual cells to be taken down for maintenance or the staged implementation of fixes and application updates. See *WebSphere Deployment and Advanced Configuration* for details.

> **"Remember that WebSphere, perhaps more than other applications, requires a close partnership between developers and system administrators"**

Key Differences in WAS ND

- EAR deployment is performed on the DMgr, which takes care of copying the files to each node.
- The console also contains configuration dialogues for the **dmgr** server and **node agents**.
- Network communication between the node agents and the DMgr must be enabled.
- The JDBC **Test Connection** option in the administration console only runs on the DMgr. This may or may not work, but it won't reliably indicate whether the data source works on the application nodes. (It's possible to work around this by running the connection test on the nodes via wsadmin.)

Product Fixes

IBM regularly releases updates and fixes for its products, including WebSphere. These can be downloaded from IBM's Web site for WebSphere support; a support agreement or contract number is not required.

WebSphere fixes come in three forms:

- ***Interim fixes:***These are fixes for individual problems or issues (referred to by IBM Support as *Authorised Program Analysis Reports* or APARs).
- ***Cumulative fixes:***These are bundled collections of fixes designed to be installed on a base release or fix pack. Each cumulative fix (CF) supersedes previous cumulative fix releases. It can be installed on any previous version of the related base and fix pack. Installing a cumulative fix updates the fourth number in the WAS version identifier (e.g., CF3 for Fix Pack 1 updates any V5.1.1.x release (where x < 3) to V5.1.1.3). Cumulative fixes for individual components, such as the Java Software Development Kit (SDK) and HTTP Plugin, are made available separately.
- ***Fix pack:***A fix pack is a bundled collection of fixes and minor new features. It can include updates to the Java SDK used by the product and the bundled IHS release.

Installing a fix pack overrides any previous fix pack or cumulative fixes. It can be installed on any previous version of the major product release. A fix pack updates the third number in the version identifier (e.g., FP1 updates V5.1.0.x to V5.1.1).

Each type of fix is released in several versions corresponding to the various supported WebSphere platforms, products, and architectures. Make sure you download the correct file for your hardware, operating system, and product name. (Note that WAS ND fixes are different from base edition fixes.)

It's recommended that you install only the most recent fix pack for WebSphere V5.1. If you encounter problems that appear to be related to the product, you may want to install the latest cumulative fix relevant to the installed fix pack. In rare cases, you may require an interim fix to address a specific issue that's not covered by any of the bundled fixes.

At time of this writing, you would download Fix Pack 1 (FP1) for WebSphere Base. If you're using WAS ND, you also have to download FP1 for ND.

WebSphere Security

By default, WebSphere is installed with its built-in security features disabled. You can configure and control WAS without authentication – and so can anyone else with access to the administration console or host system. WebSphere Security is a complex topic; you must read the Infocenter, the Security Redbook, and all related documents carefully before enabling it. Here is an overview of the subject.

Once enabled, WAS Security encrypts some connections in cells using SSL and limits administrative access to defined username/password combinations authenticated against a defined registry. It can also be leveraged by applications for user access control and authentication, and internal security.

- WebSphere is installed with a default ("dummy") set of SSL certificates and keys. Ideally, these should be replaced with a new set generated with IBM's ikeyman utility before enabling Security. The certificate sets have to be installed on each node in a cell and any Web servers that use the HTTP Plugin to communicate with the cell.
- WebSphere can use either the local OS password file, an LDAP server, or a custom registry to retrieve authentication details. Sample code is provided for a file-based custom registry, but IBM doesn't recommend using it in a production deployment.
- WAS ND can only use LDAP or a custom registry for authentication. All nodes in the cell have to be able to access the registry.
- WebSphere doesn't support replicated LDAP servers (you can only specify a single LDAP server address). Any LDAP server you use has to have a transparent high-availability mechanism.
- Once Security is enabled, a user name and password must also be given as arguments when using wsadmin or the command scripts. If you write shell scripts around them to run unattended, you'll have to embed the user name and password details in them. Obviously, this makes such information less secure (although user actions can be restricted to one of four defined *roles*).
- WebSphere products install with global file read permissions. However, many of the files within them should be protected from general user access, including the configuration repository (config/), the SSL key files (etc/), and possibly the logs (logs/). This means changing the default file permissions manually.
- If the HTTP Plugin configuration file is automatically generated, it will include the URI for the administration console. You do *not* want this to be accessible on a production Web server. Copy and edit the file to remove it, and include an ACL rule on your Web servers blocking all external access to /admin/*.

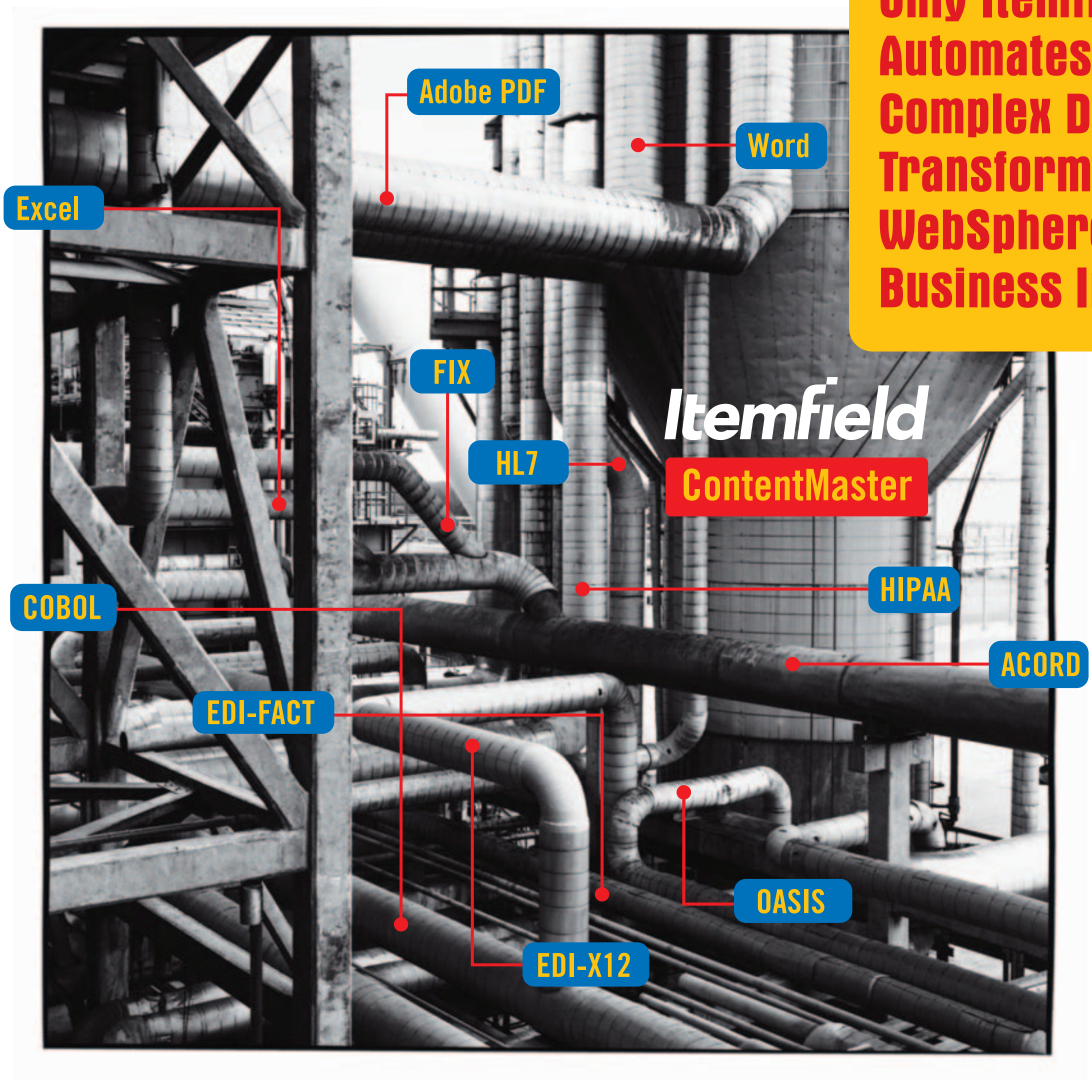
It's worth considering whether careful firewalling and IP access controls that limit connections to a small defined set of addresses can effectively control administrative access without having to enable WAS Security with its attendant overheads and complexities. This is particularly true if your application uses its own authentication system rather than relying on WebSphere to provide security.

Minimum Security Recommendation

The following configuration enables the minimum amount of WAS Security, mainly to protect the administration functions, without significant overhead. It may not be sufficient for production environments.

- Using the file-based custom registry example, create a flat file containing defined users for the Administrator and Operator roles. Use the former for the Administration Console and the latter for scripts that control WAS. Copy the file to all nodes in the cell and make sure that only the user ID under which WAS runs has read access. If you have an existing resilient LDAP infrastructure, you may want to use that instead.
- Enable WAS Global Security and configure the registry details for the chosen method.
- Run WAS processes under a non-root account with limited access rights and privileges. (See the Infocenter topic "Running the application server with a non-root user ID." This can also be done for the WAS ND Deployment Manager.)
- Limit access to the WAS configuration, application, and log files using normal Unix file permissions and ownerships.

Only Itemfield Automates Complex Data Transformation For WebSphere® Business Integration



Extend The Power And Reach of WebSphere Business Integration

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

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

From Itemfield

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

From IBM

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

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

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

WebSphere and Firewalls

WebSphere Application Server on a single node is reasonably simple to firewall; just make sure that the Web server can reach the application server HTTP and HTTPS ports, and the application server can reach any backend services such as databases. Ideally production application servers and Web servers should be on separate, dedicated network segments in demilitarised zones (DMZs).

In a distributed cell with the deployment manager firewalled from the application nodes (which is recommended), a number of ports must be opened up between the two. Regrettably, this includes the default administration console port (9090), since it's shared with the filetransfer application that the application servers use to pull data. Hence you may still need WebSphere Security to protect the console (e.g., in the event of the application servers being compromised).

FROM	TO	PORTS
DMgr	Nodes	BOOTSTRAP (2809)
		DISCOVERY (7272)
		ORB (9900)
		SOAP (8878)
		SOAP (8880) for server1
Nodes	DMgr	BOOTSTRAP (9809)
		DISCOVERY (7277)
		ORB (9100)
		SOAP (8879)
		FILETRANSFER (9090)

TABLE 1: DEFAULT WAS TCP PORT CONNECTIONS

Table 1 shows the TCP ports that must be opened between node agents and the DMgr for management and initial node federation, together with the default port numbers following a standard installation. You should confirm the port assignments by examining the serverindex.xml files in the WAS configuration repository, or the details for each server in the administration console. Additional ports may be required when WAS Security is enabled.

Conclusion

You should now have a working single application server and some knowledge of more advanced configurations. To go further, you must rely on the documentation made available by IBM, which is hopefully now more approachable.

Remember that WebSphere, perhaps more than other applications, requires a close partnership between developers and system administrators. If you encounter problems, it's often productive to tackle them together, since developers have a good understanding of Java and the application code while administrators possess an appreciation of the bigger picture and the interdependencies with external services.

Online Resources

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

References

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

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

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

To get your free copy of
the informative new article

"3 Steps to Uncomplicating Business Process Management"

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

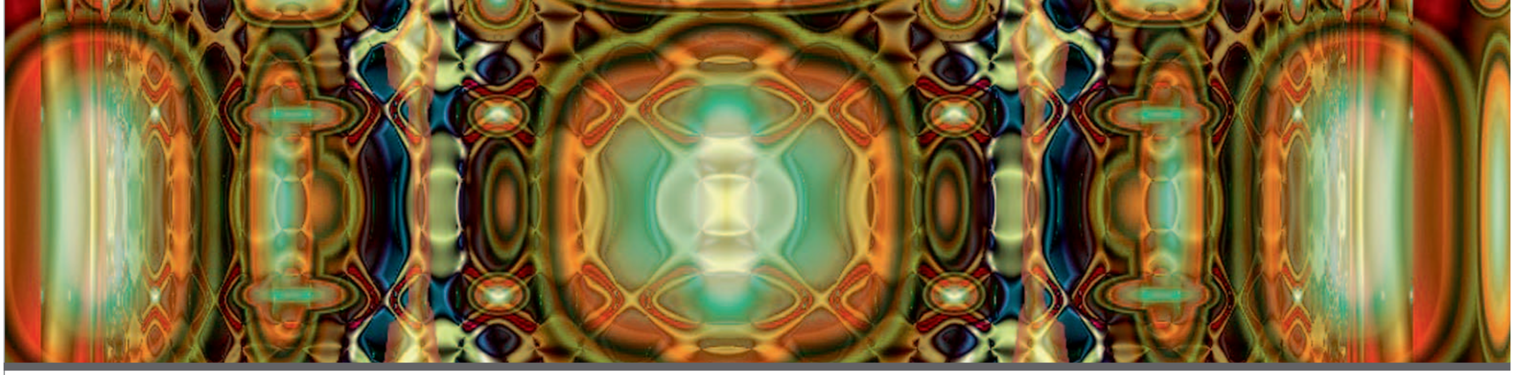
Business happens in real time.
If something impacts your business processes,
you need to know. *Now.*

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

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

NASTEL™
TECHNOLOGIES, INC.

Put Your Application Infrastructure On AutoPilot



WAS for z/OS Topology, Scalability, and Availability

Picking the best approach

BY LINFENG YU

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

WebSphere Application Server (WAS) for z/OS is IBM's z/OS implementation of the J2EE server design. To build robust and efficient WAS environments on z/OS, you must determine the topologies that address your needs. Since WAS for z/OS is middleware, the topologies can become quite complex and require forethought. Considerations should be given to scalability, availability, and factors such as security, performance, and administration.

ZSeries platform technologies have great impact on WAS for z/OS topology, scalability, and availability. Let's start with zSeries technologies.

Important zSeries Concepts To Understand

Today's zSeries Servers run in logical partition (LPAR) mode as shown in Figure 1. zSeries LPAR, delivered via the Processor Resource/System Manger (PR/SM), is a very mature server virtualization technology that improves server utilization and overall system availability. A physical zSeries box is split into multiple logical partitions. Each LPAR runs a distinct z/OS. The zSeries Intelligence Resource Director (IRD) can adjust logical resources across LPARs dynamically and automatically based on Workload Manager (WLM) and policy.

zSeries clustering technology, Parallel Sysplex, is based on a Non-Uniform Memory Access (NUMA) star clustering topology. The multi-system data-sharing technology enables up to 32 z/OS systems to be clustered with near-linear scalability.

The heart of the Parallel Sysplex is a special high-speed hardware construct, the zSeries Coupling Facility, which provides very fast communication among LPARs with latency of microsecond. It also maintains a global cache, global lock manager and shared communication area, all of which can be accessed and shared by z/OS subsystems.

Parallel Sysplex enables direct concurrent read/write access to shared data from all processing nodes in the configuration without sacrificing performance or data integrity. The CICSPlex, DB2 data-sharing group and WebSphere MQ queue-sharing group leverage Parallel Sysplex technology to achieve high scalability, continuous availability, failover, and load balance.

Dynamic Virtual IP Address (VIPA) and Sysplex Distributor are two basic z/OS TCP/IP technologies used by Parallel Sysplex for workload distribution and continuous availability.

Resource Recover Services (RRS) and Automatic Restart Management (ARM) are zSeries technologies for fast failover and recovery.

z/OS has a built in Unix System Service (USS) with an Hierarchy File System (HFS). All the WAS configuration

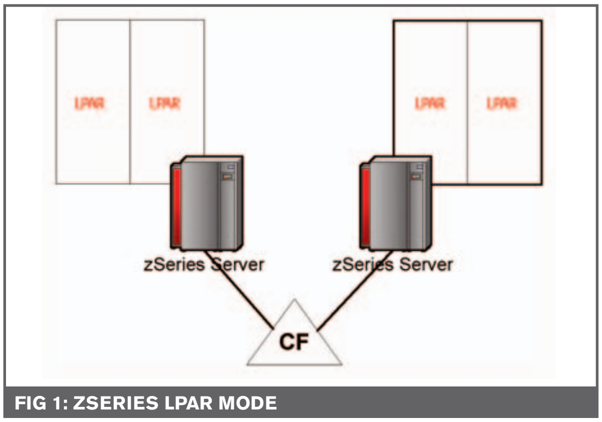


FIG 1: ZSERIES LPAR MODE

files for z/OS are held in the HFS.

That's pretty much all you need to know about IBM's zSeries server. The rest of this article will discuss various WAS topologies for z/OS. Along with the topology discussion, the scalability and availability implication of a topology will be explored too.

WAS for z/OS Building Blocks

Starting with version 5, WAS for z/OS was equivalent to the WAS Network Deployment (ND) for distributed platforms. If a specific version isn't mentioned, the discussion here applies to version 5.x. Figure 2 shows a WAS ND cell on z/OS. It has similar building blocks:

- **Server** – the basic unit where applications execute.
- **Nodes** – the collection of servers on a given z/OS image.
- **Node Agents** – a special-purpose server that acts on behalf of the administrative application to make changes in the node.
- **Deployment Manger** – a special-purpose server that runs the administrative application
- **Cell** – a collection of nodes managed by the Deployment Manager. A cell is the management boundary of WAS for z/OS.

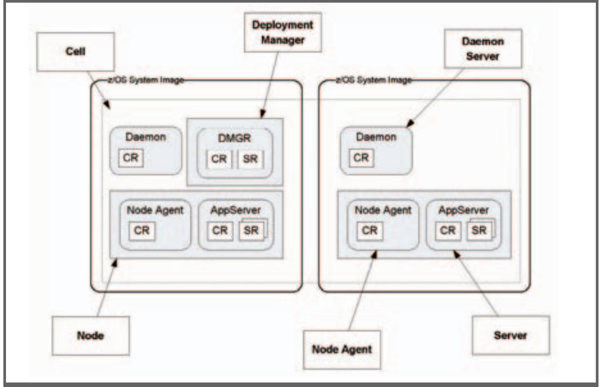


FIG 2: WAS FOR Z/OS BUILDING BLOCKS

WAS for z/OS can be configured as a standalone server as well. A standalone server is a single-application server that's not part of a Deployment Manager's cell. As shown in Figure 3, it's also a node and a cell. We're going to call it a base cell.

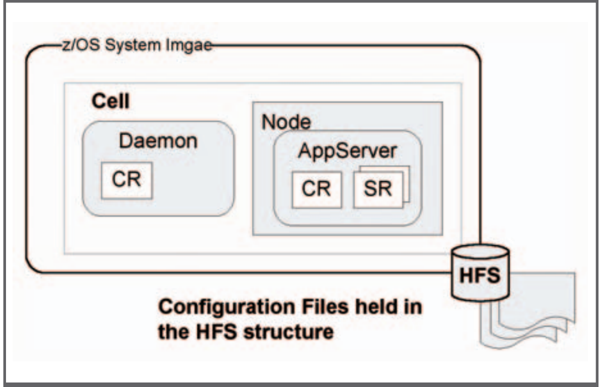


FIG 3: WAS FOR Z/OS STANDALONE SERVER

WAS Topology in Single LPAR

Three basic topologies can be used to configure WAS for z/SO in a single LPAR:

- Multiple base cells each with its own node and application server
- A single base cell having a single node with multiple application servers
- A single deployment manager cell

As shown in Figure 4, multiple base cells are configured in the same LPAR. This topology provides several basic WAS for z/OS runtime environments. It can only scale up by adding servant regions. Each building block of a base cell, LPAR, and related subsystems such as CICS, DB2 and WebSphere MQ is a single point of failure (SPOF). Obviously these base cells can't be used as a production environment.

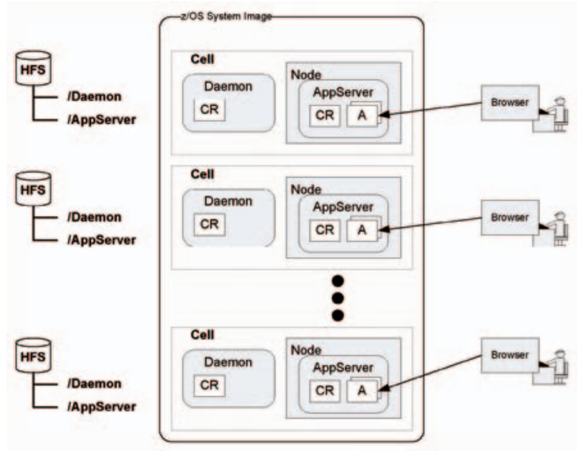


FIG 4: MULTIPLE WAS FOR Z/OS BASE CELLS IN A SINGLE LPAR

So what's this configuration good for? These base cells provide the most isolation between application servers. Each server has its own administration console. Resources for each node are separated. Large zSeries shops use this configuration as their development environment. Each development group can do whatever it wants in its own base cell.

Looking at Figure 5, a base cell having a single node with multiple application servers is configured in a single LPAR. It's a valid configuration. Compared to multiple base cell configurations, this configuration uses fewer resources. However, this topology isn't recommended. Since this configuration doesn't have a deployment manager; the administrative application runs in the first server created. It has only limited control over the other servers. Plus, key WAS functions such as clustering and some JNDI lookups aren't possible. The scalability and availability of this topology are the same as the base cell's.

An ND cell configuration in a single LPAR is shown in Figure 6. This is the most optimal topology for a single LPAR environment.

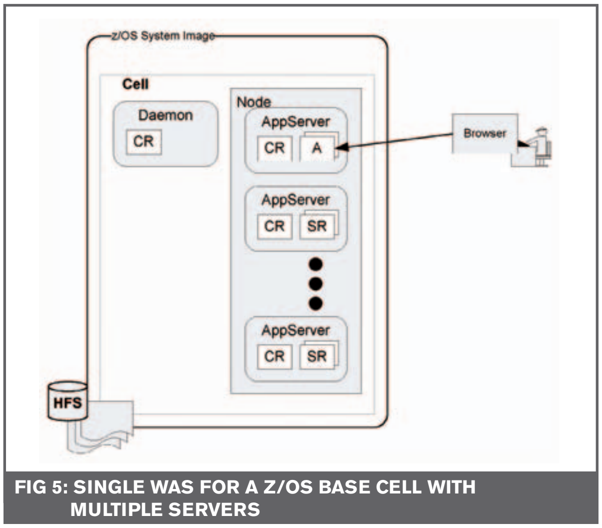


FIG 5: SINGLE WAS FOR A Z/OS BASE CELL WITH MULTIPLE SERVERS

The Deployment Manager is the central point of administrative control. The Node Agent is responsible for administration at the node level. All application servers and applications are under the control of the administrative application running in the Deployment Manager.

Since everything is configured in a single LPAR, scale-up is the only way to scale in this environment. Overall availability depends on the availability of the LPAR, related subsystems such as CICS, DB2 and WebSphere MQ, and the application servers in the cell. A vertical cluster can be created to remove single points of failure of the application servers in the cluster.

If planned and configured properly, this environment can be expanded to cross multiple LPARs leveraging Parallel Sysplex's near-linear scalability and high availability. Since the topology can produce a complete ND cell with non-production scalability and availability, it's normally used as a QA/Testing environment.

Generally speaking zSeries LPARs and subsystems are fairly stable. So some companies run production applications in a single LPAR ND cell for cost reasons if they can stand the system scalability and maintenance downtime.

It may not be a good idea to share a single LAPR ND cell among multiple development groups as a development environment. A ND cell is the least isolated from the development groups because the administrative application can control all the servers and applications in the cell. Worse, only one user can deploy applications and make server configuration changes at any given time. Otherwise, the configuration files could crash.

It's not a good idea to create two or more nodes in the same cell on a single LPAR. It doesn't provide any greater degree of application isolation than that afforded by separate application servers. It doesn't provide any greater degree of administration isolation afforded by one node. It doesn't allow applying maintenance node by node. And it results two node agents getting created; a node agent consumes significant memory.

All possible WAS configurations in a single LPAR use zWLM for workload management, and RRS and ARM for fast recovery from subsystem failure (in the single LPAR). If you'd like to leverage the zSeries's near-linear scalability and continuous availability, Parallel Sysplex must be brought into the picture.

WAS Topology in a Sysplex Environment

A WAS for z/OS ND cell can be configured across LPARs in a (Parallel) Sysplex. Figure 7 shows the topology. Like the ND cell in a single LPAR, the Deployment Manager and Node Agent work together providing a central point of managing and controlling all the application servers and applications. In Figure 7, DB2 and WebSphere MQ subsystems co-locate with application servers in the same LPAR. This is not the only choice. But in general it boosts performance. I'll cover this in subsequent articles.

Horizontal clusters can be created across LPARs for scale-out, failover, workload balance, and continuous availability (especially when the LPARs are on different physical boxes).

Sysplex distributor with dynamic VIPA works with

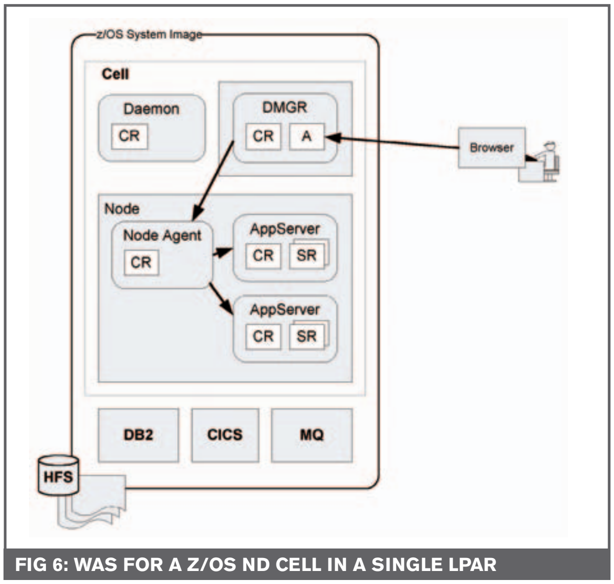


FIG 6: WAS FOR A Z/OS ND CELL IN A SINGLE LPAR

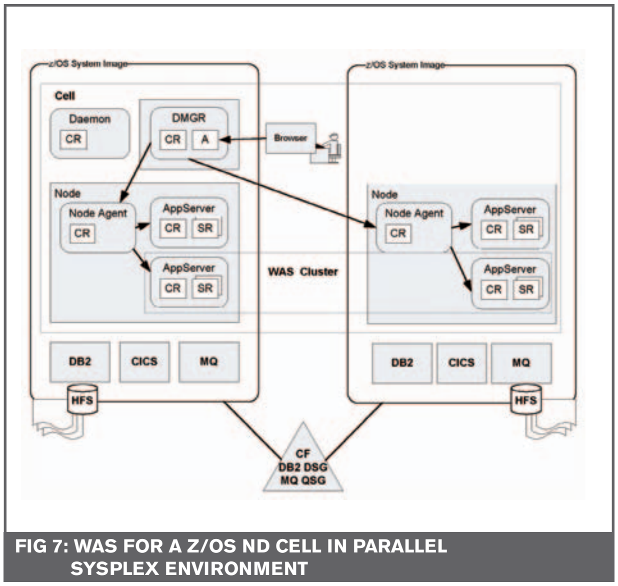


FIG 7: WAS FOR A Z/OS ND CELL IN PARALLEL SYSPLEX ENVIRONMENT

zWLM to determine how to direct inbound requests. The most suitable server (workload and resource wise) is chosen to process the work request.

If one of the servers isn't available, the requests will be routed to other servers in the same cluster. The work can be absorbed because in a truly heterogeneous application environment the zWLM and IRD can distinguish and dynamically adjust resource deployment to meet the goals of critical work.

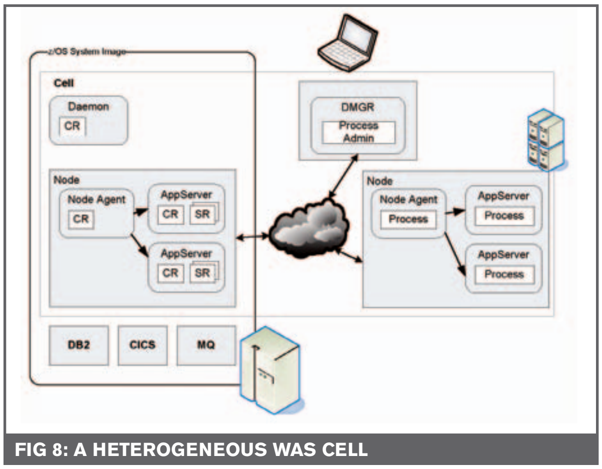


FIG 8: A HETEROGENEOUS WAS CELL

To scale out and remove SPOF, applications have to be able to access the same database and/or message queues from the different LPARs. The CICSPlex, DB2 data-sharing group and WebSphere MQ queue-sharing group need to be created. Parallel Sysplex technology makes sure the data sharing doesn't sacrifice performance and data integrity.

Data sharing in the Parallel Sysplex also provides a significant availability benefit, protecting against the loss of data and availability in light of planned and unplanned outages by enabling workloads to be shifted among the LPARs as needed to maintain the business goal.

This configuration allows rolling-out of WAS maintenances and applications if the install HFS and configuration HFS are planned and implemented correctly. This way, WAS and application maintenances and upgrades downtime can be avoided. Should you need more information, please refer to the resources.

Since this configuration leverages all the Parallel Sysplex features, it's recommended as the WAS for a z/OS production environment. If possible, the production ND cell should be configured across three LPARs. If one of the LPARs is taken down for maintenance, the other two are still be available without compromising any QoS.

Obviously the system is more complicated and consumes more resources. It may be too expensive for some companies.

There's almost no limit to the number of cells you can build in a (Parallel) Sysplex. The resources used by a WAS for z/OS aren't trivial. You shouldn't create too many ND cells unless the cell-to-cell isolation is really needed.

WAS for z/OS V6.0.1 introduces a new HA manager. It provides a continuous availability solution for singleton services in WAS for z/OS. It complements the Parallel Sysplex's continuous availability solution. It works with all the topologies discussed so far.

Other Topologies

Under certain condition and configurations, WAS for z/OS V5.0.x, V5.1 and V6.0.1 can co-exist in the same cell.

- WAS for z/OS v5.0.x, and V5.1 nodes can co-exist in the same cell if and only if they are on different LPARs.
- WAS for z/OS V5.0.x and V5.1 nodes can co-exist in the same LPAR if and only if they are in different cells
- WAS for z/OS V5.0.x and V6.0.1 nodes can co-exist in the same cell if and only if they are on different LPARs.
- WAS for z/OS V5.0.x and V6.0.1 nodes can co-exist in the same LPAR if and only if they are in different cells.
- WAS for z/OS V5.1 and V6.0.1 nodes can co-exist on the same LPAR in the same or in different cells.

WAS for z/OS V6.0.1 can also have cells consisting of nodes on different platforms as shown in Figure 8. The distributed security model and trans-operating system transactions are somewhat complex. You have to understand the implications of the design before you decide to use it.

Conclusion

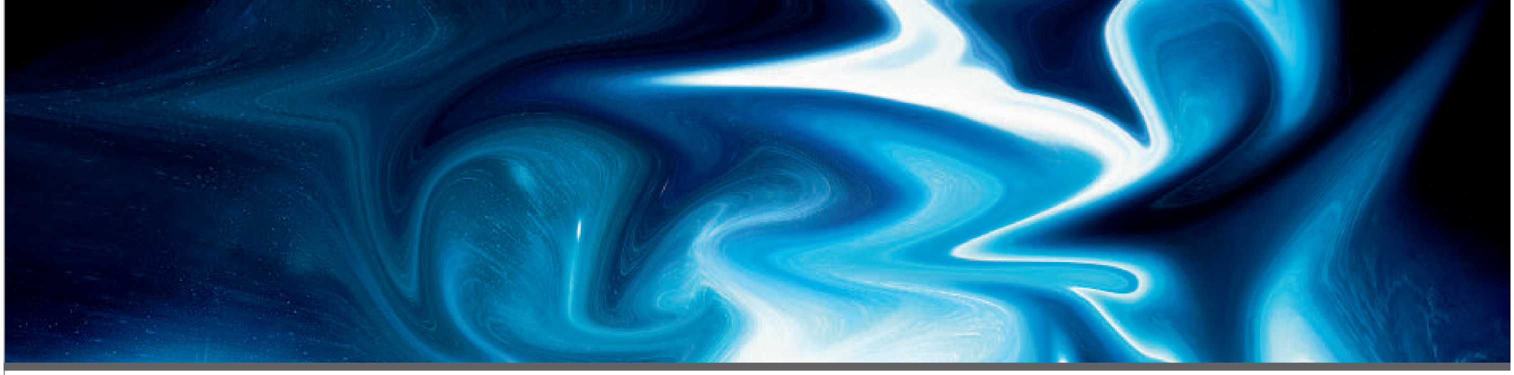
The WAS for z/OS topologies on a single LAPR is more suitable for development and testing. Should you need a highly scalable and continuously available environment, you should consider configuring ND cells across the LPARs in a (Parallel) Sysplex.

Unfortunately we don't have the space to discuss the edge components for a WAS for z/OS. If you're interested in edge components, please refer to the resources.

Should you need a step-by-step installation and configuration guide of WAS for z/OS, please refer to the resources.

Resources

- WAS for z/OS information center: http://publib.boulder.ibm.com/infocenter/wasinfo/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/welcome_zos.html http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/welcome_zos.html http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/welcome_nd.html
- Washington Systems Center Sample WebSphere for z/OS ND Configuration http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100367
- WebSphere Application Server for z/OS V5 - Planning for Test, Production and Maintenance http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100396
- WBSR6 - "Gen 6" workshop handouts http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS1290



The Power of Process, the Benefit of BPEL

Taking advantage of IBM's BPEL extensions

BY BJ GRAU

BJ Grau is a key member of Prolifics' highly specialized team of IBM WebSphere experts retained to architect, build and troubleshoot custom WebSphere, Portal and Business Integration solutions. Specializing in enterprise architectures, WebSphere administration, performance tuning, portal development, and SOA development, BJ has serviced several WebSphere customers including Lockheed Martin, Century 21, State of Florida, Blue Cross/Blue Shield, Office Depot, Royal Caribbean, Celebrity Cruise Lines, Harcourt, Siemens, and SAIC. BJ has also taught classes on Java and J2EE.

bgrau@prolifics.com

Business Process Execution Language (BPEL) is a non-proprietary, XML-based language developed by a group of major vendors including IBM, BEA, and Microsoft. Its function is to describe how Web Services interact with one another to form workflow systems that implement business processes. As a model and grammar for describing this behavior, it lets you implement complex processes by creating and wiring together different activities such as invoking a service, manipulating data, throwing faults, or terminating processes. With BPEL you can take a structured top-down approach to implementing a Service-Oriented Architecture (SOA).

The BPEL specification, like most first editions, is rather basic. While extremely powerful and flexible, it also had to gain agreement from all of its many contributors, and so it lacks some important features that make it more practical. Here is where IBM has distinguished itself by creating a useful set of extensions to the specification that increases BPEL's utility at least fivefold, making it immediately practical. They correspondingly enhanced the tooling to enable the highly effective development of BPEL-based business processes and services choreography.

Mostly likely, if you've been using BPEL with IBM's tools, you've taken many of the BPEL extensions for granted. This article offers an overview of these extensions, illustrates their use, and discusses the impact they have on service choreography. By the end of this article, one should have a better understanding of which elements are actually BPEL extensions — as opposed to regular elements of the specification — why they were made, what they were intended for, and how to take advantage of them.

Types of Extensions

BPEL extensions can be separated into two general groups, new entities and their entity extensions; and new attributes for existing entities and their attribute extensions. While there are only two extensions in the former group, there are a slew in the latter. Both types offer profound flexibility and convenience in designing processes, and without them it would be extremely awkward – if not impossible – to implement some of the capabilities they provide.

Entity Extensions

JAVA SNIPPET

As the name implies, a Java snippet allows a piece of Java code to be included in a process (see Figure 1). It doesn't make it into the BPEL specification because it's language-specific and including it would violate BPEL's goal of remaining completely language- and vendor-agnostic.

Java expressions, which are similar to Java snippets except that they aren't entities on their own, are useful in letting us include Java code that does work such as performing calculations and providing logic for other BPEL constructs like while, switch, and wait activities as well as links and joins.

It's important to note that Java snippets also provide a performance benefit by enabling the execution of local Java-based components that might otherwise have to be invoked in a high-overhead fashion such as a Web Service using a SOAP call (see Figure 2). However you have to consider that although there's a performance gain when the overhead of a SOAP invocation is removed, a component called directly through a Java snippet doesn't follow the same Web Service invocation semantics as the other service components of a BPEL process. This creates inconsistency that can lead to a higher maintenance cost later if the component eventually becomes remote to the process, exposed as a SOAP/HTTP service, etc., and can no longer be called through a Java snippet.

Another option that provides the same benefit is the Web Services Invocation Framework (WSIF). WSIF lets you invoke a Web Service regardless of the technology used to invoke the service. As long as the service can be described by WSDL, it can be accessed through WSIF, even if it's not actually invoked through a SOAP call. This means the service could be called through some other means, such as RMI/IIOP or even a local method call. IBM tools and software, such as WebSphere Studio Integration Edition and WebSphere Business Integration Server Foundation, provide built-in WSIF support.

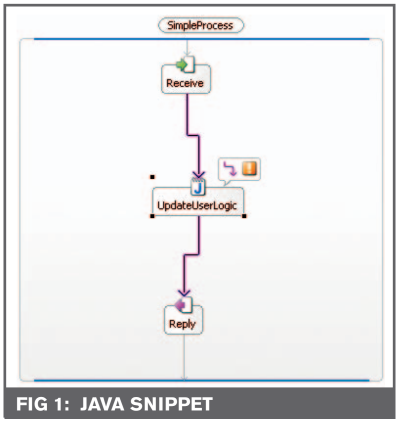


FIG 1: JAVA SNIPPET

STAFF ACTIVITY

A Staff Activity is an extension of an activity that provides a means to incorporate human interaction into a BPEL process (see Figure 3). No matter how much we automate our world, there will always be a need for human intervention and human interaction is often the norm in a business process. You can build a sleek, super-automated business process that runs on its own 99% of the time, but there are still times when human intervention and judgment are required.

For example, suppose a business process is built to automate loan approval workflow for a bank. A request for a loan can come with associated documentation, such as the customer's personal financial information, that will have to be verified. At this point a bank representative has to examine the documentation by hand and decide where to route the business process based on his analysis.

Here is where the Staff Activity fits in. One important limitation to note is that only an interruptible process can include a staff activity as one of its elements because the activity implies interruption. The process can then asynchronously interact with a human as illustrated in the loan example.

The Staff Activity serves a very important function that will be required in many business processes. Perhaps it should have been in the original BPEL specification. Perhaps it will be included in the next version.

Attribute Extensions

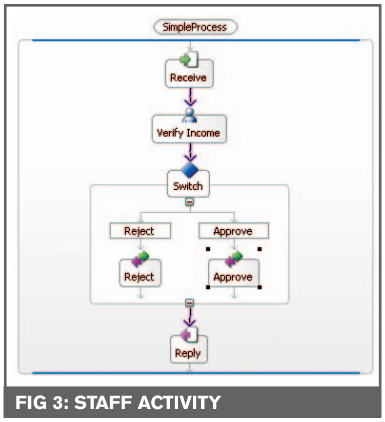
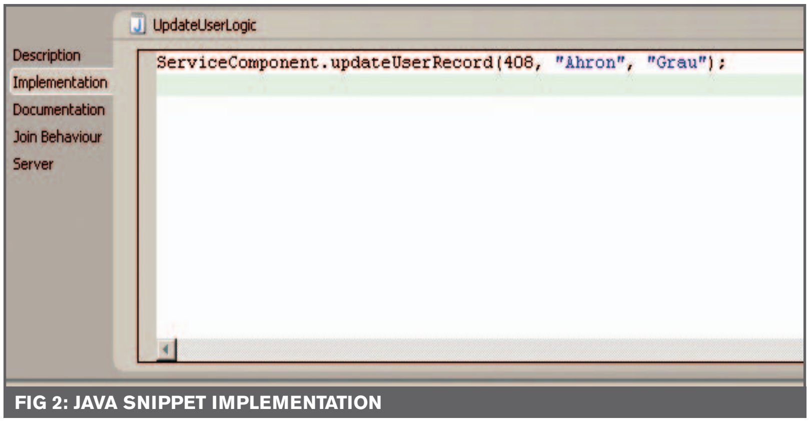


FIG 3: STAFF ACTIVITY

Autonomy Attribute: Reuse has long been a goal in the industry and it's considered a winning situation when it's achieved. Reusing an existing process, when done correctly, can pay benefits many times over. Fortunately it's very easy with BPEL processes and can be realized simply by having one process invoke another by using the invoke activity. However, it's important to understand the options available when doing this and to use them correctly.

Here we'll refer to the process that does the invoking as the calling process and the one that it invokes as the invoked process. Rules regarding the interruptible and non-interruptible properties of the processes apply in this case. If the calling process is interruptible, then the invoked process can be either interruptible or non-interruptible. If the calling process is non-interruptible, then the invoked process must also be non-interruptible for obvious reasons. An invoked process, if it's interruptible, has an extended attribute available to it called Autonomy that can take the values Peer and Child. Peer implies that the invoked process runs on its own without access to the calling process's context and that it is also outside of the calling process's compensation sphere. It's completely autonomous in this case. When the Autonomy attribute is set to Child, then the calling process is considered to be the parent and the invoked process a child. In this case the invoked (child) process shares



```
ServiceComponent.updateUserRecord(408, "Ahron", "Grau");
```

FIG 2: JAVA SNIPPET IMPLEMENTATION

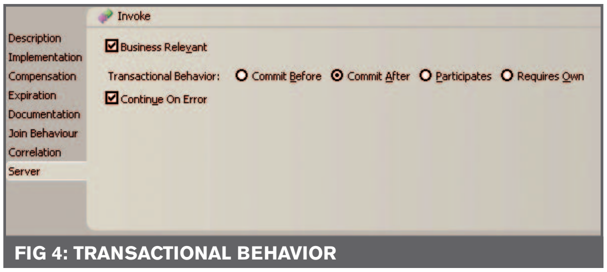
the context of the calling process (parent) and belongs to the same compensation sphere. When the Autonomy attribute is set to Child, the invoked process is effectively a sub-process nested inside the calling process. Sub-processes are sometimes referred to as micro flows.

The Autonomy attribute has an impact on what compensation support can be selected for a sub-process. If a sub-process's autonomy attribute is set to Peer, the valid compensation sphere settings are Not Supported or Requires New. If autonomy is set to Child, then only Supports or Required are valid.

ATTRIBUTES OF INVOKE ACTIVITY

Now that we've looked at an extended attribute closely tied to the Invoke Activity, let's examine some more attributes that belong directly to it: transactionalBehavior, undo, continueOnError, and expiration.

- An important aspect of the Invoke Activity is the transactional behavior associated with the invocation. This can be easily configured through the Transaction Behavior attribute (see Figure 4).
- Undo is the action taken to compensate on rollback. You won't actually see "undo" in the user interface of the WebSphere Process Choreographer, but it'll be created in the generated BPEL source if compensation is configured.
- The continueOnError extension is a useful flag that determines whether, on encountering an error, the invoke activity will go to the stopped state and cease executing the process, or merely transition to a failed state while the process continues. As its name implies, process execution continues if it's set to true.
- Finally, the expiration attribute is intended to prevent an Invoke Activity from blocking a process indefinitely while waiting for a response. Expiration can be set to a duration, date, or timeout (see Figure 5).

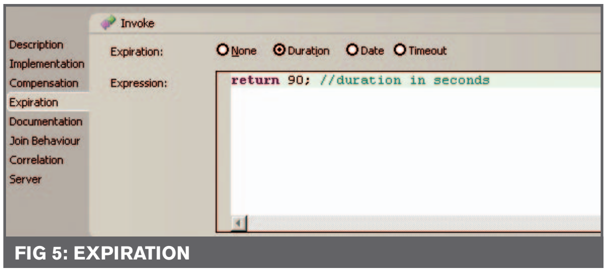


☑ Business Relevant

Transactional Behavior: ○ Commit Before ● Commit After ○ Participates ○ Requires Own

☑ Continue On Error

FIG 4: TRANSACTIONAL BEHAVIOR



Expiration: ○ None ● Duration ○ Date ○ Timeout

Expression:

```
return 90; //duration in seconds
```

FIG 5: EXPIRATION

ONALARM ATTRIBUTE OF PICK ACTIVITY

The BPEL specification provides a useful entity called a Pick Activity that's a hybrid between the Receive and Switch Activities. It waits for one out of a set of messages and continues the process with an activity chosen based on the first message received that matches one of the options in the Pick. Besides waiting for a specific message, the Pick activity can also continue execution based on the setting of an extended attribute named OnAlarm. OnAlarm can be configured to trigger based on a duration of time or a specific date. An example of where this may be useful is a situation in which you're waiting for a response from a Staff Activity and if the response isn't received by a certain date, the process needs to timeout. You can configure OnAlarm to trigger in a certain number of days and then send the process to a no-op, effectively timing the process out or making it start over (see Figures 6 and 7).

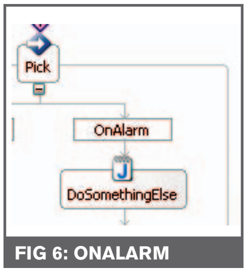
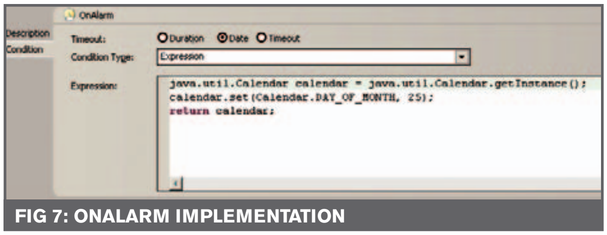


FIG 6: ONALARM



Timeout: ○ Duration ● Date ○ Timeout

Condition Type: Expression

Expression:

```
java.util.Calendar calendar = java.util.Calendar.getInstance();
calendar.set(Calendar.DAY_OF_MONTH, 25);
return calendar;
```

FIG 7: ONALARM IMPLEMENTATION

A JAVA EXPRESSION AS PART OF A WAIT ACTIVITY AND OTHER ENTITIES

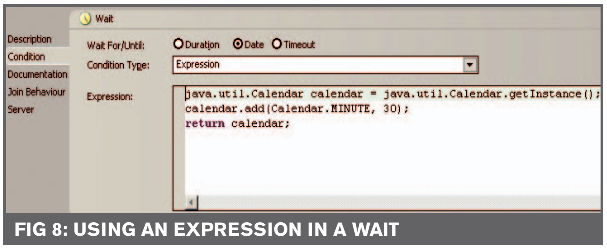
The base BPEL specification provides for enabling a Wait Activity either through duration of time or a specific date. The Wait Activity has been extended to allow for a Java Expression to calculate the amount of time to wait thus allowing for a more precise and flexible control in a Wait (see Figure 8). An example of where this could be useful would be in a retail order management system where you want to make the process wait to bill the customer for a certain number of days calculated by a formula based on the types and quantity of items ordered.

This same idea of flexible and granular control and decision making inside BPEL entities using a Java Expression also applies to the While and Switch Activities as well as Control links and Joins.

There are more extended attributes for Activity such as id, displayName, description, documentation, and layout. These are simple extensions without any profound impact, but they do make process choreography more convenient by adding meaningful information for human consumption and understanding.

THE BUSINESS RELEVANT ATTRIBUTE

So far we've examined extensions that provide for easier



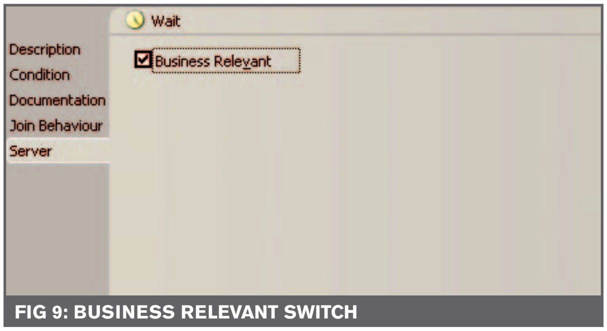
Wait For/Until: ○ Duration ● Date ○ Timeout

Condition Type: Expression

Expression:

```
java.util.Calendar calendar = java.util.Calendar.getInstance();
calendar.add(Calendar.MINUTE, 30);
return calendar;
```

FIG 8: USING AN EXPRESSION IN A WAIT



☑ Business Relevant

FIG 9: BUSINESS RELEVANT SWITCH

development and execution. A final attribute that we'll examine enables detailed and effective runtime monitoring of processes. The attribute that makes all of this possible is the Business Relevant attribute of variables, processes, links, and activities (see Figure 9). This powerful attribute can be used by tools such as the Business Process Engine in the WebSphere Business Integration Server Foundation to create a durable and persistent audit trail. This is done by logging certain events to an audit log table that can be examined to paint a vivid picture of the life of a running process including what events may have led to its failure. Events that can be logged, assuming the business relevant switch has been set for the entity, are numerous. The process instance events that can be logged are installation and starting a process, completion, failing, failed, terminating, terminated, compensating, compensation in doubt, successful compensation, initialization of correlation sets, and process deletion. You can see how much information can be gathered from the audit log table just from the audit events of a process instance. Adding even more power to the set of data that can be gathered are the audit events for activities, variables, and links.

Conclusion

The goal of this article was to illustrate the powerful benefits of IBM's BPEL extensions. While some are specific to IBM tooling and are unlikely to be adopted in the specification, others are non-proprietary and offer so much value such that they may be embraced. As the primary BPEL contributors create similar extensions and the user community uses these features, adoption becomes more likely.

More and more companies today are moving toward a Service Oriented Architecture that will enable their business to become more agile and adaptable. BPEL is the execution aspect required to implement such architecture. Together BPEL and IBM's BPEL extensions offer a powerful integration model to automate process integration in the enterprise and with partners.

WebSphere JOURNAL Coming Next Month...

Composite Applications, Part III

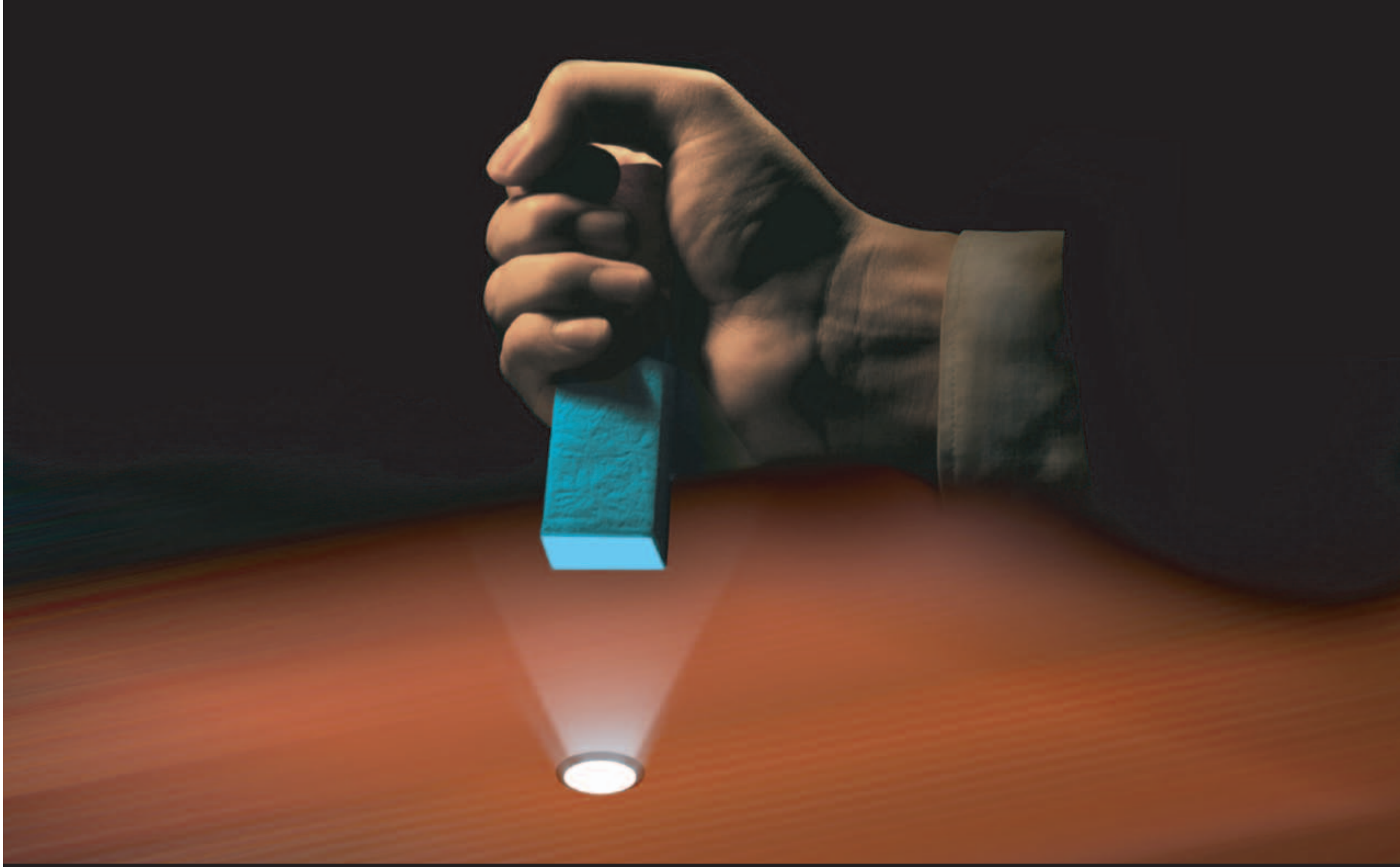
WAS for DB2

Maintaining Visibility and Control

High-Performance Server-Side SOAP

WebSphere JOURNAL Advertiser Index...

COMPANY	URL	PHONE	PG
BLOG-N-PLAY.COM	WWW.BLOG-N-PLAY.COM	201-802-3000	17
ISSJ	WWW.ISSJOURNAL.COM	888-303-5282	49
INSTANTIATIONS	WWW.INSTANTIATIONS.COM/ANALYTIX	800-808-3737	13
ITEMFIELD	WWW.ITEMFIELD.COM/WBI	800-838-1055	21
ITKO	WWW.ITKO.COM	877-BUY-ITKO	51
JINFONET	WWW.JINFONET.COM	301-838-5560	33
LINUXWORLD CONFERENCE	WWW.LINUXWORLDEXPO.COM	800-657-1474	45
MOTIVE	WWW.MOTIVE.COM/WITHIN1	512-531-2527	52
NASTEL TECHNOLOGIES	WWW.NASTEL.COM/UNCOMPLICATE	800-580-2344	23
PARASOFT	WWW.PARASOFT.COM/SOAPTEST_WSD	888-305-0041	2
PROLIFICS	WWW.PROLIFICS.COM	800-675-5419	3
SYS-CON NEWSLETTERS	WWW.SYS-CON.COM	888-303-5282	15
SYS-CON REPRINTS	WWW.SYS-CON.COM	888-303-5282	11
TANGOSOL	WWW.TANGOSOL.COM	617-623-5782	31
WEEKEND WTH EXPERTS	WWW.WEEKENDWITHEXPERTS.COM	732-598-4027	39
WILY TECHNOLOGY	WWW.WILYTECH.COM	888-GET-WILY	5



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

Think JReport.

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

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

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

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

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

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

JReport

JINFONET SOFTWARE

Caching can be implemented in portlets

Memory Caching in WAS

BY:POOJA GUPTA & SHERMAN LEE

Pooja Gupta is an IT Specialist with GeT(Global eTransformation), IBM Software Group. She has seven plus years of experience in IT field. She holds a Bachelor degree in Computer Science and is an IBM Certified WebSphere Portal 5.0 Solution Developer. Her area of expertise include working on WebSphere products and end to end development of J2EE architecture based web applications.
pooja_gupta@us.ibm.com

Once upon a time if a development team wanted to implement memory caching, the developers had to code a Java Singleton class and maintain key value pairs using the Java API's hashtable in memory.

The IT industry has evolved and so has the caching mechanism. WebSphere Application Server (WAS) 5.x now comes with build-in, easy-to-use caching APIs. By implementing WAS's built-in caching feature, developers can configure caching for their application in a couple of hours. The WAS caching feature is quite sleek since it comes with cache monitor (which needs to be installed, but it's a very easy installation). This cache monitor helps developers view the cache contents, clear the cache, and so on.

In this article I want to educate developers to use WAS's built-in caching mechanism. I'll also include a snippet of code and show you how easily it can be implemented in a portlets/Java program.

WebSphere Application Server 5.x and above offer system defined APIs to implement Memory Cache/Command Cache. The cacheable commands are stored in the cache for reuse as and when pages are repeatedly requested. Each incoming request is cached as an object and is associated with an unique cache ID. These IDs can be generated based on the methods and fields (variables) defined in the command as input parameters.

Command Caching can be implemented in portlets since portlets run on Portal Server, which in turn makes use of an instance of the WAS.

The following sample program will show how to implement and integrate the command cache with portlets. This code was developed using Rational Application Developer (RAD).

1.First and foremost the jar files that come with the product should be included with the program in Listing 1 (else the code will not compile) (there are three of them and can be found in the following dir or WAS.):
- Installdir WAS/lib/commands.jar
- Installdir WAS/lib/ distexcep,jar
- Installdir WAS/lib/ dynacache.jar

2. Implement a cache command interface as shown in Listing 2
3. Now create a regular servlet or a portlet that will run on WAS or WPS, respectively. (All the code in this case is in the same package, not mandatory to have all the code under same package) The following snippet of code (Listing 1) can be placed in Servlet or a Portlet (doGet or doView method accordingly)

Listing 1

```
try
{
//Reading a xml file
SampleCacheCommnad scc=new
SampleCacheCommnad(“c:/websphere/
portalserver/shared/app/sample.
xml”);
scc.execute();
String XMLDoc=scc.getXmlresult();
System.out.println(“Command Cache
working... “);
}
  catch(Exception e)
{
  System.out.println(“The command
cache exception”+e);
}
```

4. Now, we'll have to place the cachespec.xml with the servlet/portlet project. This xml file is placed in the web-inf folder. (This folder is created when we create a portlet project using the RAD.) Place the following entry in cachespec.xml (the cachespec.dtd can be found in the WAS Install dir\properties folder. (With this entry, we specified basis on which caching will happen within WAS)

```
<?xml version=”1.0”?>
<!DOCTYPE cache SYSTEM “cachespec.
dtd”>
<cache>
<cache-entry>
        <class>command</class>
        <sharing-policy>not-
shared</sharing-policy>
        <name>com.ibm.websphere.
sample.SampleCacheCommand</name>
        <cache-id>
                <component
type=”method” id=”getFilename”>

    <required>true</required>
                </component>
                <priority>2</
```

```
priority>
                 <timeout>3600</
timeout>
        </cache-id>
  </cache-entry>
</cache>
```

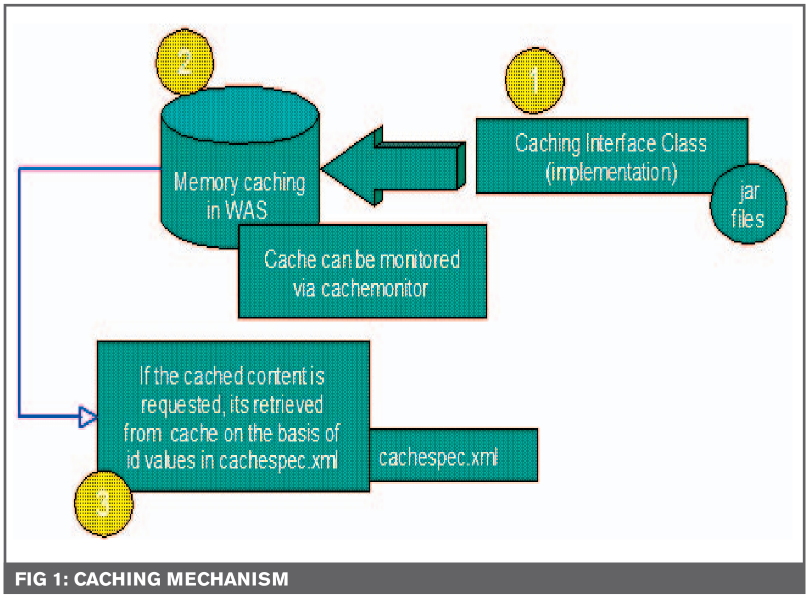

2
1
Memory caching in WAS
Caching Interface Class (implementation)
jar files
Cache can be monitored via cachemonitor
If the cached content is requested, its retrieved from cache on the basis of id values in cachespec.xml
cachespec.xml
3

FIG 1: CACHING MECHANISM

(To get the XML allowed tag values, refer to the WAS infocenter.)

After implementing this step the command cache is ready to use with the servlet/portlet. This sample code will cache the value of xmlResults on the basis of the fileName. If the fileName remains unchanged and chache is valid, the request will be addressed from the cache.

Cached results can be viewed and managed via cache monitor. The cache monitor provides a GUI interface to change the cached data manually. The monitor has to be installed manually using the CacheMonitor.ear file (in the Installed WAS dir/installableapps) through the WAS Admin GUI.

This installable Web application provides a real-time view of the state of the dynamic cache and can be found in the install_root/installableApps directory. The only way to manipulate the data in the cache is by using the cache monitor.

After installation, the cached monitor can be accessed as: http://your host_name:your port_number/cachemonitor.

The diagram in Figure 1 represents the functional interaction code.

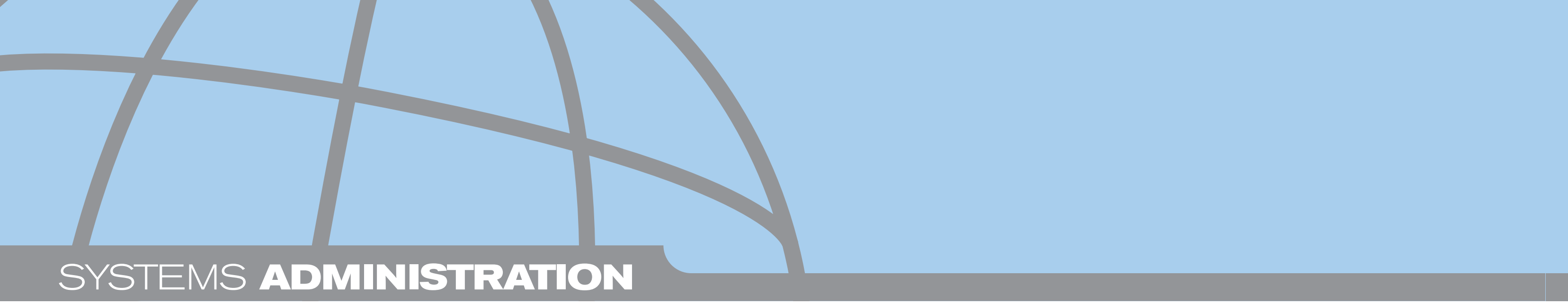
Acknowledgements

- WAS Infocenter http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp

LISTING 2

```
/** Created on May 24, 2005**/
package com.ibm.websphere.sample
import com.ibm.websphere.command.CacheableCommandImpl;

/** @author Pooja Gupta**/

import java.util.*;
import com.ibm.websphere.command.CacheableCommandImpl;

public class SampleCacheCommand extends
CacheableCommandImpl {

   private String fileName = null;
   private String xmlResults=null;

   public SampleCacheCommand () { }   //empty constructor
   //one argument constructor
   public SampleCacheCommand  (String fileName) {
         this();
         this.fileName = fileName;
         }
   /**
    * @see com.ibm.websphere.command.Command#isReadyToCallE
xecute()
    */
   public boolean isReadyToCallExecute() {
         return fileName != null;
   }

   /**
    * @see com.ibm.websphere.command.TargetableCommand#perf
ormExecute()
     */
   public void performExecute() throws Exception
   {
         System.out.println(“In the perform Execute of
SampleCacheCommnad”);
         //Reading of the file
         try
         {
         System.out.println(“In the try loop of Cache
Command”);
         xmlResults=readfileclass.readFile(getFilename(),”
”);   // I am calling a function to call readFile from the
calss readfileclass to read in an XML file, you can cus-
tomize this according to your needs...whatever needs to be
cached goes in here
         System.out.println(“the results are “+xmlRe-
sults);
         }
         catch(Exception e)
         {System.out.println(“The value of exception “
+e);}
         }
   //set a setter for the XMLResults variable
   public void setXmlresult(String xmlResults) {
         this.xmlResults = xmlResults;
   }
         //set a getter for the XMLResults variable
   public String getXmlresult() {
         return this.xmlResults;
   }
   //setter for the filename that needs to be read
   public void setFilename(String fileName) {
         this. fileName =  fileName;
   }

   //getter for the filename that needs to be read
   public String getFilename() {
         return this.fileName;
   }
}//end of class
```

Sherman Lee is a Consulting I/T Specialist with the IBM Software Group. Since joining the IBM Software Group from IBM Global Services where he previously served as an infrastructure architect, Sherman has been a technical leader in the Upstate NY/New England area covering WebSphere technologies and more recently specializing within the Portal and IBM Workplace products. Sherman holds a Bachelor degree in Computer Engineering and is also an IBM Certified professional with over 10 years experience.



Leverage, evolution, and flexibility are key

Modernizing Legacy iSeries Systems

BY: MIKE MARLOWE

Mike Marlowe leads the pre-sales team at ASNA as a pre-sales engineer andWindows and Web development trainer specializing in both .NET- and COM-based technologies. Mike is instrumental in educating prospects on ASNA's technology message as well as assisting with conceptual applicationplanning and development. Mike also assists with developing and deployingASNA's training curriculum.

marketing@asna.com

IBM iSeries systems are installed in an estimated 250,000 organizations. If yours is one of them, then most likely your group is either interested in or committed to modernization, which could mean you wish to enhance the user interface while maintaining the existing core business logic, or to integrate data on an AS/400 with Windows-centric, SQL-based back-office applications, or to implement a service-oriented architecture (SOA) and offer Web services to better serve your customers and your enterprise's own competitiveness.

Whatever the level of modernization you are currently seeking, the existence of the iSeries presents both challenges and opportunities. The challenge is to determine the best way to bring the enterprise system into the 21st century without incurring unnecessary disruption to the business, while managing costs and achieving a reasonable timeline. The opportunity is to leverage the valuable asset that is the iSeries itself – including its reliability, performance, ease of use, and familiarity – along with the associated assets in existing data, developer-born applications, and staff expertise.

The options for modernization are many. To evaluate and select among the possibilities, an articulated set of goals and an awareness of the assets to be leveraged and preserved will be valuable. These goals and assets are interrelated, in the sense that among your goals, it is important to consider the ability to *maximize* iSeries assets. You may consider these assets in terms of the literal building blocks of the system: RPG programs, CL programs, DDS display, print files, data files, and so forth. For a more enterprise-wide perspective, however, also consider your assets in terms of the "four pillars" of an organization's IT "ecosystem." These are:

1. The *people* (your existing programming staff, including RPG programmers, C# people, VB folks, and so forth, as well as anyone else who interacts with, is affected by, or has expertise in the system AND the company's business)
2. The *applications* (which represent the sum total of your company's IT existence, and are often complex and comprehensive – but whose value erodes when they are not readily extended to other applications, interfaces, and Web services)
3. The *processes* (which depend on IT technologies for support, and benefit from leaner, integrated methodologies – but these require change and adaptation throughout the organization)
4. The *data* (both on the iSeries and on Windows-based or any other system you may already have)

Each of these categories of asset is important not only to the iSeries system but to the organization as a whole. Each represents an investment that can be either lost, preserved, or enhanced through the modernization initiative. All four merit careful consideration as to the way they may be impacted by, respond to, or contribute to the changes you are considering.

With this insight into your organization, your goals, and your assets, you can begin to consider the modernization options, ideally without becoming mired in the plethora of possibilities.

Java/WebSphere or .NET?

Although this oversimplifies the situation, the looming question at the outset is whether to proceed with a Java/WebSphere solution or to consider an alternative – that is to say, .NET – for your Web/SOA realm. IBM's original iSeries Developer Roadmap was certainly intended to steer iSeries shops toward a Java-centric future. The disadvantages to this route, however, have proven so great that even IBM has backed off from its insistence on Java/WebSphere and begun to recognize a role for third-party solutions.

For large enterprises that can bear the risk and costs of Java projects, the Java/WebSphere route may make sense. For small and midrange companies (which make up the majority of OS/400 shops), budget, staff, and time considerations require them to be more circumspect. They must consider alternatives that decrease the risks and costs of switching to an unfamiliar language, retraining or hiring new programmers, and rewriting reams of code.

Microsoft's .NET platform has emerged as an alternative modernization path for iSeries shops that are looking for a faster, less expensive, and less cumbersome route. Adoption of .NET is particularly of interest to companies that are implementing Web services and SOA. In addition, because it embraces many programming languages, it provides flexibility and advantages in terms of cost and staff considerations – developers can use whatever language is most appropriate for them and the project.

The Options

Although Java/WebSphere and .NET are the routes of choice for many enterprises that are undertaking modernization of iSeries systems, these are not the only alternatives. All of the following are means by which an organization can achieve a degree of modernization (or at least the veneer of it):

- Replace or rewrite applications
- RPG/400 emulation
- RPG/400 screen scrape
- Host-based ad hoc
- Migration to Java/C#
- RPG portfolio transformation

In reality, these options are not mutually exclusive. An organization will choose the path that is best for the job and for the enterprise, and sometimes that means a *combination* of alternatives. This may mean the use of DB2/400 databases, SQL databases, Windows Servers, and IBM iSeries eServers in any combination according to requirements and goals.

That said, let's consider each of the modernization options in turn, and its effects on the "four pillars" mentioned above – people, applications, processes, and data. As you read about each one, keep in mind not just these four pillars but also these good pieces of advice:

> "Whatever the level of modernization you are currently seeking, the existence of the iSeries presents both challenges and opportunities"

- *Remember the meaning of modernization.* IBM's definition of "modernization" is: "the enhancement, conversion, or rewriting of iSeries applications to include new technological capabilities that did not exist when the applications were originally developed." This is the target, but it must be achieved in the context of your business's specific circumstances and goals.
- *Know where you are going.* Consider the evolution of processes, the ease of integration, and the way people will fit into this ongoing evolution. As you develop plans for modernization, base these on a vision for the future of your business, and be certain that the new system will facilitate evolution toward that vision.
- *Don't compromise.* The iSeries is robust and stable – your target platform should be no less. Organizations that have "grown up" with an iSeries appreciate its safe environment of almost never-fail hardware (iSeries systems rarely crash) and resistance to hacking or security issues (few iSeries systems are hacked or become prone to virus attacks). Given that the applications running on the iSeries are the critical back-office applications of an enterprise – ERP, manufacturing and supply chain, and so forth—it is reasonable to maintain high standards for stability and security. Such solutions can be found.

REPLACE OR REWRITE APPLICATIONS

"Rip and replace" means reinventing entire applications. Enterprises that have chosen this route – typically in an attempt to duplicate applications in Java – have usually found it to be an exercise in futility. In reality, an organization that is replacing its systems is not engaging in modernization but in an expensive, time-consuming, and highly disruptive effort to create an entirely new system. This is not an entirely sensible way to approach the challenges of minimizing disruption and cost—or to take advantage of the opportunity to preserve, leverage, and enhance the investment existing on the iSeries system and related assets.

RPG/400 EMULATION

"Lift and shift" emulation leaves the application exactly as the original, nonmodern application, except that now it exists in a proprietary environment. Again, this option is not entirely sensible. It is demonstrably an expensive lateral move, and once accomplished, this sort of a platform shift is no closer than before to exploitation of modernization technologies such as Web services.

RPG/400 SCREEN SCRAPE

Screen scraping (or "Web facing") really isn't so much a method of Web-enablement as it is a look-and-feel "upgrade." It has been very popular among users seeking to make applications and data available through a graphical interface or to convert host screens to a Web-like appearance, without altering the applications or processes. The resulting Web pages are limited in functionality and cannot be enhanced. If any changes to the application are made, all of the affected screens have to be re-scraped – again, without the true benefits of Web enablement. As a result, business applications cannot evolve, and even simple modifications become risky, cumbersome, and costly.

An expensive option that rarely returns its value, screen scraping fails to leverage or enhance the organization's iSeries assets. Programmers simply pass the legacy source code through a tool that creates the output. The data, which remains on the iSeries, is not modernized in any way, and the legacy source code and applications also remain. Only the interface is modern. Screen scraping is at best a stop-gap measure; eventually, true modernization will have to be achieved through some other means.

HOST-BASED AD HOC

In a host-based "ad hoc" system, both Java and .NET are avoided. The iSeries remains central, preserving existing data assets. However, these solutions are extremely proprietary, placing the enterprise and its IT evolution at the mercy of the vendor or vendors involved with the solution. In this sense the "application" pillar is compromised, as is the "people" pillar, since the programmers will be largely dependent on an external party to maintain and enhance the system. This solution may be attractive to a small shop with a very small programming staff that would be overwhelmed by a change of platform. However, the disadvantages of this route suggest that alternatives should be explored if modernization for Web services is an eventual goal.

MIGRATION TO JAVA/C

Language converters attempt to translate RPG source code to C# or Java. It may be claimed that an RPG programmer can recognize the "ghost" of the original code, but in fact the converted code is neither native RPG nor native Java or C#. This code presents challenges for both current and future developers. It may be impossible for RPG programmers – who understand the application – to maintain. Interestingly, C# and Java programmers typically find the converted code just as un-maintainable. This is because C# and Java are object-oriented languages, whereas RPG is procedural. The translation results in an abomination to C# and Java language constructs.

RPG PLATFORM TRANSFORMATION

From the foregoing brief descriptions of modernization options, it may seem that there is no good alternative. Every option presents disadvantages that, depending on the organization's specific goals and circumstances, may be prohibitive, and often a combination of alternatives, and even of platforms, is the best solution to meet the challenges and opportunities of the enterprise's specific case. In many cases, however, a more integrated and flexible solution can be found in new tools that have emerged out of the demand for iSeries modernization.

It's now possible to migrate an existing system to .NET with little reengineering, while preserving scalability and positioning the migrated application to implement Web services and SOA. The most promising solution may be an enhanced version of RPG that is both MSIL compliant and fully object oriented. By migrating to such a language, the organization can leverage the skills and knowledge of its engineering staff – allowing the RPG programmers, who possess intimate knowledge of the applications and processes, to develop new Windows and Web-based applications. In addition, because the new language is MSIL compliant, RPG programmers and MSIL programmers (VB, VS, C#, and so forth) can work together on the same projects. The language bridges platforms both on the technological and the human level.

For implementation of Web services and service-oriented architectures, the availability of an MSIL-compliant RPG language provides a vital link. It ensures that the RPG applications participate in .NET, which in turn supports all of the standards – WDSL, XML, and so on – that enable Web services and SOA. Through such a language, RPG applications can fully participate in Web services and be modernized to and included in an SOA, while retaining the skills and knowledge of existing programming resource (RPG as well as OO), and without rewriting or re-hosting the applications. In other words, the iSeries and each of the pillars that supports both the IT system and the organization – the RPG programming support staff, the critical iSeries business applications, the inherent business processes they support, and the data that resides on the iSeries – can be transformed into a modernized, Web-enabled system, thanks to tools that bridge the iSeries/RPG and .NET.

In addition, an SOA, once implemented through an MSIL-compliant language, provides an elegant path to modernization that no other technology or methodology has accomplished to date. By encapsulating business functions performed in the legacy environment inside Web services, a company can now leverage the investment in the business system while making it available to other heterogeneous environments. Furthermore, using the SOA model, a company can extend those parts of their legacy application that they need over a time frame that makes the most business sense to them.

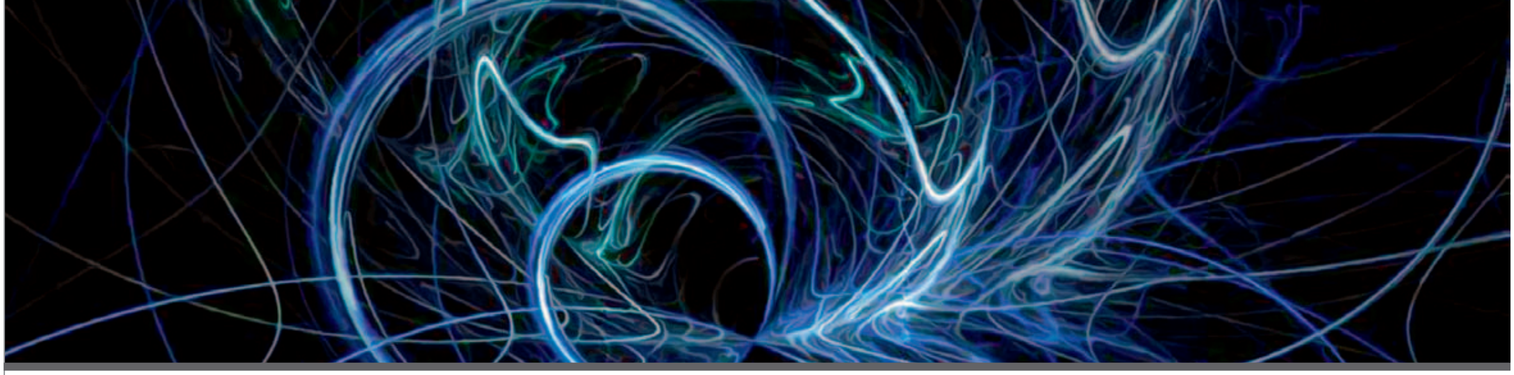
Checking the "MAP"

In 2004, Microsoft teamed with nine iSeries service providers and ISVs in an alliance – the "Midrange Alliance" – intended to encourage iSeries shops to consider a variety of alternatives when they modernize. The project was spearheaded by ASNA – a longtime iSeries vendor that markets extension, modernization, and migration software products, including ASNA Visual RPG, an MSIL-compliant, fully object-oriented RPG language, and its Monarch product, an iSeries/400 to .NET replatforming analysis and transformation solution. The Midrange Alliance, which now has 19 members, provides resources for organizations planning modernization efforts on their Web site, including case studies, white papers, and pertinent information on IBM, Microsoft, and third-party products. Although the alliance may have a reputation for offering migration-only options, members, which include ASNA and LANSA, another founding member, also provide solutions for organizations that want to retain sensible loyalty to the iSeries.

Summary and Evaluation

For the six modernization options listed earlier, we can now provide a concise summary: the first two, replace or rewrite applications and RPG/400 emulation, are not sensible approaches to modernization, if they could be considered modernization at all. Screen scraping creates the look of modernization without doing so in reality, and in fact results in programs that resist modernization. Host-based ad hoc systems bind the organization to proprietary, non-standard technologies and remove maintenance, enhancements, and capabilities from the organization to the vendor. Language conversion results in the loss of the original RPG developers (and their "tribal knowledge") and in code that is difficult to maintain for RPG and Java or C# developers alike.

Platform transformation tools, such as an MSIL-compliant RPG language, offer the greatest promise for meeting the challenges of modernization: bringing the enterprise system into the 21st century while avoiding unnecessary or costly disruption to the business. In short, it provides the best means of managing costs and achieving a reasonable timeline while making the most of opportunities that leverage an organization's assets.



From Mobile to Multimodal

Mobile computing gets vocal

BY LES WILSON

Les Wilson is an IBM Senior Technical Staff Member. He has been responsible for a variety of user interface research and development projects over many years in IBM R&D. Les is the architect for multimodal systems in IBM's Software Group.

lesw@us.ibm.com

From battery powered tablet computers to Internet connected smart phones, advances in networking and electronic miniaturization have created more ways to access Internet services and data than ever before. Old problems can now be solved with new mobile computing technologies. The ubiquity of access to data and services via Internet Web browsers has enabled large scale integration of business processes and workers.

Anytime and anywhere access to your business is a great marriage of Internet technology and IT but it creates a new problem that requires a new solution. Enter the world of mobile speech recognition.

Battery-powered handheld devices are mobile and miniature. Wireless networks untether workers from offices and data centers creating virtually anywhere access to data and services. But something happened when computing left the desktop: the keyboard and mouse stayed behind. Filling out forms and moving from one application to another on a mobile device becomes a tedious process of pointing and clicking with fingers on a small touch screen, or virtual keyboard.

So while mobile e-business can transform business processes and bring access to more places, smaller screens and compact devices create a user interface problem.

Using Web applications on mobile devices begs for easier ways of entering and accessing data. Smaller screens mean less space for drop down lists and links. Adding speech input and output as a natural interface to a Web application can solve many user problems brought on by the advances of mobile computing. It means hands and eyes can be busy doing other things.

Joining speech recognition with Web applications blends the best of both worlds and can allow individuals to get the most out of computing. This multimodal approach to computing is a perfect union of user interface technology that provides alternate ways of entering and accessing information, including speech, keyboard, handwriting recognition and visual data output.

In form-centric industries like healthcare and insurance, mobile devices can put data acquisition and reporting at the bedside, roadside or doorstep of a disaster relief mission. Multimodal interaction can mean faster access, fewer errors and less stress because data entry and retrieval is natural and accurate. Adding speech recognition to mobile applications can be done with the help of new technologies that allow mixing the Web and speech programming environments together.

Just Add Voice

XHTML plus Voice, or X+V for short, is a multimodal Web application programming language based on existing Web standards. X+V applications can be created with the IBM Multimodal Toolkit for WebSphere Studio. This toolkit adds plug-ins to the WebSphere Studio Site Developer or WebSphere Studio Application Developer toolsets so developers can easily turn existing WebSphere applications into multimodal ones. That is, the tools enable developers to "Just Add Voice."

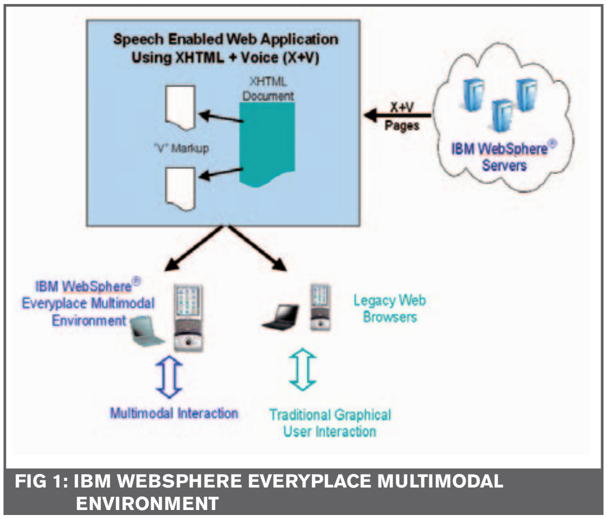
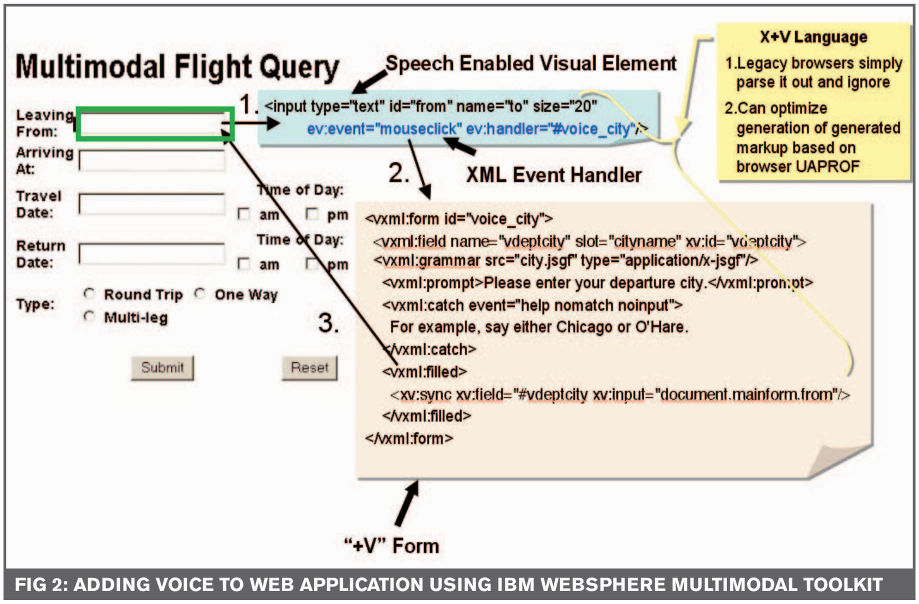

Speech Enabled Web Application
Using XHTML + Voice (X+V)
XHTML Document
"V" Markup
X+V Pages
IBM WebSphere® Servers
IBM WebSphere® Everyplace Multimodal Environment
Legacy Web Browsers
Multimodal Interaction
Traditional Graphical User Interaction

FIG 1: IBM WEBSPHERE EVERYPLACE MULTIMODAL ENVIRONMENT



```
<input type="text" id="from" name="to" size="20"
   ev:event="mouseclick" ev:handler="#voice_city"/>
```

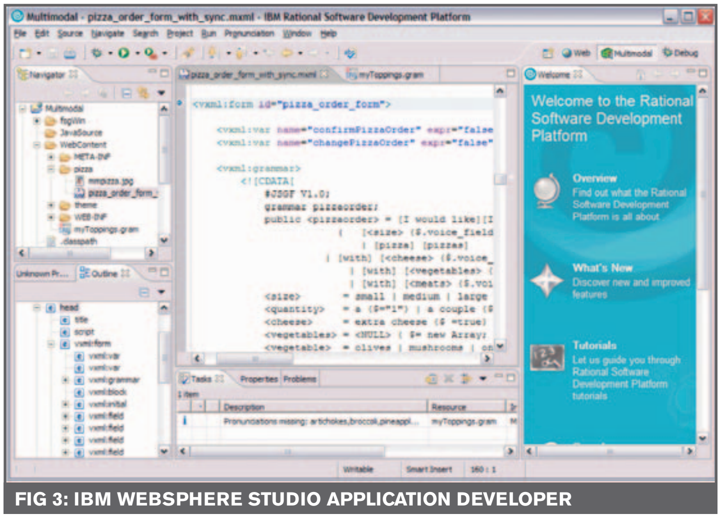
```
<vxml:form id="voice_city">
  <vxml:field name="vdeptcity" slot="cityname" xv:id="vdeptcity">
  <vxml:grammar src="city.jsgf" type="application/x-jsgf"/>
    <vxml:prompt>Please enter your departure city.</vxml:prompt>
    <vxml:catch event="help nomatch noinput">
     For example, say either Chicago or O'Hare.
    </vxml:catch>
    <vxml:filled>
     <xv:sync xv:field="#vdeptcity xv:input="document.mainform.from"/>
    </vxml:filled>
</vxml:form>
```

FIG 2: ADDING VOICE TO WEB APPLICATION USING IBM WEBSPHERE MULTIMODAL TOOLKIT

Because X+V uses the same tools and techniques that WebSphere programmers use today, the time it takes to add voice is much shorter than building a whole new application. With an XHTML application already working, you're well on your way to having a working multimodal application for your WebSphere deployment.

As with all Web technologies, the Web browser is a key component in delivering your application to users. IBM WebSphere Everyplace Multimodal Environment is a suite of Web browsers that support the X+V language and run on a variety of mobile devices, tablet computers, laptops and desktops.

With IBM WebSphere Everyplace Multimodal Environment, your multimodal applications can now deliver a better user experience. Users can be saved from the hassle of scrolling through long lists and filling in fields when their hands are busy or their devices too small. Input and output can be facilitated by voice when the situation or user preference warrants it, while navigation around and among applications can be done directly with voice instead of the usual clicking up and down menu hierarchies.



```
<vxml:form id="pizza_order_form">

    <vxml:var name="confirmPizzaOrder" expr="false
    <vxml:var name="changePizzaOrder" expr="false"

    <vxml:grammar>
        <![CDATA[
            #JSGF V1.0;
            grammar pizzaorder;
            public <pizzaorder> = [I would like][I
                              |    [<size> ($.voice_field
                                   | [pizza] [pizzas]
                            | [with] [<cheese> ($.voice_
                                 | [with] [<vegetables> (
                                 | [with] [<meats> ($.voi
            <size>        = small | medium | large
            <quantity>    = a ($="1") | a couple ($
            <cheese>      = extra cheese ($ =true)
            <vegetables> = <NULL> | $= new Array;
            <vegetable>  = olives | mushrooms | on
```

1 item

Description	Resource
Pronunciations missing: artichokes,broccoli,pineappl...	myToppings.gram

Welcome to the Rational Software Development Platform

Overview
Find out what the Rational Software Development Platform is all about

What's New
Discover new and improved features

Tutorials
Let us guide you through Rational Software Development Platform tutorials

FIG 3: IBM WEBSPHERE STUDIO APPLICATION DEVELOPER

Finally, with IBM WebSphere Everyplace Multimodal Environment, users are able to fill in more than one field at a time just by speaking and they'll be automatically asked for any information they may have missed.

Prior approaches to speech enabling applications tried to create new "one size fits all" language and technology. But like most things that try to be all things to all people, they don't do any one thing well. X+V blends the best of both worlds together in a way that makes sense for users, application developers and business. And, because it's built on Web standards, X+V applications can be fully compatible with Web browsers that don't yet have X+V support.

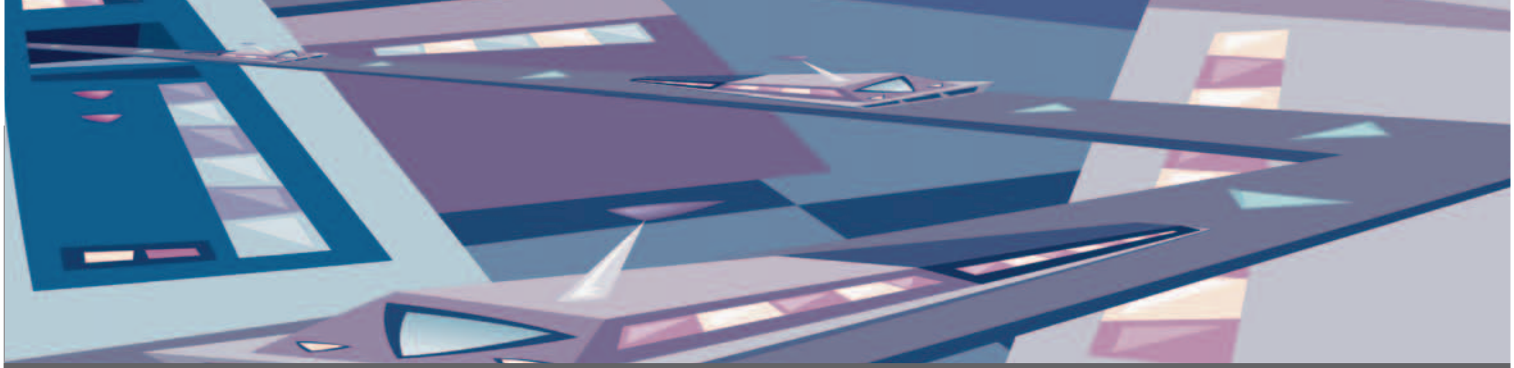
Step Outside the Box

Beyond making the usual data input easier in a mobile world, new ideas sometime create new ways of doing things. In hospitals, warehouses or the factory floor, hands-free access to data can mean hands free to care to for a patient, handle stock, or control equipment. Eyes-free access to information alerts and queries that can be read aloud means eyes can be available to examine a patient, find a part, or perform a repair.

The possibilities extend beyond handheld devices. Embedded software technology and wireless computing open up new possibilities for all sorts of devices to be connected to customized services and data.

For instance, sophisticated bedside care equipment operated locally but connected to the hospital back-end can enable real-time collaboration between medical experts a world apart. For healthcare, that means patient information is available to specialists who need it virtually anywhere and anytime. From smart phones, to PDAs, to tablets, wireless connectivity links the back end to the front desk - wherever that is. Accurate collection of patient information, inventory, order status, diagnosis, and treatment can mean better results faster. And that can mean better business.

Not everyone may be able to spell "pneumonia" or be able to read it on a handwritten patient chart. But if it can be spoken, mobile computing combined with multimodal interaction can create whole new information gathering, access and collaboration possibilities to help transform your business.



Inter-Portlet Communications

Using IBM Rational Application Developer V6.0

BY ASIM SADDAL

Asim Saddal works in the portal practice of IBM Software Services for WebSphere.
asaddal@us.ibm.com

This article demonstrates the steps performed to implement JSR 168 compliant cooperative portlets using IBM Rational Application Developer V6.0 and WebSphere Portal Server V5.1. The article illustrates passing multiple values from source portlet to target portlet without defining complex data type inside WSDL file.

The term *cooperative portlets* refers to the capability of portlets on a page to interact with each other by sharing information. One or more cooperative portlets on a portal page can automatically react to changes from a source portlet triggered by an action or event in the source portlet. Portlets that are targets of the event can react so that users are not required to make repetitive changes or actions in other portlets on the page. Cooperation between source and target portlets is facilitated by a WebSphere Portal runtime entity called the property broker. Portlets on a page can cooperate in this way even if they were developed independently, without the programmer's awareness of the existence of the other cooperative portlets.

JSR 168 is a specification from the Java Community Process for portlet development. IBM WebSphere Portal V5.1 provides support for the JSR 168 API. With an IBM extension, WebSphere Portal V5.1 supports cooperative portlets for JSR 168 portlets, in which one JSR 168 portlet can communicate with another JSR 168 portlet.

To develop and deploy the sample application, we used the following IBM products:

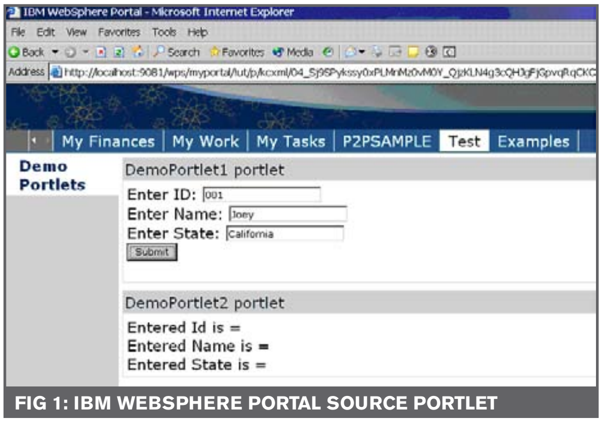
1. Rational Application Developer for Rational Software Development Platform V6.0
2. WebSphere Portal V5.1.x

Introducing the DemoPortlets scenario

In the DemoPortlets scenario we are going to create two JSR 168 portlets. The DemoPortlet1 will pass the multiple values to the DemoPortlet2 without defining complex data type inside WSDL file.

DEMOPORTLET1:

Gets the input from the user in three fields and then passed these input values to the target portlet. This portlet is our source portlet. (See Figure 1.)



IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
My Finances
My Work
My Tasks
P2PSAMPLE
Test
Examples
Demo Portlets

http://localhost:9081/wps/myportal/!ut/p/kcxml/04_Sj9SPykssy0xPLMnMz0vM0Y_QjzKLN4g3cQH3gFjGpvqRqCKC

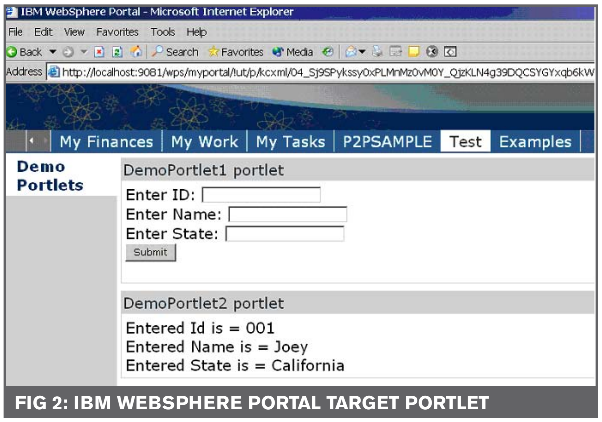
DemoPortlet1 portlet

Enter ID: 001
Enter Name: Joey
Enter State: California
Submit

DemoPortlet2 portlet

Entered Id is =
Entered Name is =
Entered State is =

FIG 1: IBM WEBSPHERE PORTAL SOURCE PORTLET



IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
My Finances
My Work
My Tasks
P2PSAMPLE
Test
Examples
Demo Portlets

http://localhost:9081/wps/myportal/!ut/p/kcxml/04_Sj9SPykssy0xPLMnMz0vM0Y_QjzKLN4g39DQCSYGYxqb6kW

DemoPortlet1 portlet

Enter ID:
Enter Name:
Enter State:
Submit

DemoPortlet2 portlet

Entered Id is = 001
Entered Name is = Joey
Entered State is = California

FIG 2: IBM WEBSPHERE PORTAL TARGET PORTLET

DEMOPORTLET2:

Act as a target portlet and retrieve those three entered values from DemoPortlet1 and display them on the page. (See Figure 2.)

Create Portlet Project (JsR 168)

Start the IBM Rational Application Developer (IRAD).

1. With the IRAD workbench started, switch to the Web perspective by clicking **Window > Open Perspective >Web**.
2. Click **New > Other**
3. Select **Portlet Project (JSR 168)** from the list. This launches the New Portlet Project (JSR 168) wizard (See Figures 3 and 4.)
4. Enter DemoPortlets as the **Name**.
5. Clear the **Create a portlet** checkbox. You will create your portlets separately in order to have better control over portlet naming conventions.
6. Click the **Show Advanced** button.
7. Select WebSphere Portal v5.1 Unit Test Environment in the **Target Server** list.
8. Accept defaults for the other fields.
9. Click **Finish**

Create DemoPortlet1 (JSR 168)

1. Select the **DemoPortlets** project in the Project Navigator view.
2. Right-click to bring up the context menu, and click **New > Portlet**. This launches the New Portlet wizard
3. Enter DemoPortlet1 as the **Default Name prefix**, click **Next** (See Figure 5.)
4. Accept default values and click **Finish**.

Create Portlet2 (JSR 168)

___As a same way create DemoPortlet2 portlet.

Enabling the DemoPortlet1 as a source

JSR 168 portlets can cooperate with each other by exchanging properties via the property broker. A WSDL file describes publish (or send) to the property broker.

DESCRIBING THE SOURCE WITH WSDL

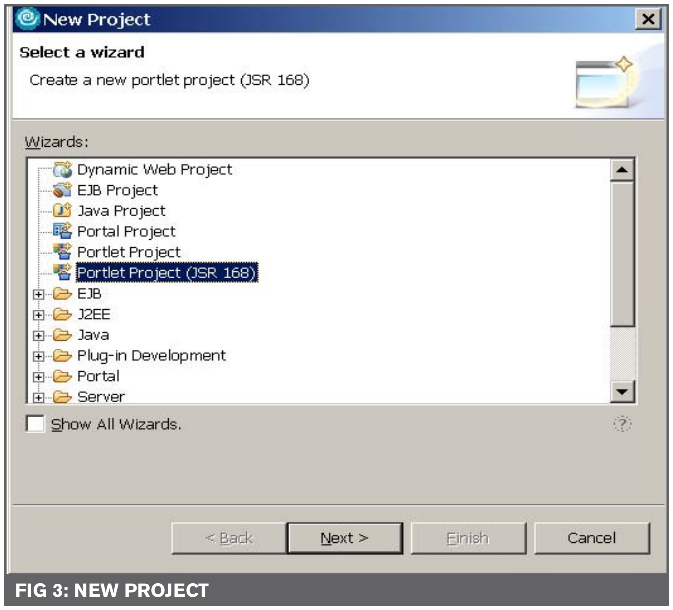
To enable our DemoPortlet1 portlet as a property source, simply right-click the portlet in the Project Explorer view to display the **Enable Source** (See Figure 6.)

Once the Enable Cooperative Source wizard launches enter the illustrated values: (See Figure 7.)

- A name for the **Data type**
- The **Namespace** for the new data type
- What you want your parameter **Bound to**

The term *parameter* refers to how the value will be transferred from the source portlet to the target portlet. The choices are:

1. ***None:*** This setting implies that you will not specify the way the value will be passed, so the default behavior for portlet


New Project

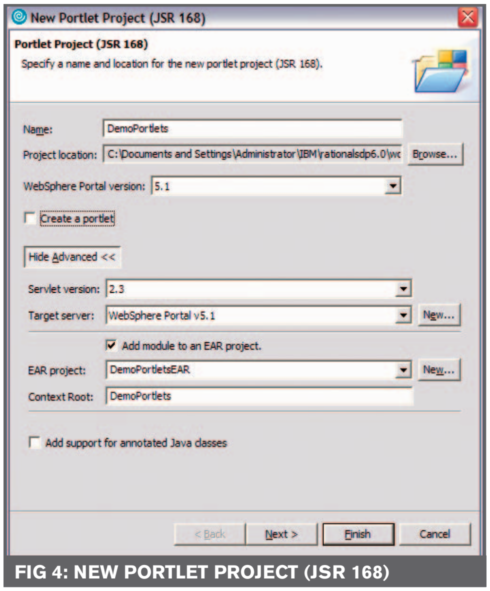
Select a wizard

Create a new portlet project (JSR 168)

Wizards:

FIG 3: NEW PROJECT

2. ***Render Parameter:*** Supports only strings. The string value will be bound to the RenderRequest object. The render phase and retrieved during the render phase of the portlet lifecycle. It *cannot* be retrieved during the action phase.
3. ***Request Parameter:*** Supports only strings. The string value will be bound to the ActionRequest object, and *can* stage of the portlet lifecycle. The parameter value will be meaningless at the conclusion of request processing (that invocation).


New Portlet Project (JSR 168)

Portlet Project (JSR 168)

Specify a name and location for the new portlet project (JSR 168).

Name: DemoPortlets

Project location: C:\Documents and Settings\Administrator\IBM\rationalsdp6.0\wc

WebSphere Portal version: 5.1

□ Create a portlet

Servlet version: 2.3

Target server: WebSphere Portal v5.1

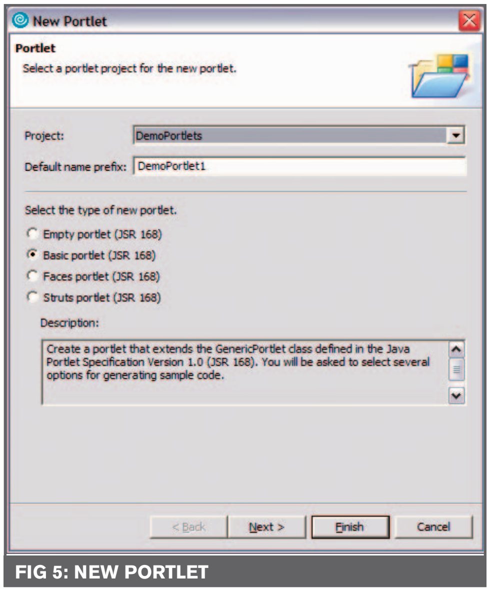
☑ Add module to an EAR project.

EAR project: DemoPortletsEAR

Context Root: DemoPortlets

□ Add support for annotated Java classes

FIG 4: NEW PORTLET PROJECT (JSR 168)


New Portlet

Portlet

Select a portlet project for the new portlet.

Project: DemoPortlets

Default name prefix: DemoPortlet1

Select the type of new portlet.

○ Empty portlet (JSR 168)
● Basic portlet (JSR 168)
○ Faces portlet (JSR 168)
○ Struts portlet (JSR 168)

Description:

Create a portlet that extends the GenericPortlet class defined in the Java Portlet Specification Version 1.0 (JSR 168). You will be asked to select several options for generating sample code.

FIG 5: NEW PORTLET

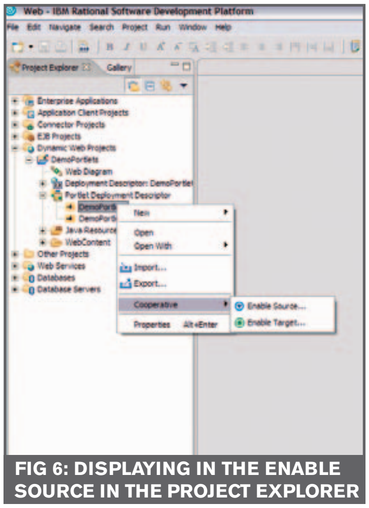
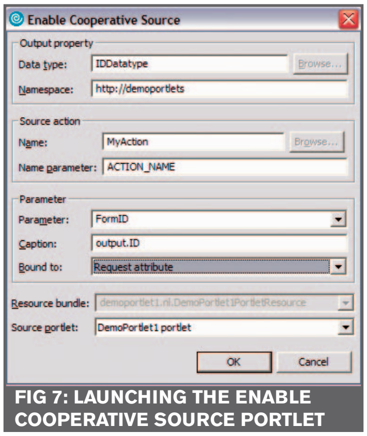


FIG 6: DISPLAYING IN THE ENABLE SOURCE IN THE PROJECT EXPLORER


Enable Cooperative Source

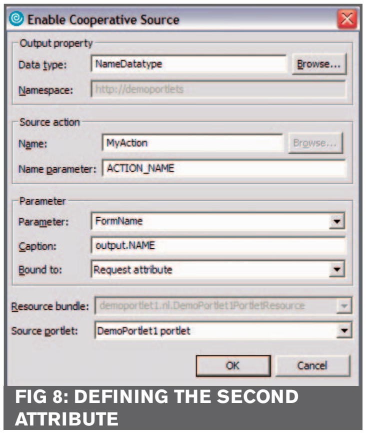
Output property
Data type: IDDatatype
Namespace: http://demoportlets

Source action
Name: MyAction
Name parameter: ACTION_NAME

Parameter
Parameter: FormID
Caption: output.ID
Bound to: Request attribute

Resource bundle: demoportlet1.nl.DemoPortlet1PortletResource
Source portlet: DemoPortlet1 portlet

FIG 7: LAUNCHING THE ENABLE COOPERATIVE SOURCE PORTLET


Enable Cooperative Source

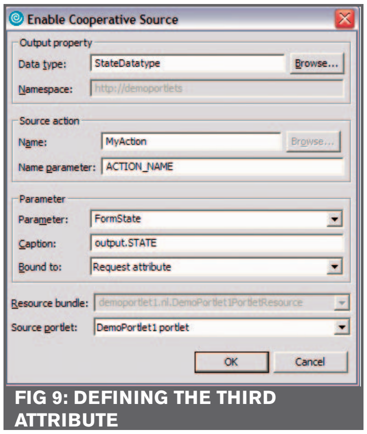
Output property
Data type: NameDatatype
Namespace: http://demoportlets

Source action
Name: MyAction
Name parameter: ACTION_NAME

Parameter
Parameter: FormName
Caption: output.NAME
Bound to: Request attribute

Resource bundle: demoportlet1.nl.DemoPortlet1PortletResource
Source portlet: DemoPortlet1 portlet

FIG 8: DEFINING THE SECOND ATTRIBUTE


Enable Cooperative Source

Output property
Data type: StateDatatype
Namespace: http://demoportlets

Source action
Name: MyAction
Name parameter: ACTION_NAME

Parameter
Parameter: FormState
Caption: output.STATE
Bound to: Request attribute

Resource bundle: demoportlet1.nl.DemoPortlet1PortletResource
Source portlet: DemoPortlet1 portlet

FIG 9: DEFINING THE THIRD ATTRIBUTE

4. ***Request Attribute:*** Supports any JavaBean type. The bean will be bound to the ActionRequest object. Lifecycle Request Parameter.
5. ***Session:*** Supports any JavaBean type. The bean will be bound to the session object and will persist for the duration and the portal server.

The Enable Cooperative Source wizard generates a WSDL file that describes the portlet to the property broker tagged with a distinctive icon to indicate that it is a property source. The WSDL file contains the following sections:

- ***Types:*** This section describes data types (using XML schema) that can be emitted by the source portlet.
- ***Messages:*** This section describes messages that can be produced or consumed by the portlet.
- ***Port Type:*** This section describes the abstract interface of the portlet as seen by the property broker.
- ***Binding:*** This section describes how the abstract interface (port type) is implemented.

To define the second attribute, right-click the portlet again and select **Cooperative->Enable Source** Define the following values: (See Figure 8.)

For the third attribute, perform the same step, as mentioned above, and provide following values: (See Figure 9.)

Source WSDL

Once we done with the wizard, the WSDL file should look like this:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<definitions name=”DemoPortlet1portlet_Service”
  targetNamespace=”http://demoportlets”
  xmlns=”http://schemas.xmlsoap.org/wsdl/”
  xmlns:portlet=”http://www.ibm.com/wps/c2a”
  xmlns:tns=”http://demoportlets”
  xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>
  <types>
        <xsd:schema targetNamespace=”http://demo-
portlets”>
                <xsd:simpleType name=”IDDatatype”>
                        <xsd:restriction
base=”xsd:string”></xsd:restriction>
                </xsd:simpleType>
                <xsd:simpleType
name=”NameDatatype”>
                        <xsd:restriction
base=”xsd:string”></xsd:restriction>
                </xsd:simpleType>
                <xsd:simpleType
name=”StateDatatype”>
                        <xsd:restriction
base=”xsd:string”></xsd:restriction>
                </xsd:simpleType>
        </xsd:schema>
  </types>
  <message name=”IDDatatype_Response”>
        <part name=”IDDatatype_Output” type=”tns:
IDDatatype” />
        <part name=”NameDatatype_Output”
type=”tns:NameDatatype” />
        <part name=”StateDatatype_Output”
type=”tns:StateDatatype” />
  </message>
  <portType name=”DemoPortlet1portlet_Service”>
        <operation name=”DemoPortlet1portlet”>
                <output message=”tns:IDDatatype_
Response” />
        </operation>
  </portType>
  <binding name=”DemoPortlet1portlet_Binding”
        type=”tns:DemoPortlet1portlet_Service”>
<portlet:binding />
<operation name=”DemoPortlet1portlet”>
<portlet:action name=”MyAction” actionNameParamet
```

LINUXWORLD®
CONFERENCE & EXPO

CONFERENCE:
AUGUST 8 – 11, 2005
EXPO:
AUGUST 9 – 11, 2005
Moscone Center West
San Francisco, CA

WHERE open minds MEET > >

explore > >   analyze > >   gain > >

LinuxWorld Conference & Expo is the world's leading and most comprehensive event focusing on Linux and Open Source solutions. At LinuxWorld, see and learn how to best leverage the technology for your organization.

> **Explore** your options on the exhibit hall floor, which features the world's leading hardware and software vendors.

> **Analyze** the latest Linux and Open Source technology and discover how companies across the globe can show you how to achieve higher profits and increase productivity.

> **Gain** knowledge about best practices and solutions by attending LinuxWorld's outstanding educational program.

It's the Linux & Open Source event you can't afford to miss!

linuxworldexpo.com

> Register Online With **Priority Code: D0109**

PLATINUM SPONSORS

Computer Associates® · hp invent · IBM · Novell · ORACLE · Sun microsystems

IDG WORLD EXPO

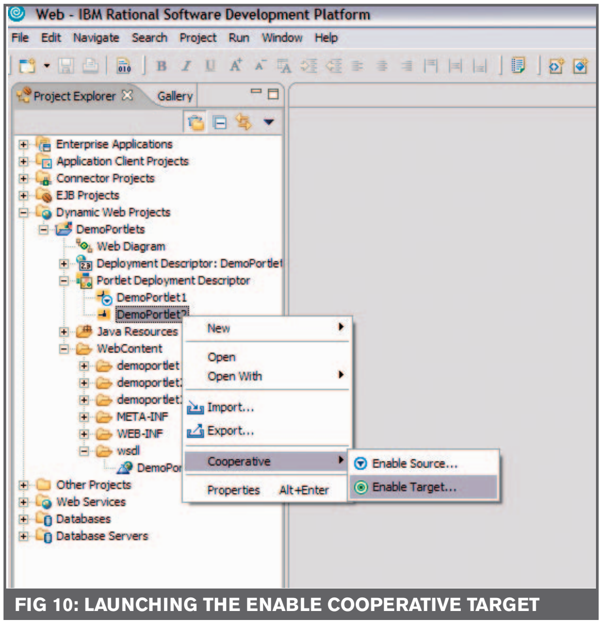


FIG 10: LAUNCHING THE ENABLE COOPERATIVE TARGET

```
er="ACTION_NAME" type="standard" caption="output.
data" description="Output.Data" />
<output>
<portlet:param name="FormID" partname="IDDatatype_
Output" boundTo="request-attribute"
caption="output.ID" />
<portlet:param name="FormName"
partname="NameDatatype_Output" boundTo="request-
attribute" caption="output.NAME" />
<portlet:param name="FormState"
partname="StateDatatype_Output" boundTo="request-
attribute" caption="output.STATE" />
</output>
</operation>
</binding>
</definitions>
```

Note: Make sure the captions and description are defined for <portlet:action> and <portlet:param> attributes. However, these fields are optional but very useful while creating wires.

Enabling DemoPortlet2 as Target

GENERATING THE WSDL

You will use the Enable Cooperative Target wizard to generate the WSDL. To launch the wizard, simply select the portlet and click **Cooperative > Enable Target** (this step is nearly identical to enabling cooperative source).(See Figure 10.)

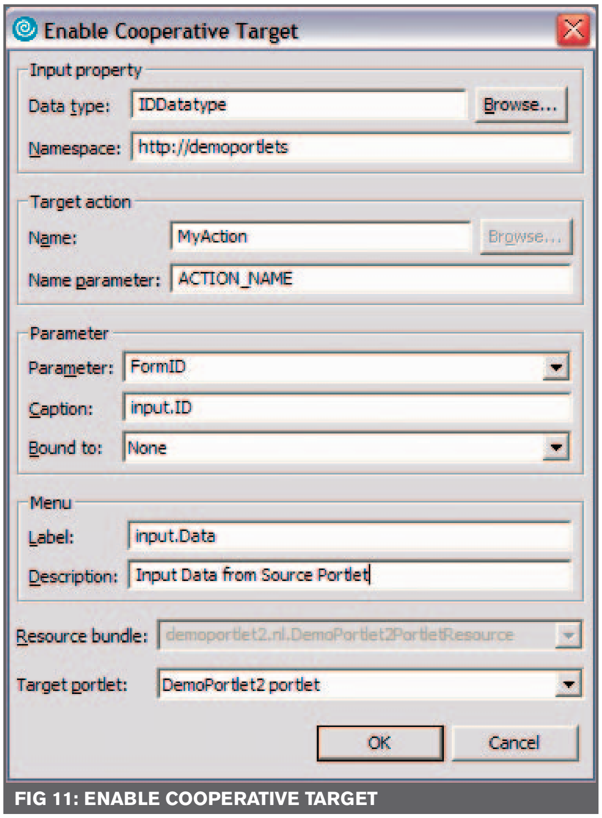
Once the Enable Cooperative Target wizard launches enter the illustrated values:(See Figure 11.)

- ***Data type:*** Use the exact same name that you used when you enabled the source portlet.
- ***Namespace:*** Again, the namespace that you enter here should be the same one you used for the source.
- ***Action:*** Use the exact same name that you used when you enabled the source portlet.
- ***Parameter:*** Use the exact same name that you used when you enabled the source portlet.
- ***Bound to:*** Choose **None** from the list.
- ***Label and Description:*** These fields are optional, but should be filled in; as they will help you create the wires

In order to define the remaining two attributes for the same target portlet, DO NOT use the wizard. We have to define them inside the WSDL file directly. Open the DemoPortlet2portlet.wsdl file and make the modifications according to the WSDL Source.

WSDL CODE

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="DemoPortlet2portlet_Service"
  targetNamespace="http://demoportlets"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:portlet="http://www.ibm.com/wps/c2a"
  xmlns:tns="http://demoportlets"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <types>
        <xsd:schema targetNamespace="http://demo-
portlets">
                <xsd:simpleType name="IDDatatype">
                        <xsd:restriction
base="xsd:string"></xsd:restriction>
                </xsd:simpleType>
        </xsd:schema>
        <xsd:schema targetNamespace="http://demo-
portlets">
                <xsd:simpleType
```


Enable Cooperative Target

Input property
Data type: IDDatatype
Namespace: http://demoportlets

Target action
Name: MyAction
Name parameter: ACTION_NAME

Parameter
Parameter: FormID
Caption: input.ID
Bound to: None

Menu
Label: input.Data
Description: Input Data from Source Portlet

Resource bundle: demoportlet2.nl.DemoPortlet2PortletResource
Target portlet: DemoPortlet2 portlet

FIG 11: ENABLE COOPERATIVE TARGET

```
name=”NameDatatype”>
                        <xsd:restriction
base=”xsd:string”></xsd:restriction>
                <xsd:simpleType>
        </xsd:schema>
        <xsd:schema targetNamespace=”http://
          demoportlets”>
                <xsd:simpleType
name=”StateDatatype”>
                        <xsd:restriction
base=”xsd:string”></xsd:restriction>
                </xsd:simpleType>
        </xsd:schema>
  </types>
  <message name=”IDDatatype_Request”>
        <part name=”IDDatatype_Input” type=”tns:
IDDatatype” />
        <part name=”NameDatatype_Input” type=”tns:
NameDatatype” />
        <part name=”StateDatatype_Input”
type=”tns:StateDatatype” />
  </message>
  <portType name=”DemoPortlet2portlet_Service”>
        <operation name=”DemoPortlet2portlet”>
                <input message=”tns:IDDatatype_
Request” />
        </operation>
  </portType>
<binding name=”DemoPortlet2portlet_Binding”
type=”tns:DemoPortlet2portlet_Service”>
<portlet:binding />
<operation name=”DemoPortlet2portlet”>
<portlet:action name=”MyAction” actionNameParameter
=”ACTION_NAME” type=”standard” caption=”input.Data”
description=”Input Data from Source Portlet” />
<input>
<portlet:param name=”FormID” partname=”IDDatatype_
Input” caption=”input.ID” />
<portlet:param name=”FormName”
partname=”NameDatatype_Input” caption=”input.NAME”
/>
<portlet:param name=”FormState”
partname=”StateDatatype_Input” caption=”input.
STATE” />
</input>
        </operation>
  </binding>
</definitions>
```

Note: Make sure the captions and description are defined for <portlet:action> and <portlet:param> attributes. However, these fields are optional but very useful while creating wires.The IRAD generates the DemoPortlet1portlet.wsdl and DemoPortlet2portlet.wsdl file. (See Figure 12.)

Java Class Modifications:

After defining the attributes and the portlet action in the WSDL file, the following code need to be written:

1. Open DemoPortlet1Portlet.java file.
2. Declare the following attributes:

```
  public static final String FORM_ID
= ”FormID”;
  public static final String FORM_NAME
= ”FormName”;
  public static final String FORM_STATE
= ”FormState”;
  public
static final
String ACTION_
TEXT

= ”MyAction”;
```

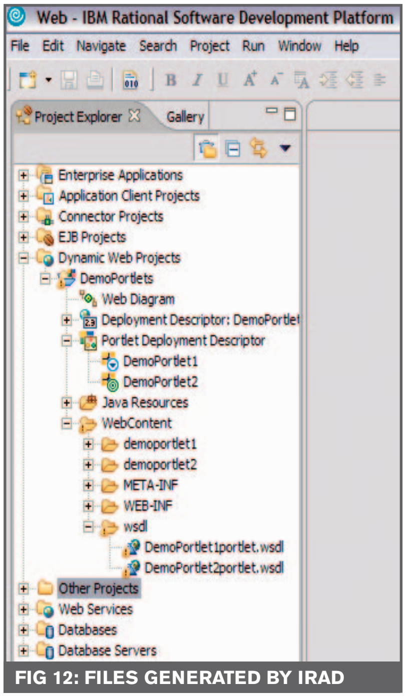


FIG 12: FILES GENERATED BY IRAD

Note: Make sure the values of FORM_ID, FORM_NAME and FORM_STATE must be same to the name defined for <portlet:param> in the WSDL file.

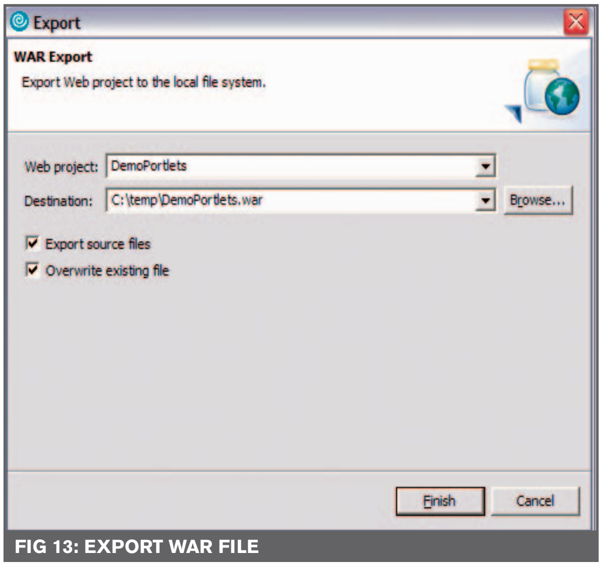
3. Modify processAction() method

```
public void
processAction(ActionRequest request, ActionResponse
response) throws PortletException, java.
io.IOException {

String id       = request.getParameter(FORM_ID);
  String name    = request.getParameter(FORM_NAME);
  String state   = request.getParameter(FORM_
STATE);

  request.setAttribute(FORM_ID, id);
  request.setAttribute(FORM_NAME, name);
  request.setAttribute(FORM_STATE, state);
}
```

3. Save the changes
4. Open DemoPortlet2PortletSessionBean.java and define following attributes

```
  private String id = ””;
  private String name = ””;
  private String state = ””;
```

5. Generate getter and setter methods for the above attributes and save the changes.



WAR Export

Export Web project to the local file system.

Web project: DemoPortlets

Destination: C:\temp\DemoPortlets.war

☑ Export source files

☑ Overwrite existing file

FIG 13: EXPORT WAR FILE

6. Open DemoPortlet2Portlet.java file.
7. Declare the following attributes:

```
  public static final String FORM_ID
= “FormID”;
  public static final String FORM_NAME
= “FormName”;
  public static final String FORM_STATE
= “FormState”;
  public static final String ACTION_TEXT        =
“MyAction”;
```

Note: Make sure the values of FORM_ID, FORM_NAME and FORM_STATE must be same to the name defined for <portlet:param> in the WSDL file.

8. Modify processAction() method

```
public void processAction(ActionRequest request,
ActionResponse response) throws PortletException,
java.io.IOException {
  DemoPortlet2PortletSe
ssionBean sessionBean =
getSessionBean(request);
  String id = request.
getParameter(FORM_ID);
  String name = request.
getParameter(FORM_NAME);
  String state = request.
getParameter(FORM_STATE);
sessionBean.setId(id);
  sessionBean.setName(name);
  sessionBean.setState(state);
}
```

9. Save the changes

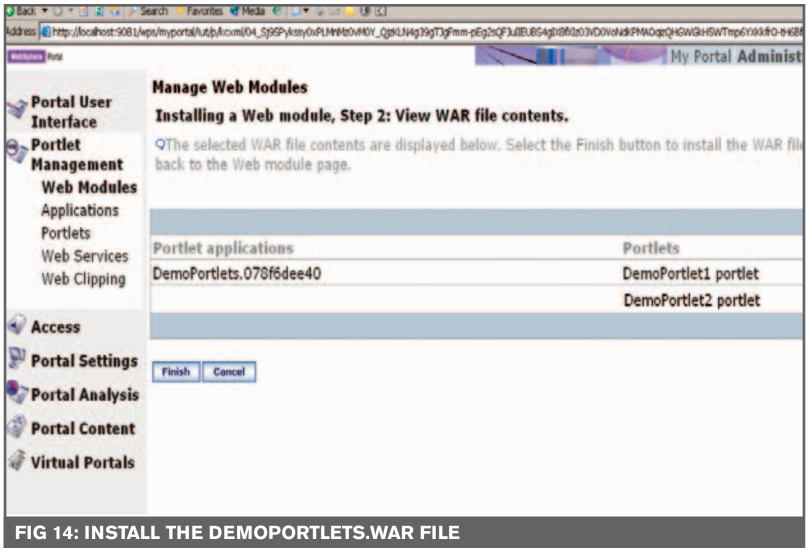
JSP Modifications:

1. Open DemoPortlet1PortletView.jsp
2. Modify the JSP code as follows:

```
<%@ page session=”false” contentType=”text/html”
import=”java.util.*,javax.portlet.*,demoportlet1.*”
%>
<%@taglib uri=”http://java.sun.com/portlet”
prefix=”portlet” %>
<portlet:defineObjects/>

<%
  PortletURL actionUrl = renderResponse.createAc-
tionURL();
  actionUrl.setParameter(”ACTION_NAME”,
DemoPortlet1Portlet.ACTION_TEXT);
%>

<FORM method=”POST” action=”<%= actionUrl.
toString() %>”>
Enter ID: <INPUT name=”<%=DemoPortlet1Portlet.FORM_
ID%>” type=”text”/> <br/>
Enter Name: <INPUT name=”<%=DemoPortlet1Portlet.
FORM_NAME%>” type=”text”/> <br/>
Enter State: <INPUT name=”<%=DemoPortlet1Portlet.
FORM_STATE%>” type=”text”/> <br/>
<INPUT name=”<%=DemoPortlet1Portlet.FORM_SUBMIT%>”
type=”submit” value=”Submit”/>
</FORM>
```

3. Save the changes
4. Open DemoPortlet2PortletView.jsp
5. Modify the JSP code as follows:

```
<%@ page session=”false” contentType=”text/html”
import=”java.util.*,javax.portlet.*,demoportlet2.*”
%>
<%@taglib uri=”http://java.sun.com/portlet”
prefix=”portlet” %>
<portlet:defineObjects/>

<%
```



http://localhost:9081/wps/myportal/!ut/p/kcxml/04_Sj9SPykssy0xPLMnMz0vM0Y_QjzKLN4g39gTJgFmm-pEg2sQF3UEUBS4gDx8fXdz03VD0VoNdkPMAOqzQHGWGkH5WTmp6YXkkfrO-tH6B

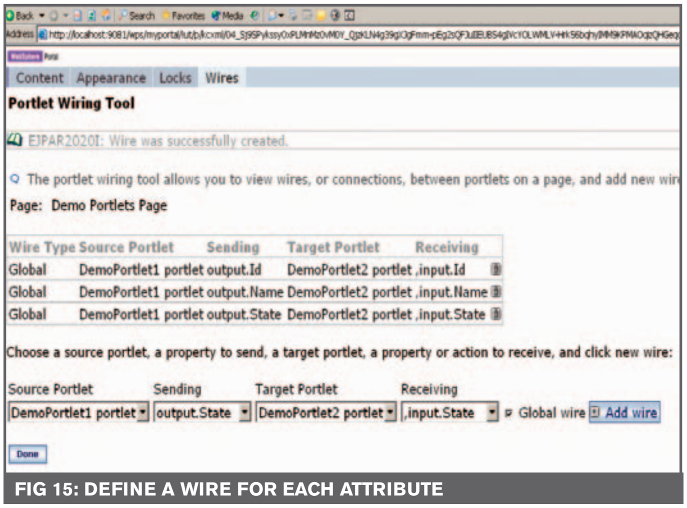
Manage Web Modules

Installing a Web module, Step 2: View WAR file contents.

The selected WAR file contents are displayed below. Select the Finish button to install the WAR fil back to the Web module page.

Portlet applications	Portlets
DemoPortlets.078f6dee40	DemoPortlet1 portlet
	DemoPortlet2 portlet

FIG 14: INSTALL THE DEMOPORTLETS.WAR FILE



Back Search Favorites Media

Address http://localhost:9081/wps/myportal/!ut/p/kcxml/04_Sj9SPykssy0xPLMnMz0vM0Y_QjzKLN4g39gJQgFmm-pEg2sQF3Jl2EUBS4glVcYOLWMLV-Hrk56bqhy9MM9kPMAOqzQHGeq

Content Appearance Locks Wires

Portlet Wiring Tool

EJPAR2020I: Wire was successfully created.

The portlet wiring tool allows you to view wires, or connections, between portlets on a page, and add new wir

Page: Demo Portlets Page

Wire Type	Source Portlet	Sending	Target Portlet	Receiving
Global	DemoPortlet1 portlet	output.Id	DemoPortlet2 portlet	,input.Id
Global	DemoPortlet1 portlet	output.Name	DemoPortlet2 portlet	,input.Name
Global	DemoPortlet1 portlet	output.State	DemoPortlet2 portlet	,input.State

Choose a source portlet, a property to send, a target portlet, a property or action to receive, and click new wire:

Source Portlet: DemoPortlet1 portlet
Sending: output.State
Target Portlet: DemoPortlet2 portlet
Receiving: ,input.State
☑ Global wire

FIG 15: DEFINE A WIRE FOR EACH ATTRIBUTE

```
DemoPortlet2PortletSessionBean sessionBean = (DemoPortl
et2PortletSessionBean)renderRequest.getPortletSession().
getAttribute(DemoPortlet2Portlet.SESSION_BEAN);
%>

Entered Id is = <%= sessionBean.getId()%> <br/>
Entered Name is = <%= sessionBean.getName()%> <br/>
Entered State is = <%= sessionBean.getState()%>
```

6. Save the changes

Deploy the DemoPortlets.war portlet

1. Export the DemoPortlets Project as a WAR file.(See Figure 13.)
2. Install the DemoPortlets.war file in WebSphere Portal Server (See Figure 14.)
3. Once the installation of the portlet is completed. Create the page
4. Place both the portlets on the same page
5. Create the wires between both portlets by selecting the Wire link on top of the page
6. Define a wire for each attribute defined and click **Add wire**, as shown in the figure (See Figure 15.)
7. Once all the wires created press **Done**.
9. Go to the page where both portlets are placed.
10. Test the portlets by entering dummy data and press Submit. You should see the values on the second portlet.

Conclusion

The WebSphere Portal Server Property Broker allows passing complex data between portlets without defining complex data type inside the WSDL files.

Resources

- WebSphere Portal Server V5.1 InfoCenter (http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/index.html)
- WSDL specifications (www.w3.org/TR/wsdl)
- WebSphere Portal Zone (www-128.ibm.com/developerworks/websphere/zones/portal/)
- IBM Redbooks: IBM WebSphere Portal V5 A Guide for Portlet Application Development [http://publib-b.boulder.ibm.com/abstracts/sg246076.html?Open]

FINAL **THOUGHTS**

Whose Globe/Sphere Is It, Anyway?

BY ROGER STRUKHOFF

A few months back I used to this column to cogitate about the nature of a sphere. I subsequently compared that type of thinking to the wrongheaded "the world is flat" new paradigm that is going around the world (so to speak) because of a new book with that offensive phrase as its title.

Then, this month's issue of *WebSphere Journal* takes a deliberately eclectic approach that confirms neither that the world is spherical or flat, but shows that it is definitely complex and often confusing.

Confusion lies, in my opinion, at the root of the current debates about the nature of the world and the nature of how all of us should view it and treat it. Encapsulated in worldwide debates over "globalization," the debates are often not debates at all, but rather, strident arguments that often start with street battles (and now, apparently, terrorism), and end with uncompleted agendas.

IBM has certainly become part of the globalization debate with the successful sale of its PC business to Lenovo, and effectively to the government of China. Lauded as a smart business move for both companies—it let IBM immediately realize cash gains that would have taken more than a decade to realize through profits and it established relatively obscure Lenovo as a world player—the deal nonetheless gave a queasy feeling to U.S. citizens (both on the left and the right) about an impending global takeover by the emerging Chinese nation.

The queasiness make simply be a kneejerk reaction that portends nothing serious. Many will remember when Japanese companies were buying up things such as New York's Rockefeller Center and California's Pebble Beach Golf Course in the 80s, prompting outcries from all corners of the U.S.

In those cases, it turned out that the Japanese were hardly able to physically move either of these properties to Japan, that they were simply investors the same as any other, and that they got taken badly in both deals. A decade of stagnation in Japan has alleviated any American fears that Nihongo would be the language of choice in the 21st century.

Now comes fear of China, certainly not tied directly to the IBM-Lenovo deal, but tied to the potential Unocal takeover that comes in the wake of it. But it comes with a simultaneous fear of the U.S. in many parts of the world, of a highly militaristic nation that seeks to promulgate iconic business brands such as Starbucks and Nike in every corner of the earth. This fear, of course, follows at least 30 years of international mocking of earlier icons such as McDonald's and our fabricated Hollywood heroes and mindset.

Whose sphere/globe is it anyway? Do "American" oil companies actually produce any significant portion of their oil in the U.S. anymore? They do not. Don't most large American companies (such as IBM) generate at least half of their revenue and profits outside of the U.S.? They do.

Don't companies who are outsourcing offshore provide employment of human beings, often at wages far above the prevailing wage? Doesn't a rising tide lift all boats, as most U.S. technologists and globalization promoters insist?

I think IBM was visionary, and more important, correct in integrating Lenovo into the global PC infrastructure as a full playing partner. Surely everybody knows that there are hugely significant design and manufacturing industries throughout the world who have contributed mightily to the IT revolution over the past 20 years. IBM's move was simply a logical move within this reality. Expect more of this sort of thing, whether coming from China, India, or maybe Russia…or maybe even the United States.

And more to the point, WebSphere application developers are an eclectic bunch, coming from all corners of the world, developing applications to be deployed to all corners of the world. We live in a global business village, and attempting to leave that village will just put you out somewhere in the vicinity of the middle of nowhere.

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

YOUR TESTING SUCKS.

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

We understand.

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

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

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

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

LISA is different.

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

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

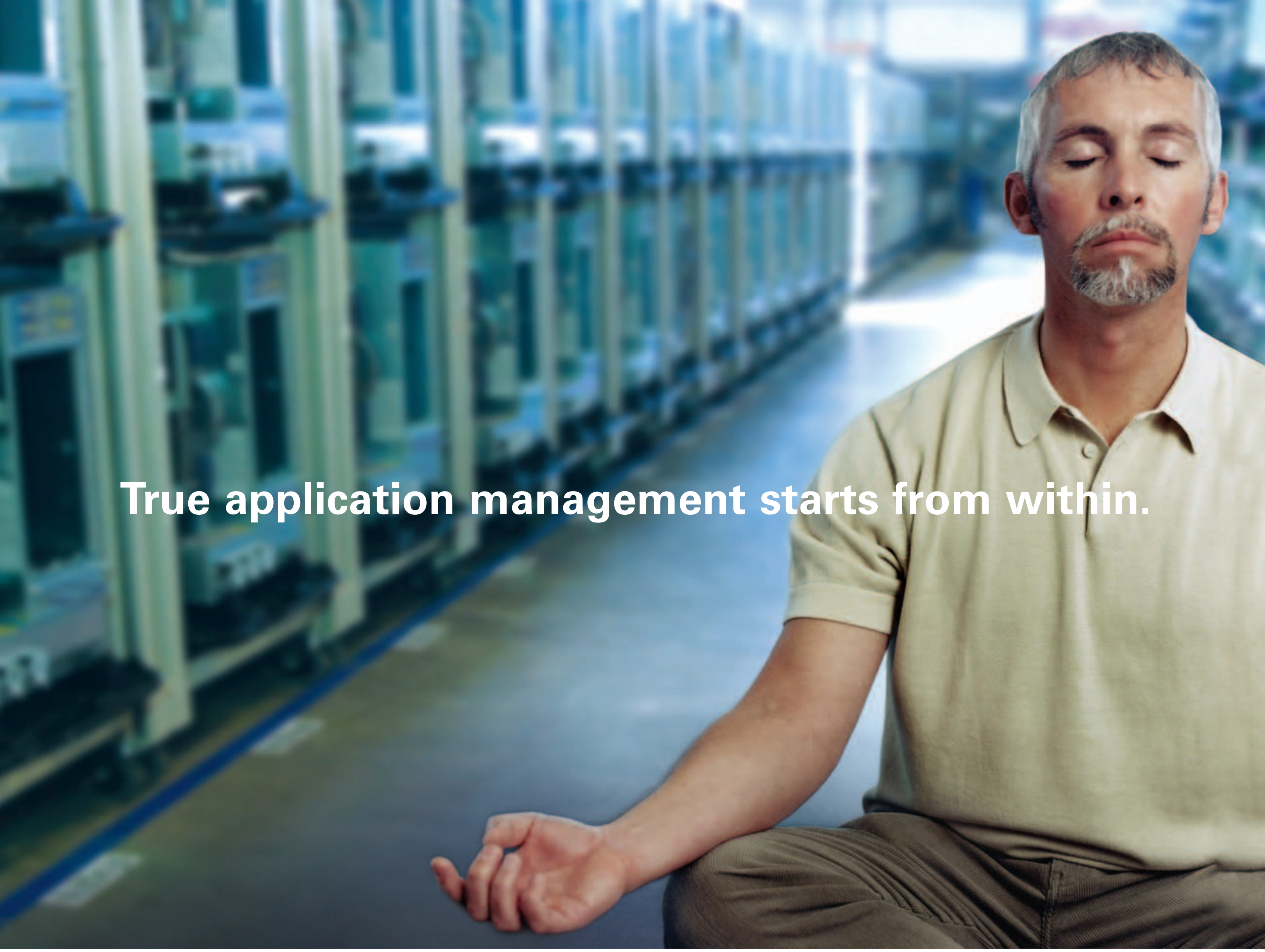
LISA automates test design, staging and monitoring for unit, regression, load and performance tests. And since LISA is pure Java, she's equally stylish on Linux, OS X, Unix or Win clients.

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

LISA from iTKO.

Because everyone should own quality.

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


True application management starts from within.