WEB SERVICES • REMOTE PORTLETS • REMOTE SERVER

www.SYS-CON.com/WebSphere

WebSphere® JOURNAL

The World's Leading Independent WebSphere Resource

DECEMBER 2004 VOLUME 3 ISSUE 12

RICHARD **GORNITSKY** & ROB **WILL** PAGE 6

DISPLAY UNTIL FEBRUARY 28, 2005
$8.99US $9.99CAN
12>
0 09281 03422 3

Tightening The Reins

BY JACK **MARTIN**

Another month is upon us and the world continues to change. The world of corporate governance is making its presence felt in all corners of information technology as the Sarbanes-Oxley Act requirements begin to take effect with regard to Section 404. You only need to read the daily newspaper or watch some television to get a feel for the level of corporate scandals that pervades our society.

The scandals run the gamut from whales like Enron and WorldCom to the growing trouble within the corporate and consumer insurance industries.

As I write, the attorney general of Ohio, Jim Petro has offered up an 87 page securities fraud class action lawsuit in federal court on behalf of the State of Ohio and all other shareholders in the Federal National Mortgage Association (Fannie Mae), alleging that the mortgage buyer manipulated its earnings to artificially inflate the price of its common stock.

"These defendants manipulated earnings in a fraudulent scheme to deceive investors about Fannie Mae's true financial state. This deception could cost shareholders billions of dollars," Petro said.

The complaint alleges that Fannie Mae and its top executives Franklin D. Raines, J. Timothy Howard, and Leanne G. Spencer artificially inflated the company's publicly traded common stock through false public financial statements. "Because these executives were compensated primarily on Fannie Mae's stock performance, this artificially high stock price allowed the executives to get rich at the expense of the company's shareholders," Petro said.

It seems Krispy Kreme is also stepping up to the plate with recent disclosures that the company has been financing their franchisees to keep them from going out of business.

Krispy Kreme management refused to take questions on its earnings call recently, instead demanding that questions be submitted by e-mail with the answers to be posted later on the company's Web site. Krispy Kreme CEO Scott Livengood cited the "legal environment we're navigating through" as the reason for not taking live questions. Yikes! Could there be a hole in Krispy Kreme's business story that matches the ones in their donuts? If you remember, Krispy Kreme was one of the few companies to go public during the dot.com era that was not a technology company made of vaporware. Stay tuned for the latest on the high calorie baker that was run by geniuses (until recently), which seems to have a few holes in its business model.

Several independent analysts predict that during 2005-2006, public firms will accelerate business and IT projects to ensure they are in compliance with SOX and a host of other regulatory edicts, consolidating in a corporate governance office. Firms will seek to optimize compliance processes through IT infrastructure (e.g., business applications, security), and many will also seek to improve business efficiency by using the compliance justification. Industry analyst firm Meta Group concludes a recent worldwide survey stating "by 2007, global compliance will raise control expectations for all multinational firms."

Sarbanes-Oxley compliance costs average $16 million per company and have caused companies like GE to spend a reported $30 million on internal control requirements alone. AIG chairman and CEO Maurice Greenberg recently indicated that the world's largest insurer was spending $300 million a year fulfilling the new requirements.

Deloitte & Touche has rolled out a sustained compliance solution framework as part of an integrated offering that is designed to help organizations move beyond the first year legal requirements of the Sarbanes-Oxley Act of 2002 and achieve sustained compliance.

According to Deloitte, most companies have focused principally on documenting, evaluating, testing, and remediating controls in preparation for the first year internal control reporting requirements mandated by Section 404 of Sarbanes-Oxley.

A recent survey found that 93 percent of corporate IT executives were unaware of their Sarbanes-Oxley compliance responsibilities. Interesting, considering that for the most part all of the money that has been spent has gone to information technology. I guess some people are planning to go to Club Fed next year.

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

WebSphere® JOURNAL

SUBSCRIPTIONS

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

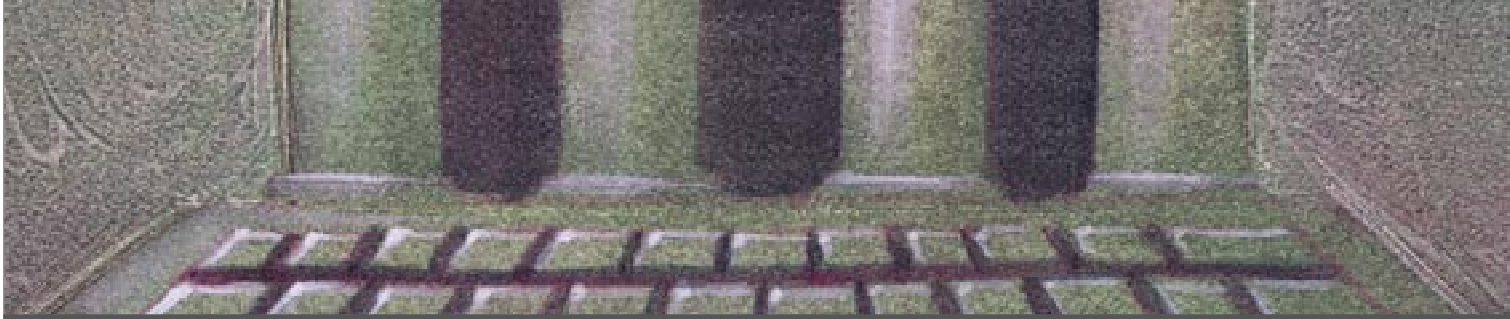
EDITORIAL OFFICES

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



PORTAL

What's New in WebSphere Portal v5.1?

Take enterprise integration to the next level

BY RICHARD GORNITSKY & ROB WILL

Richard Gornitsky is a WebSphere Portal architect with IBM's Software Services for Lotus. His expertise is in integrating WebSphere Portal into Fortune 500 firms from concept to production. Richard is a requested technical speaker and is a coauthor of Wiley Technology Publishing's Mastering WebSphere Portal. rsgornit@us.ibm.com

WebSphere Portal, the leading enterprise portal package on the market, is about to ship a major new release designed to take integration and flexibility deeper and further than ever before. WebSphere Portal lets programmers focus on developing the business functionality by managing the look and feel, personalization, content management, and security components.

WebSphere Portal has always been about integration. It lets users integrate different applications from disparate locations and enables them to seamlessly exchange information. Much of the functionality that is required for the Web experience does not have to be programmed but can be found in the portal library – a collection of "shrink wrap packaged" portlets or portal applications. Beyond just combining the portlets into a unified presentation, WebSphere Portal allows you to personalize the user experience based on the user's role and preferences as well as on the client device the user is accessing the site with.

Another key element of WebSphere Portal is that it allows you to build your site in a manageable way. The portal framework enables you to separate site design (what pages are where in the navigation hierarchy; navigation is along the top or the side; the banner goes across the top; etc.) from page design (what portlets go where on which pages) from application design (how do I connect to the back-end systems, and do I show the results in a table or a list?). By using the portal framework, your site designers, administrators, and programmers can all work independently without interfering with one another's jobs.

But if you thought WebSphere Portal was already powerful and fully functional, wait until you see Version 5.1! The new WebSphere Portal goes beyond integration at the glass to provide an infrastructure for integrating your content applications, as well as managing the user interaction front ending your service-oriented architecture's back end. We have also improved the speed and ease of deployment by providing a fast and reliable archive-based install, made it simpler to deploy additional portals with support for virtual portals, and made it even easier to create portlets with new tools for portlet development. The Web Content Management (formerly Lotus Workplace Web Content Management) has been redesigned to better fit into the portal. All the authoring and administration of Web Content Management (WCM) are now performed by portlets.

But that's not all. The search engine has been enhanced to allow searching within the portal, and navigation has been dramatically improved and now enables consistent use of the browser back button. The JSR 168 support has been enhanced to support additional IBM extensions. The Portal Document Manager and the Collaboration has also been functionally enriched. Now let's look into some of the exciting new features in WebSphere Portal v5.1.

Business Process Integration

More and more customers are adopting a service oriented architecture approach to their IT infrastructure. This means that they expose their infrastructure as a set of standards-based Web services connected together by an enterprise service bus (ESB) that enables applications to connect to these services as needed.

As that implementation matures, they are able to combine these services into business processes to meet their business needs. IBM is one of many companies behind the Business Process Execution Language (BPEL) standard, which is a standardized way to define and execute business processes composed of Web services. IBM's BPEL implementation is known as Business Process Choreographer (BPC). BPC ships as part of WebSphere Portal as well as a part of WebSphere Business Integrator (WBI).

With v5.1 of WebSphere Portal, we introduce an important integration with WebSphere Process Choreographer (WPC) and BPEL. First, there is the My Tasks portlet as shown in Figure 1. The My Tasks portlet allows users to claim and process WebSphere Process Choreographer tasks assigned to them. As each task is selected for processing, WebSphere Portal launches the corresponding "page application instance." By page application we mean a portal page that has been defined to contain the set of portlets necessary for the user to complete the specific task. This page is passed the context necessary for the portlets on the page to present the appropriate information to the user. For example, if the task were to approve a travel request, the task context might include the ID of the traveler and the destination. The portlets on the page could then display the traveler's past trips, manager information, lowest cost airfares to the selected destination, and other information necessary to approve or reject the travel request. Likewise, if the third item in the task list dealt with the creation of an RFI response, the corresponding page to instantiate could be filled with portlets that assisted in that task.

Since the user might have to work on multiple tasks at once, suspending work while waiting for more information for example, WebSphere Portal's My Task portlet instantiates an instance of the appropriate page for each task being processed. The page instances remain until the user completes the task or logs off (when the user logs on later, and clicks on the items in the My Tasks portlet, the pages can be instantiated again).

This is a significant enhancement to WebSphere Portal. Not only does it supply a way to provide a consistent user interface in front of an enterprise diverse application, but now we have extended Portal's navigation paradigm from just role-based navigation selection to enable the navigation to change based on the task at hand. This is important because companies have many processes and some are rarely used by any one employee. With just a role-based paradigm, the user would be exposed to all of these pages all the time. With task-oriented navigation, these extra pages are only visible when needed and go away when they aren't needed. This helps make the user more productive and improves the flow throughout your site.

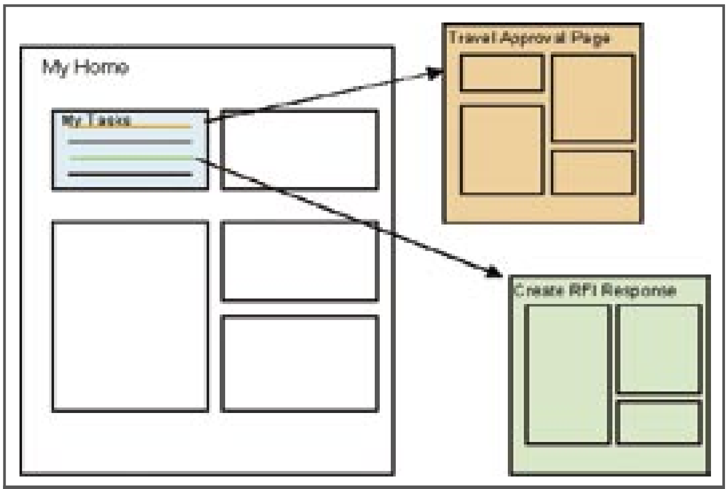


FIG 1: MY TASKS PORTLET

Virtual Portals

In Portal v5.0 and earlier, each unique portal requires its own portal installation: a WAS server, a portal server install, and a set of installed portlets, etc. As customers build out a portal implementation, they see how it can be leveraged in other areas of their business. In some cases, it just means adding more user communities to their portal rollout.

However there are cases where customers want and need distinct portals. For example, one of our customers is an insurance company that has deployed WebSphere Portal for its employees. It has been so successful that they have been asked to roll out a similar solution for the insurance agents. While there are some similar capabilities they want to offer the agents, there are also differences in terms of the anonymous pages and the overall site navigation and style. And of course, it is two distinct user communities (LDAPs). So this really is a second portal, but they don't want to have to maintain a second portal infrastructure.

With Portal v5.1 we introduce virtual portals. Virtual portals are entities that act as distinct WebSphere Portal instances, but run on a shared infrastructure. They are created on the administration page as seen in Figure 2.

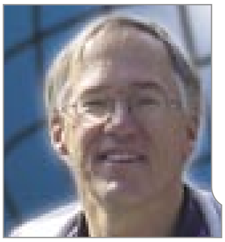
By distinct we mean that virtual portals have their own URI, places, pages, and set of users, as well as their own set of anonymous pages, log in pages, and sign-up pages.

Each virtual portal can have its own LDAP server or be assigned a subtree in a shared LDAP server. So from the users' perspective, they can't tell that these virtual portals are running on the same infrastructure.

The virtual portal administrators can more or less manage their virtual portals independently. There are some things that are shared between the virtual portals. For example, portlet applications only need to be installed once and then all virtual portals can use them. The portlet data can be specific to each virtual portal. Also, themes and skins are common to all virtual portals, although of course each virtual portal can be assigned its own unique set of themes and skins.

Portal Document Management Enhancements

Portal Document Management (PDM) has been enhanced in v5.1. As shown in Figure 3, we have created a new user interface including the ability to show an explorer-like folder hierarchy as well as introducing live names. Also, we've streamlined the "round trip" editing process. By round trip editing we mean launching a client-side editor (such as Microsoft Word), editing a file locally, and then storing the result back in PDM. With v5.1 this requires fewer clicks and is more seamless. Finally, we have re-based PDM and WCM onto the future DB2 Content Manager runtime based on the current JSR 170 draft specification.



Rob Will is a distinguished engineer in IBM's Software Group and the chief architect of WebSphere Portal. He has been a member of the WebSphere product development team since the beginning, working on both the WebSphere Application Server and WebSphere Studio focusing on Web applications. His current responsibilities include architecture for WebSphere Portal, Web Content Management and Document Management.
willrc@us.ibm.com

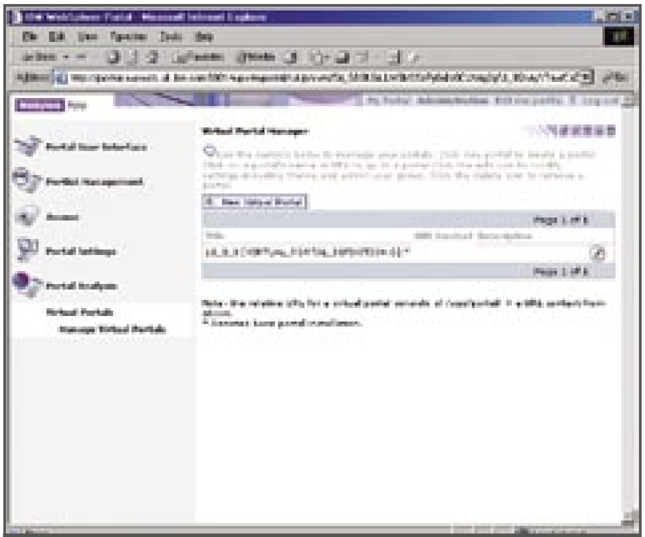


FIG 2: VIRTUAL PORTAL ADMINISTRATION PORTLET

Web Content Management v5.1

Web Content Management, formerly known as Lotus Workplace Web Content Manager (LWWCM), has a number of recent enhancements. It includes versioning support, a documented set of APIs for importing and exporting content, the ability to reference external content from within IBM content management repositories within LWWCM templates, and a number of performance enhancements.

WCM is much better integrated with Portal v5.1. It uses a portlet-based user interface and stores its users and groups in WebSphere Member Manager. Figure 4 shows the new authoring portlet.

WCM can now reference PDM documents within WCM presentation templates. In an upcoming fixpack, WCM will support personalization rules within WCM templates.

Portal Search Enhancements

The Portal Search has been enhanced to make it easier to crawl and search the portal itself. When indexing the portal, you select a specific user ID to crawl the portal as the index is developed using that level of security. When a user subsequently searches the portal, the results are filtered based on the searching user's authorities and the results shown in the context of the portal page on which they reside. A new search bar is now provided within the default themes so that the search bar can appear on every page of the portal without having to add the search portlet to each page.

Another major search enhancement is the introduction of the search center. The search center allows the administrator to define a set of search targets within a single portlet. Using this portlet the user can go to one place and search multiple repositories supported by the portal search engine. DB2 Information Integrator Omni Find supports customers who need broader functionality. We have added new portlets for managing the indexing of the portal server and managing the search taxonomy.

The portal search engine can now be remotely invoked. This means that the indexing and search can be run on a separate server. This takes load off of the portal server and allows a cluster of portal servers to share a single search index for more consistent search results and less overhead.

Installation and Configuration Enhancements

In WebSphere Portal v5.0, major enhancements were made to the install making it simpler and more reliable. The new procedures enabled a working portal infrastructure to be quickly implemented with a cloudscape configuration. Customers could then configure their choice of databases and LDAP server when they were ready. While this was a major improvement over earlier releases, there were still three areas in which we could improve. First, it took too long to do the install; second, there was no simple interface for creating the Ant scripts needed to configure the end results; and third, the transfer of data from the base install Cloudscape database to the selected database and LDAP was not as efficient as it could have been.

Portal v5.1 addresses all of these problems. For initial install, most customers will see substantial performance improvements using our archive-based install. Basically, archive install copies a working image to the location of your choice and then does the necessary customizations such as host name, admin passwords, etc.

We've also added a wizard to help customers configure their LDAP and databases instead of having to edit the Ant scripts directly.

Finally, we've improved the DB transfer performance by creating a direct connection between the Cloudscape database and the selected database, removing the intermediate copy to disk.

Operations and Deployment Enhancements

A key focus area for Portal v5.1 is to simplify operations and deployment. With Portal v5.0, WebSphere Portal did a good job moving full portal configurations from staging to production, but did not provide automated support for staging of incremental and differential releases of portals. That is, after getting to a running portal on staging and production, you need to be able to make and test changes on the staging server. When these changes were modifications or additive, everything was fine, but when it included deletions, the deletions had to be done manually on the production server. Portal v5.1 includes the Release Builder to aid in this process. The Release Builder can take the existing portal configuration and compare it to the new portal configuration from the staging server and produce a differential file, and then execute that differential file to bring the production server to the same level as the staging server. We have also improved our operations support for clustering. Specifically, we now allow WebSphere Portal to be installed into an existing WAS v5 cell and allow configuration changes to WebSphere Portal without first taking the node out of the cell.

Additional operational and deployment enhancements include:

- Scripting support has also been provided for key administration functions
- Ability to enable XMLAccess import and export from within portlets instead of just via a command line interface
- Support for moving pages and page hierarchies within the portal hierarchy
- The ability to administer portlet and page caching

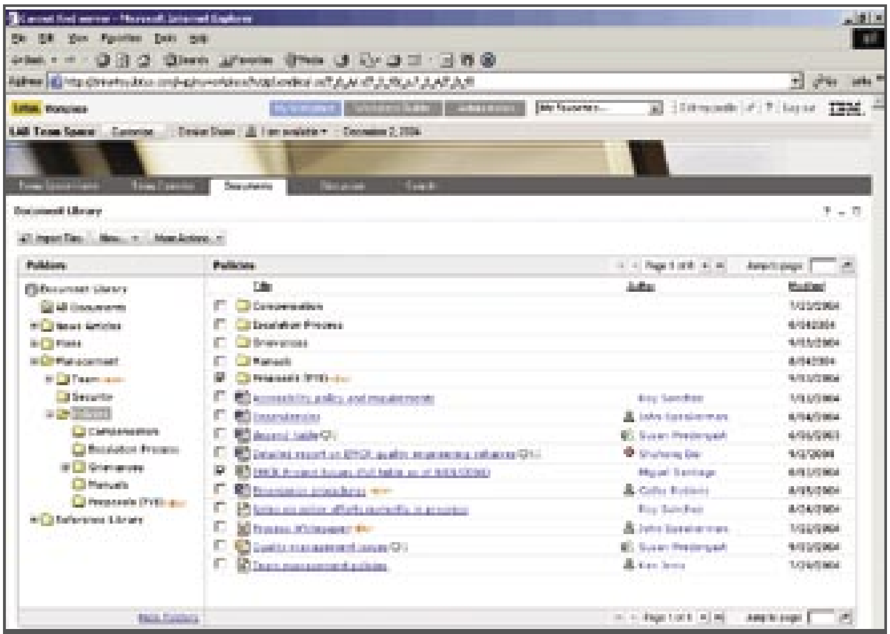


FIG 3: PORTAL DOCUMENT MANAGEMENT

Navigation Enhancements

The biggest change in navigation is in providing consistent support for the navigation state within the portal. Navigation state deals with where you are in the portal versus application state. By formally defining navigation state and how to deal with it in your portlets, we enable the portal to provide overall consistent back button behavior.

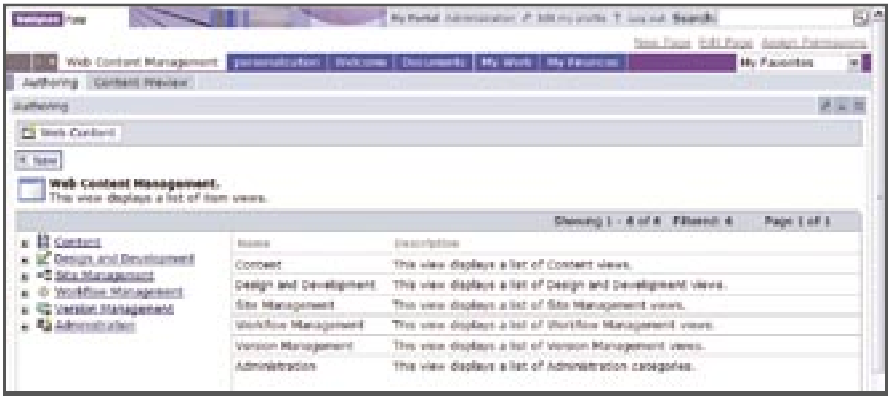


FIG 4: WCM AUTHORING PORTLET

Development Enhancements

The programming model has been enhanced by enabling JSR portlets access to WebSphere Portal enhanced features. JSR 168 portlets have access to key services such as portlet cooperation (excluding click2Action). WebSphere Portal's JSR portlets have the support of Struts v1.1, remote caching in edge servers, the ability for portlets to set the cache scope, and support of sessions for anonymous users.

The number of supported public APIs has been increased and has been clearly tagged as either public or non-public.

Screens are being phased out as we enable log in, enrollment, and self-care within portlets.

Performance Enhancements

Significant performance enhancements have been made including:

- Remote cacheable portal pages
- Shared caching of topologies
- Cache group membership in WMM
- High performance themes and skins
- Increased admin performance
- Asynchronous model cleanup

Security Enhancements

WebSphere Portal v5.1 offers a number of security enhancements and it now supports Java 2 Security. J2SE security can be used to control "which code is allowed to execute which other code" (also called code source security). With Java 2 Security enabled on the WebSphere Portal Server instance, the administrator can control code privileges of foreign code (e.g., portlets) by setting up corresponding J2SE policy files. This allows the administrator to make sure that semi-trusted portlets installed on the system only have limited access to system resources (e.g., file or network access).

The User and Groups permission portlet has been improved by providing a visual extension. The User and Group permission portlet now displays all role assignments for the selected user or user group and all child resources of a selected parent resource (e.g., all subpages of a given page) in one view making it much easier to understand all the capabilities of a given user.

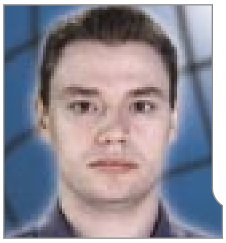
Summary

This article just touched on some of the great enhancements available in WebSphere Portal v5.1. Complete information on WebSphere Portal v5.1 can be found at www-306.ibm.com/software/genservers/portal/. With Version 5.1, IBM continues to maintain its leadership position in the portal marketplace and allows you to take enterprise integration to the next level.

Web services and SOA revisited

From Blueprint to Skyscraper

BY JOSEPH R. MARQUES



Joseph R. Marques is a member of Prolifics' WebSphere Consulting Division – a highly specialized team of experts that IBM calls on to deliver training, mentoring, and development services, and to solve the toughest of their customers' challenges. Specializing and certified in Java, WebSphere architecture and best practices; application development and deployment; and portal development, Joseph is responsible for delivering distributed, J2EE WebSphere solutions to clients worldwide.
jmarques@prolifics.com

As many people correctly predicted, Web services and their corresponding service-oriented architecture (SOA) have proven their promise. Companies and businesses are starting to not only integrate new components and technologies, but are also opening up back-end systems and legacy processes. The former is helping to end compatibility and integration issues of the past while the latter is working to foster growth and more forward-thinking ideas, thus exempting the need to waste time on costly rewrites of existing technology.

Web services do not solely add benefits to individual enterprises; they also help to speed partnerships through quicker integration across businesses. This is a direct result of the abstract way a Web service describes itself, as it is almost completely agnostic of both medium and platform. So it is no longer a requirement for companies that wish to employ business to business (B2B) services to agree on a common language, infrastructure, or tool anymore. In this way, it is clear that Web services have been a primary player in a large paradigm shift, moving the emphasis from a technology-centric world to one that can focus purely on the business problem at hand.

The success of Web services and SOA means that companies want to build more complex systems, and need to construct blueprints detailing how the different systems link together.

A Map, Not a Route

Web services and SOA, as with any sophisticated architecture, add benefits as well as new complexities. You must realize that architectures are analogous to procedures. They are methods and processes for doing things. They describe how elements are to be bound together. Therefore, these technologies in and of themselves are not going to solve the problems of business integration overnight. They provide the pathways or the map, but not the specific direction in which to head nor the specific route to take.

Thinking About The Big Picture

Bear in mind, new technology *is* great, but it must be used with caution, care, a thorough understanding of its pros and cons, and knowledge of how it fits into "the big picture" – anything less is just asking for trouble. Businesses must understand that in order to produce a well-oiled overall product, each of its constituent parts must work well *before* the oil is applied. If each were constructed with care and detail, they will all come together nicely and operate cleanly and consistently with one another. However, if you lapse on even a single item, the smallest component, or a solitary procedure, it could spell trouble down the line when that piece is incorporated into a larger plan.

Furthermore, no matter how well architected a system is, if the parts that produce the whole have been neglected, the deficiencies are guaranteed to show up in the final product. A Web service doesn't hide implementation issues; it magnifies them and opens up the problem for all the world to see. In other words, no tool today is going to make something happen magically, or turn a poorly architected system into a wonderfully coherent and understandable one. To reiterate, Web services provide the pathways, but not the specific direction.

Making the Blueprint a Reality

Designing the blueprint for a complex, integrated architecture is not a simple task. However, there is a tool today that can significantly simplify the process. It's called WebSphere Business Integration (WebSphere BI).

WebSphere BI is actually a suite of products that can model and analyze scenarios at the business process layer, not solely at the application layer as the seasoned developer is used to. Through something known as the business process engine, applications are loosely coupled with one another in tremendously flexible ways. Now, instead of statically coding the order in which applications or processes are called, instead of writing custom wrappers or communications connectors, instead of arduously and awkwardly writing exception-handling routines at varying levels of granularity, WebSphere BI brings everything together. It is an easily understood and quickly adaptable environment that provides the business analysts and developers with a tool of immeasurable value.

Think of all of the factors that go into writing even small applications, which may or may not be part of a larger architecture. You need to worry about design issues, reusability and maintainability, exception handling and user-input validation, support for transactions, authorization and authentication, encryption, chokepoints that require optimizations, hidden and/or missed business rules, changing business rules, etc. The list goes on and on. The point here is that with such a diverse and complex set of issues that must be addressed, there is a high probability that mistakes will be made along the way. Moreover, because so many companies are finding ways to reuse each other's services to cut costs or to share data because of mergers or partnerships, there has been an increasingly pressing necessity to find a solution that quickly adapts to these highly dynamic business models. It cannot be stressed enough just how important understanding the big picture is.

Look No Further

Thankfully, the solution to process management comes in the form of a standard – BPEL4WS (Business Process Execution Language for Web Services – BPEL for short). A mouthful to say, it introduces an array of features that can help control the seeming chaos at the business level. BPEL does not introduce anything particularly new or unthought of. It does, however, introduce a new way of approaching old problems, with a reusable framework that makes describing intricate dependencies, complex error handling, and difficult synchronization a snap. BPEL still supports capabilities from the application layer – all different sorts of control flow mechanisms, timer activities, a publish-subscribe architecture, and fault handling – but what it adds on top of that layer is the important part. Below I will highlight what I think are the most important concepts of BPEL and IBM's extensions to the language the have simplified the choreography of traditional back-end processes.

PARALLELISM OF TASKS

First and foremost is parallelism of tasks. Even in a highly object-oriented environment, this is much more easily said than done. In an ideal situation, all tasks to be executed would be asynchronous. That is, each process would be called and immediately return control to the main processing thread. Then, after all tasks of interests are started, a separate module could sit and listen or poll for responses from each of the individual tasks that were fired (see Listing 1).

However, what happens if one or more of the processes were synchronous? If a call into a method didn't immediately return, and was blocked until its completion? Essentially, it then requires the use of separate threads. After encasing each synchronous method in a lightweight thread wrapper (see Listing 2), they could be started and forgotten about just as in the asynchronous scenario (see Listing 3).

Any way you look at it, there is still going to be a level of indirection. A process will have to complete and then either explicitly notify the original caller of its results, or perhaps somehow write the result to a well-known location that the caller can later poll. This may not sound too bad for the case of a simple business process where the control flow is relatively uncomplicated. However, what happens in situations where a parallel series of tasks is nested

"it is clear that Web services have been a primary player in a large paradigm shift, moving the emphasis from a technology-centric world to one that can focus purely on the business problem at hand"

"A Web service doesn't hide implementation issues; it magnifies them and opens up the problem for all the world to see"

within a sequential series of tasks, which is in turn just a subtask of another series of parallel tasks? For a very complex process this could be a veritable nightmare to program by hand at the application level.

There are two tools from the WebSphere BI suite that make dealing with parallel and complex nesting of tasks simple. A business analyst is going to tend to want to use the WebSphere BI Modeler tool while a developer would lean towards Process Choreographer, which is built right into WebSphere Studio Application Developer Integration Edition. Modeler is a high-level tool and does not attempt to map a process flow to any implementation, whereas Process Choreographer does. Aside from that, each tool provides a similar graphical front end, which makes working with parallel tasks a simple matter of visualizing the task and dragging and dropping components onto a palette.

Figure 1 shows a WebSphere BI Process Choreographer rendering of a complex business model including both parallel and sequential processing and nested tasks, some of which require manual intervention. It was written using Process Choreographer in less than three minutes! With just a few clicks and drags, a process can be structured with highly complex control flow, composed of both synchronous and asynchronous processes. Horizontal blue lines with a shaded gray background define the parallel processing parts, whereas those grouped within a solid white background define tasks to be executed in series.

When looking at it, one can clearly see that a great deal of the power and flexibility comes from the fact that links can be drawn explicitly from the output parameters of one service to the input parameters of another. In other

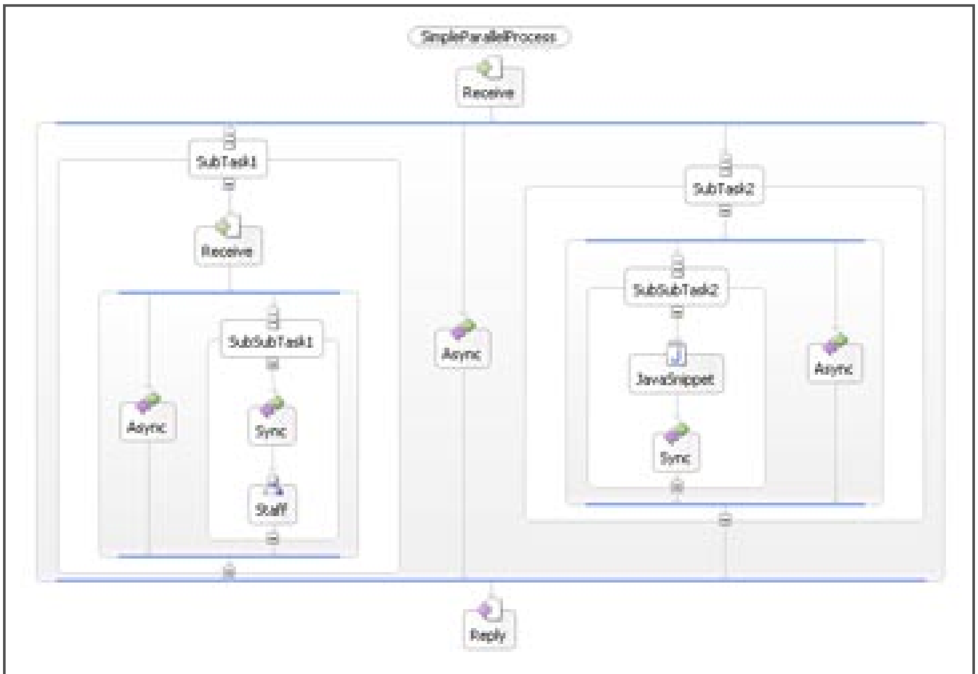


FIG 1: A COMPLEX BUSINESS MODEL

words, the feedback mechanisms through polling and/or writing to a globally well-known location are all handled as part of the business process itself thanks to BPEL. The business process engine (the infrastructure that will eventually be executing the model) seamlessly keeps track of all appropriately labeled, contextually relevant data that needs to be communicated between two or more cooperating services, regardless of whether these services are local or hosted by external business partners.

COMPENSATION HANDLER

Going back to the possibility of a highly complex business model, what happens if something goes wrong? Without going into as much detail as I did for parallel tasks, just imagine how difficult it would be to correctly write code to handle exceptions at the various levels of nesting for a larger process. If something fails along the way in one branch, have you adequately written a framework for notifying other branches that they should stop executing and begin the rollback process? Furthermore, have you been recording what you have done cumulatively up until that point so you can perform a logical rollback? Perhaps. I'm not saying it's impossible, and the experienced programmer might be bold enough to think it is not too difficult to handle such a case.

The BPEL language, however, has again come to the rescue with the compensation activity. Here a "reverse" action can be programmed for any given scope within a process. If some business rule fails or a fault is generated, a series of compensation activities can automatically take place, rolling back the process to some previous, stable state. The process engine ensures that the activities are called in the correct reverse order to completely undo all uncommitted or unfinished work up until that point, regardless of whether this happened in a process that spans seconds, hours, days, weeks, or even months. The duration of the process is not really an issue anymore; the tools focus on the interaction. You do not directly see the compensation handlers in Figure 1, but the handlers can be and are associated with some task or scope. And no, these compensation activities are not exactly the same as your good old exception handlers. As I said above, they are associated with long-running business processes, not just segments of code. They go on to further dwarf their historic counterparts with their ability to span disparate services boundaries.

It cannot be denied that time is saved through the ability to construct these models and all their levels of complex flow control through an intuitive, graphical environment, which can also explain why problems with maintenance would be dramatically decreased too. If a part of the model changes down the line and requires slightly different compensation activities, or needs more or less communication to or from another process, trying to look at the hundreds if not thousands of lines of code that it would take to accurately represent the entire model (had this been written by hand) should be frightening. Again, the fact that everything can be configured and quickly changed through this graphical representation not only helps to make it possible to write and maintain complex business processes, but also helps to make it simpler for fewer technical personnel, such as the business analyst, to validate and verify that each model correctly represents what happens or what needs to happen in real life. Remember how long it took for me to create the process in Figure 1? That is not necessarily uncharacteristic of the time it would take to update the model in the future. Did someone say "on demand"?

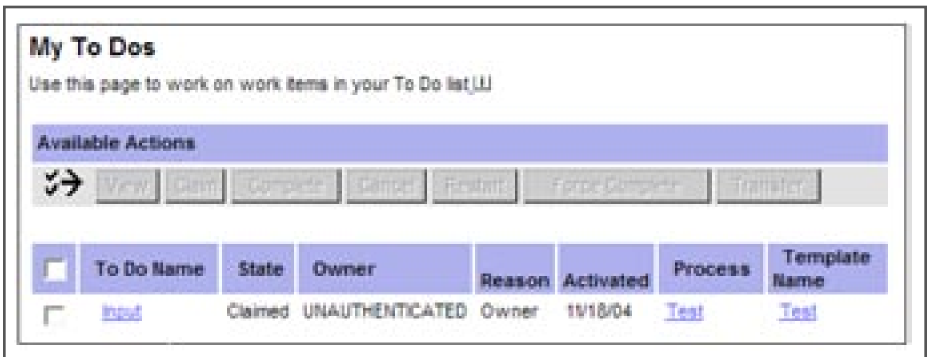


FIG 2: QUEUE SHOWING A SINGLE "INPUT" TASK

Forward Thinking

BPEL was never meant to solve every conceivable issue that you might encounter while writing the definition for a business process. It was only meant to catch most of them in a standards-based way. To that end, there would not be much reason to go with any one vendor rather than another unless one of them were able to differentiate itself in the marketplace and provide unique value to the product and an enhancement to the standard. IBM has done just that by introducing a series of powerful and necessary extensions to BPEL.

STAFF ACTIVITY

In my opinion the most important extension is the staff activity. In standard BPEL, there is no notion of manual intervention in a process, nothing that mediates a human interacting with the system. However, this requirement will tend to occur time and time again and must be dealt with. For instance, consider a system that needs a final "yes" or "no" on a loan application before the funds can be reserved, or another system that requires a human analysis of the results of a background check before considering someone for employment. Using this extension, operators at various stages of a process can conveniently retrieve work from a queue, where it can be handled in absolutely any order.

Figure 2 shows you exactly how this queue is used. An employee could log into the system and first "claim" the task; in other words, that he or she is assigned the manual work and promises to perform the required user intervention. Once the line item is claimed, the person can choose to perform the work needed, view additional information about the current item, return it to the pool for a different employee to pick up, or even transfer it to someone else who might need to do more work on it. From a modeling perspective, Figure 2 shows just how easy this is to set up. The staff activity was created just like any other design element, and so is effortlessly dropped onto the palette and quickly incorporated into the complex process.

Furthermore, by integrating these manual activities into long-running processes, there is no concept of blocking in the traditional sense. The next process or service in the flow of the model is no longer waiting indefinitely to receive user data, wasting precious CPU cycles and forcing it to repeatedly switch contexts back and forth between services that have no data yet to receive. This common notion in complex models has been developed into its own feature with the full capabilities of any synchronous process (one that executes on its own, without user interaction) – the matter of synchronicity is now delegated to the staff personnel interacting with the system.

"Web services and SOA provide the pathways or the map, but not the specific direction in which to head nor the specific route to take"

JAVA SNIPPET

Finally, the Java snippet cleans up where standard BPEL and staff activities cannot. Realize that the BPEL standard did not necessarily leave anything out. No standard, regardless of what technological idea it characterizes, should ever try to be too emcompassing. BPEL did its job of defining the core competencies of the solution and left the details up to the implementers to customize certain parts of the technology for what they believe would benefit their customers most, or what niches of the market they wanted to delve into. (Thus, the staff activity was born, but it is so ubiquitous and clearly needed that it really seems to have been accidentally left out of the standard, as opposed to strategically chosen to not be put in.)

For any other shortcomings of the BPEL standard, IBM solved the problem with the Java snippet. Each snippet is compiled as a method within a Java class that is attached to the actual business process itself. Since RMI isn't actually used in this case, a Java snippet can be a form of optimization. Yes, this entire article does revolve around the fact that Web services are great for reuse, integration, asynchronous modeling, etc. However, it cannot be ignored that Web services projects come at a particular cost. Deserialization and parsing of incoming messages, as well as serialization of outgoing messages (the three major steps in Web services communications) come at a strikingly high cost, compared to other existing methodologies. Sometimes, if the cost cannot be justified, or simply cannot be ignored because of some critical functionality or chokepoint in the system, IBM's BPEL engine seizes the day and lets you refactor that slow running process into a piece of Java code, which is executed within the same container as the processing engine, has direct access to any variables in scope in the process, and can still make calls out to external resources such as databases, Web sites, or EJBs, etc. There is complete flexibility.

Skyscrapers, not Mazes

This software has far-reaching implications and can dramatically reduce the amount of time needed to implement business models, regardless of their complexity. The most important thing to realize is that Modeler and Process Choreographer are tools that, in addition to building Web services, focus on their interaction. Building Web services in an ad hoc fashion will produce a product that might, more or less, resemble a maze. Each of the services represents a wall, and in this bottom-up fashion, hadn't been thoroughly thought about in terms of their position in the big picture. The WebSphere BI tools force the analysts and developers to create systems using a top-down approach, where you can gain fine control of the blueprint and build a truly impressive, finely polished structure – a skyscraper, not a maze.

Where to Proceed

As this technology is relatively new, there isn't an abundant amount of material on it. However, since I am a big advocate of Redbooks for self-paced study (and if you want to learn more about the product, its related products, and their full capabilities and features), I suggest you begin by heading to www.ibm.com/redbooks and downloading "sg246381 - BPEL4WS Business Processes with WebSphere Business Integration: Understanding, Modeling, Migrating." As of this writing it was still in draft form, but it gives an extensive overview of BPEL, its history, the standard, and of course, the IBM extensions. From the same site, if you search on topics such as "business integration" or "server foundation," you will find more documents describing how process choreography – as a general term, not the product – can be leveraged to produce systems capable of quickly adapting to ever-changing business models.

LISTING 1: PARALLELISM WITH ASYNCHRONOUS METHOD CALLS

```
asyncOne();
asyncTwo();

.
.
. // other asynchronous processes

while (/* some result still need */) {
   // blocked, awaiting finish of some asynchronous
      process
   int result = listen ();
   if (result == RESPONSE_TYPE_1) {
         // do some processing
   } else if (result == RESPONSE_TYPE_2) {
         // do some other processing
   } else ... {
         // yet more asynchronous process handling
   }
   // but how do you cancel one of these processes?
}
```

LISTING 2: LIGHTWEIGHT THREAD WRAPPER

```
abstract class SyncToAsync extends Thread {

   protected Object result;

   public Object getResult() {
         return result;
   }

   // used to force cancellation of processing
   public void cancel () {
         try {
                  this.join();
         } catch (InterruptedException ie) {
                  // ignore purposely
         }
         // perform some logical rollback
   }

   public boolean finished() {
         return (result != null);
   }
}
```

LISTING 3: PSEUDOPARALLELISM WITH WRAPPED SYNCHRONOUS METHOD CALLS

```
SyncToAsync asyncOne =
   new SyncToAsync ()   {
         public void run() {
                  // do some synchronous
                           processing
                  result = doSyncOne();
   }
   };

SyncToAsync asyncTwo =
   new SyncToAsync()   {
         public void run() {
                  // do some synchronous
                          processing
                  result = doSyncTwo();
   }
   };

asyncOne.start();
asyncTwo.start();
.
.
. // other asynchronous processes

while (/* some result still needed */) {
   if (asyncOne.finished()) {
   /* a particular result from asyncOne happens to
      short-circuit asyncTwo */
   if (someTest(asyncOne.getResult())) {
         asyncTwo.cancel();
         break;
   }
   } else if (asyncTwo.finished()) {
         // do some other processing
   } else ... {
         // yet more asynchronous process handling
   }
}
```

Looking Beyond MQ

Improve levels of service and avoid costly upgrades

BY DAVID MAVASHEV & CHRIS HOLLAND

David Mavashev is CEO for Nastel Technologies, Inc., where he develops and maintains the corporate and technology vision. He has extensive experience in enterprise systems management specifically in the area of performance monitoring of middleware technologies.
dmavashev@nastel.com

It's likely you've been working with WebSphere MQ (WMQ) for years developing, deploying, monitoring, or all of the above. It's also likely that by now you have assembled a tool bag full of items to support your implementations and ongoing operations. And you've no doubt become accustomed to dealing with problems within the MQ domain. But what if you could see more of the transactional journey on either side of MQ? Organizations doing this are finding new ways to improve levels of service and avoid costly upgrades.

Initially, just seeing the touch points, you know - the "puts" and the "gets" – along with MQ administration probably seemed entirely sufficient in order to manage WMQ. A common assumption was that as long as we saw messages coming and going, things were okay. Analogous to Archimedes' principle of water displacement, the health of MQ or any other middleware system could be assumed as long as both sides remained within a reasonable state of balance.

But it didn't take very long at all to see that more was needed. The difficulty was that by the time an imbalance was noticed, too many problems had already been caused; it was obvious that an earlier warning was needed to avert problems or at least correct them sooner. The need to watch things like queue depth, channel status, or any of the other 40 standard events that MQ raises became clear and pretty much commonplace.

If you are still requiring more precise information to detect problems earlier and to see their impact on the applications and business processes, some of you may have had the experience of configuring your own new custom events based on conditions within your unique MQ implementation. If you're really on top of the MQ management game, you've implemented automatic corrective actions, which launch the moment warning signs appear – preventing problems before they occur.

The More Things Change...

Despite your success implementing and managing WMQ, and at the risk of stating the obvious, your needs continue to change due to some constant forces:

- ***Increasing EAI complexity.*** Even though most have implemented WMQ for a specific single project, the need to integrate it with other applications and systems continues to grow.
- ***The need to do more with less.*** Due to decreasing resource availability to manage the complexity with smarter tools and processes.
- ***The demand for higher levels of service.*** Not just to track and report on the level of service delivered, but also to resolve problems faster – in real time before they impact the business.
- ***The requirement to align IT services with business objectives.*** Delivering and supporting the service specifically to the needs of the business process. Amid a push to run faster and jump higher while carrying heavier loads, we commonly find this mistake: looking over the most practical solutions. What is the most practical solution to these challenges? We've seen that one of the most straightforward and effective things you can do is to expand your monitoring of middleware to include more of the entire application infrastructure.

Achieving An Application Perspective

If WMQ is the only middleware technology you have, you are unique. If you have any enterprise applications (commercial or homegrown) that do not also talk to Oracle, DB2, or SQL Server, then let's just say that you are in a class of your own! Unless we've just described your environment, you also have a wide collection of management and monitoring tools for all of these various platforms and systems – each of which lacks the ability to see or do much of anything beyond its own domain.

The solution is to monitor and manage more of the overall application infrastructure. It may sound like the holy grail to track communications across all of your applications and through all of your middleware systems as a contiguous whole, while at the same time correlating the events generated by each platform involved. However, the fact is that it is being done – and it's much simpler to do than most expect.

We found that to be the case at Debenhams, one of the United Kingdom's preeminent retailers, and saw an opportunity to improve our performance. Diagnosis of the problem wasn't really the issue – all the information we needed was there. But it was just taking too long to put it all together.

We had been doing a good job of sorting out problems within the MQ space for some time. Our problem was that most of our time was spent traipsing through the API layers of our applications that are integrated into MQ in order to find problems. The difficulty was to see the whole picture. Think of it as pushing sausage meat along a sausage casing without any knots in it. We never knew quite where we were.

This is not an isolated problem. Often "blind spots" in the round trip of messages make it difficult to see just where the hang-ups are. Even more troublesome is the cover that these blind spots usually provide for the vendors involved to play the finger-pointing game while your valuable time is being wasted.

Expanding The Myopic MQ Perspective

At Debenhams we didn't have to look far for a solution. From our experience with Nastel AutoPilot, our existing MQ monitoring solution and our dialogue with its vendor, we saw how easily this type of problem could be handled. With all of the facts we could publish with the agents we had already deployed across our AS/400s, we were easily able to get access to all of these other points of information.

When you've got the right monitoring platform, the biggest task can often be defining what data metrics you want to monitor. But, if the tool you are using for monitoring is built on a service-oriented architecture such that it supports open standard interfaces and treats each metric as a portable object - each with its own metadata and methods – and if it makes it easy to define your own new events, then you should be able to quickly and easily monitor any data metric with rule and correlation engines, and invoke automated or manual corrective actions.

For example, in Debenhams' case, there are three business-critical applications "glued" together by WMQ. This allows them to see the end-to-end transactional journey involved in correlating facts from the API layers of their application with those they were already getting from MQ.

As you can see in Figure 1, both the applications and their API layers are being monitored – all the way into the database that supports the application, and facts are being published along the way. The API layer can initiate a piece of work that can take a good deal of time. For example, if we are processing price changes for a substantial department having tens of thousands of SKUs, it can take up to half an hour to complete. We need to know this beforehand, as opposed to those jobs that are perceived to be long-running and turn out to be only one hundred records or so.

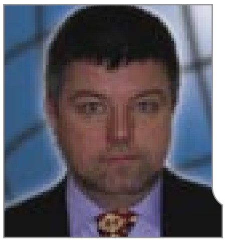
So what was it like implementing all of this monitoring? It took a bit of work, but don't get the idea that it was difficult or impractical. It was relatively straightforward with the tools we used at Debenhams. We've got one guy who's written some generic code on the AS400, which is hooked up to how we publish the facts with our monitoring system. We have several different sorts of things we're collecting. For example, you can easily find out how many records have been received. So that's an easy sort of fact to publish. Some of the other facts that we are getting from the API layer are far more complex. So it's actually working out things like, what processes are active, and if the processes are active, then we need to interrogate the files that they're reading to see how the queue depths are going.

The Payback

The expanded visibility of seeing beyond the bounds of your middleware (when you begin to see more of what is happening within your applications) will always yield better ways to monitor and tune the system and make smarter use of that system, too.

At Debenhams, we estimate that our monitoring saves us at least two to three hours in staff time every day. The most sizable savings is in the time it takes to find problems. For example, our application's vendor used to ask, "How did you know we were doing that?" But they work very closely with us and have been on-site when we were working out the problems inside of MQ. Therefore, they saw the tool we have and learned that we also have quite a lot of MQ knowledge. So now when we think we've got a problem, they don't question it – they just go fix it.

During Debenhams' busy time of the year, they run



Christopher W.J. Holland is the technical integration manager at Debenhams Retail PLC where he has worked in various positions, most recently in data warehousing and technical architectural capacities. He has designed and implemented Debenhams' MQSeries infrastructure and is involved in numerous technical initiatives across supply chain, buying and merchandising, B2B, and B2C areas.
chris.holland@debenhams.com

"we estimate that our monitoring saves us at least two to three hours in staff time every day"

"we saved thousands of dollars that we would have otherwise wasted just treating the symptoms"

more shifts and the volumes start going up. Chris' group wanted to be able to isolate which particular jobs were causing an issue in the API layers. Once they were identified, the group could then see if they could either throw more resources at them or get them rewritten. As a result of this visibility, we saved thousands of dollars that we would have otherwise wasted just treating the symptoms.

One of the critical processes within Debenhams' operation involved a relatively simple transformation on the source application where the data gets pushed through MQ quite quickly when users gathering the data come off their shifts. Within the target application, however, there is a rather tortuous route. It was at this point in the process that we were finding some problems. It goes through a number of files and a number of rigorous data transformations and eventually it gets into the target database, and then pops out the other side. Not only were we able to easily see what needed to be fixed, but we also extended visibility of the process to our users. Now they see the path of transactions pictorially, and can watch the whole process – easily spotting problems without having to understand any of the complexities.

Best Practices

So where do you start with this practical approach to monitoring your application infrastructure in a proactive manner? Here are the ways you can make it happen, and get the visibility to go beyond mere MQ monitoring:

- Use a monitoring platform that is an SOA designed to operate in real time, and is modular and extensible with support for any open standard.
- Identify specific data metrics (facts) that will give you better insight into the health of your system. This has a compound effect in that the more you can watch these metrics, the more you'll discover those points of data that are the real determinants. Remember the facts about your environment are all out there; you just need to locate them and pull them together in order to monitor the health of your system.
- Build simple modular views of these metrics, so that you can correlate them with each other into a hierarchy that represents the complex events within your system. This way you'll see the warning signs of problems before your application goes to production (or before your customers pick up the phone, if it's already there). Most of all, you'll spend much less time chasing false alarms.
- Share the visibility with other stakeholders of the system, particularly application groups and business units. You'll empower them to solve many of their own problems without bothering you. It also reduces time wasted on needless mundane inquiries, and makes the questions you do get of a higher quality.
- Implement corrective actions for those recurring conditions where the resolution is consistently the same set of actions.

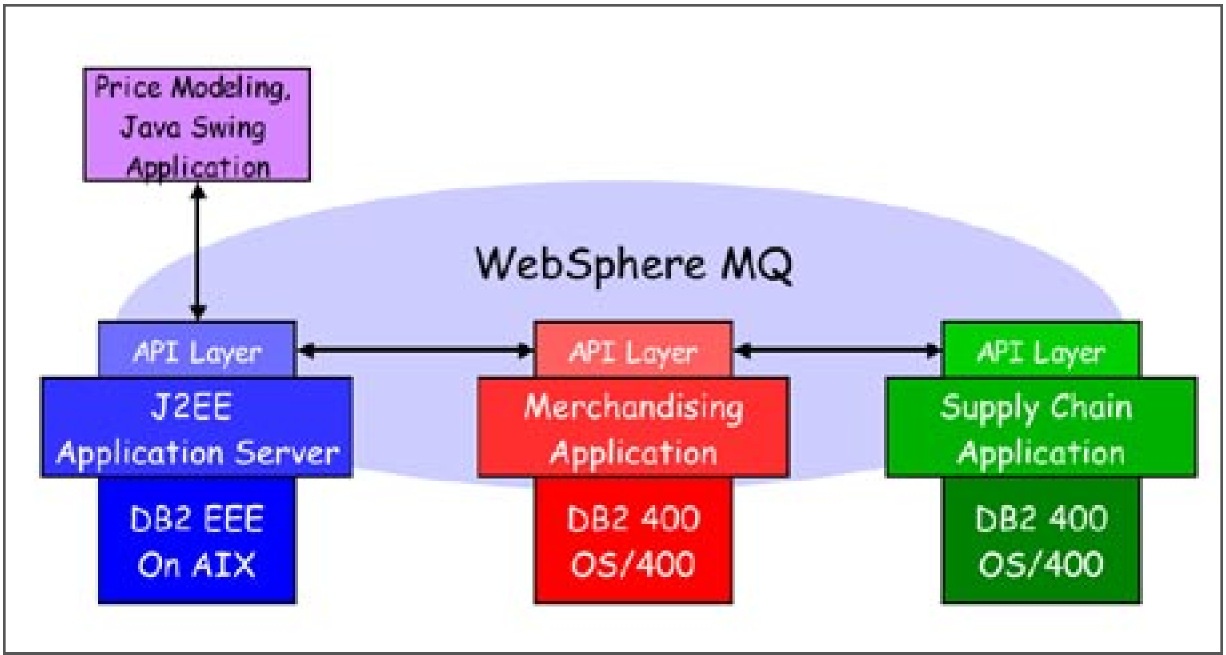


FIG 1: LOOKING BEYOND MQ AT DEBENHAMS

Ignorance isn't bliss.

Knowing about performance problems before they happen is.

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

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

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

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

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

diagnoSys

Register for an upcoming Web seminar. Find ways to improve your Web app projects while driving profitability. Guest speakers share their knowledge. Register now at: www.hwcs.com/wsj16.asp

H&W

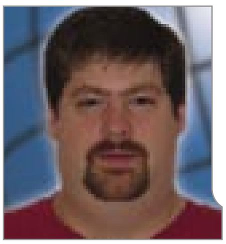
www.hwcs.com | 1.800.338.6692

Bridging the Enterprise Computing Gap For 25 Years

Proven solutions for software promotion

Managing Environment-Specific Properties

BY JASON NEWMAN
& DAVE JONES



Jason Newman is a Java architect with Perficient, Inc. He has been working with J2EE technologies since 1999, and the WebSphere Application Server since version 3.02.
newman@perficient.com

WebSphere applications often rely on property files to contain environment-specific values. These files usually contain any settings that change between environments, and the files can be in a variety of formats – Java properties (key/value pairs), XML, or a custom format. Many values can be kept the same throughout the environments, but values such as host names, user IDs and passwords, database schema names, and URLs usually change in each.

We see many of our customers struggling with the problem of how to manage the changes in application property files as the application gets promoted through a number of different environments, such as development, testing, and production. This article will outline some of the more common and proven solutions. While none of these solutions will always be the best for all situations, it is likely that one of the solutions presented here will fit your needs.

Roles and Responsibilities

In a perfect world, groups within an organization would map nicely to the roles and responsibilities defined by J2EE. In reality, a disproportionate amount of the responsibilities fall within the realm of development. We find this especially true for property files where Operations' only responsibility may be to change a password, and development creates and maintains all of the remaining properties.

Previously many customers tended to manage all of the property files externally to the application, often deploying and updating these files by hand. Now it's more common for development teams to create property files for each environment and package them all into the EAR file – building their applications to use a cue from each runtime environment to indicate which properties are to be used in that environment. We have found that roles and responsibilities generally have the largest impact on which of these solutions work best within an individual organization.

Options

We have seen environment-specific properties managed in a number of ways; many of them are outlined in Table 1.

How to Load Them?

AT BUILD TIME USING ANT

There are two common strategies for replacing environment-specific information at build time; (1) replacing the entire file with files marked for each environment, or (2) replacing tokens in all property files from an environment-specific name/value pair. Both of these techniques can easily be done using Ant.

The following copy task shows how you can use Ant to replace all property files that end in a specific extension (such as ".dev", or ".stage") to another directory while removing the extension. The extension to replace is specified by the property ${env}, and could be passed into Ant when the build is invoked. The <mapper> element is the key to the copy command to perform the replacement.

This method is useful when your property files vary greatly between different environments, such as log4j.property files, which generally contain entirely different logging strategies and configurations in production than development.

```
<copy todir="${destdir}"
    overwrite="true">
  <fileset dir="${srcdir}"
    includes="**/*.${env}"/>
  <mapper type="glob"
     from="*.${env}" to="*"/>
</copy>
```

You may also want to first copy all files except those ending in any of the environment-specific endings, so that any property files that don't require an environment-specific ending are picked up.

```
<copy todir="${destdir}">
  <fileset dir="${srcdir}">
        <exclude name="**/*.
            dev"/>
```

"In a perfect world, groups within an organization would map nicely to the roles and responsibilities defined by J2EE"

```
        <exclude name="**/*.
          stage"/>
        <exclude name="**/*.
          prod"/>
  </fileset>
</copy>
```

An alternate method is to replace tokens among all property files that would specify environment-specific values. This method is more useful if you have just a few substitutions to make in many different files. The <replace> task accepts a file (replacefilterfile) which contains a lists of tokens and the values to substitute. Again, if the ${env} property is set to "DEV" when invoking Ant, the following example would replace the tokens specified in the replace.properties.DEV in all of the .property and .xml files, such as "sample.properties".
build.xml:

```
<replace dir="${build.dir}/proper
    ties"
replacefilterfile="replace.
    properties.${env}">
  <include name="**/*.proper
    ties"/>
  <include name="**/*.xml"/>
</replace>
```

replace.properties.DEV:

```
@@SERVER_NAME@@=mydevserver
@@USER_NAME@@=devuser
```

sample.properties:

```
connect.url=@@SERVER_NAME@@
connect.username=@@USER_NAME@@
```

These two techniques could be used together if your application contains property files that fit into both categories. In this case, the file copy method should be used first to move the files into a build directory, and then the replace task applied to substitute tokens within those files.

The disadvantage of this process is that unique EAR files need to be built for each environment. This could conflict with your organization's policies on configuration management, which likely require the EAR file to be locked after successful testing. Alternatively, since some organizations build fresh for each environment from frozen source, this technique might work well.

AT RUNTIME USING RESOURCE BUNDLES

Java provides a simple, out-of-the-box solution which can be exploited to easily load the correct files dynamically at runtime. By setting a "variant" on the WebSphere Application Server instance, the ResourceBundle class will load the appropriate property file with no additional configuration necessary short of naming your property files correctly. The ResourceBundle class should load the correct property file as long as the language and locale on your servers matches what you have named the file.

The variant is set by passing the "-Duser.variant" property on the JVM command line. This can be set on a server instance from the WebSphere admin console. The argument is added on the Application Server's -> <servername> -> Process Definition page under the field labeled Generic JVM arguments (see Figure 1).

Once set for a server, the following code can be used to load an environment specific property file.

```
ResourceBundle rb =
   ResourceBundle.getBundle("mypro
     perties");
String url =
  rb.getString("connect.url");
```

On a server installed on a machine set to the use the English language and US locale, the ResourceBundle will first look for a file named:

```
myproperties_en_US_dev.properties
```

and will then move up the filename, until it will finally try to load a file called:

```
myproperties.properties
```

The downside to this approach is that it can be confusing to mix language/location strategies with environment determinations. It can also potentially conflict with third-party utilities that use the variant to further delineate the locale of the runtime environment.

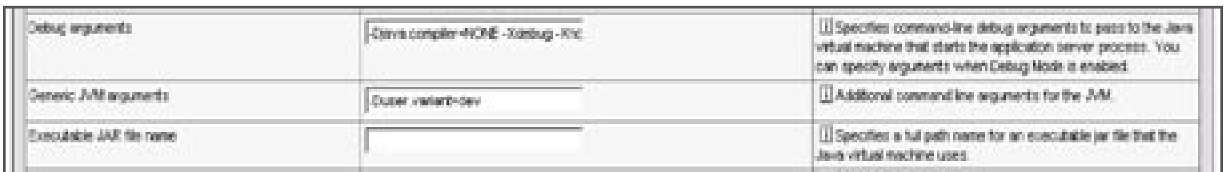


FIG 1: APPLICATION SERVER PROCESS DEFINITION PAGE

Dave Jones is a WebSphere specialist at Perficient. He has been working with Java since 1997, and has been consulting on WebSphere and server-side Java since 1999 dave.jones@perficient.com

OPTION	ADVANTAGES	DISADVANTAGES	PROVIDED BY
Build-time solution	Only one set of properties exist in deployed EAR (no unused properties – less confusing)	Assumes consistent and reproducible build process already exists.	Development/Build team build process
		Usually will need a new build for every environment.	
Resource Bundles	Flexible runtime loading options	Limited to Java properties files and compiled resource bundles	Java language
Name space bindings (JNDI and URLs)	Can be configured through Deployment Manager at the cell level (do not have to change on each machine)	Must be done through a WebSphere administrative interface (requires code or administration console	J2EE, WebSphere
		File URLs will usually point to a location outside of the EAR, meaning a separate deploy process for the properties files.	
J2EE Environment Entries	Recommended and built-in with J2EE.	Module specific (must be defined for each module that references it)	J2EE
	IDE support	Can only be referenced by J2EE module (not utility jars)	
		Requires additional step during packaging to set entries. Those entries are probably saved in a properties file anyway.	

TABLE 1: COMPARISION OF ENVIRONMENT-SPECIFIC PROPERTY MANAGEMENT OPTIONS

AT RUNTIME USING JNDI RESOURCE ENVIRONMENT REFERENCES

JNDI can also be used to store environment-specific information, and can be referenced by your application through local references just as EJBs or DataSources are. These bindings can be used to provide specific string values, or URL references to property files containing environment-specific information located on the server (see Figure 2).

These references are referred to as Resource Environment References, and are specified with the <resource-env-ref> tag in the Web or EJB deployment descriptors.

If you wish to use these references with the WSAD test environment, a WebSphere binding must be specified in the Deployment Descriptor wizard, and the appropriate binding defined in your local test server.

Unfortunately, there is no place in the WSAD Test Server Configuration wizard to set these references. You must use the administration console to create the namespace bindings for both WAS and the WSAD test environment. The binding can be set from Environment >Naming >Name Space Bindings page (see Figure 3).

Be aware of the scope on which you set the environment references. Although you may want to set a reference on a cell basis, to cover all the machines in a cluster you will need to point the binding to the correct place (namely, "cells/persistent/<x>"), whereas server-scoped namespace bindings will appear right in the root.

Looking up the references is no differen t than looking up any local environment reference:

```
Context ctx = new
    InitialContext();
String propFile = (String)ctx.
    lookup("java:comp/env/url/
       properties");
```

If you want to comply with the J2EE specifications, once you have obtained the reference it should be read via the java.net.URL class instead using direct file IO.

```
URL propFileURL = new
```

```
    URL(propFile);
InputStream in = propFileURL.open
    Stream();
```

AT RUNTIME USING A CUSTOM PROPERTY MANAGER

Writing your own custom code to load environment-specific properties has some advantages, such as the ability to include logging or dynamic reloading on file change, but there are a few things to keep in mind if you roll your own:

- ***Beware of classpath isolations.*** Chances are you are going to want to read in those property files off the classpath, and not directly off the filesystem. Because of the way WebSphere 5.x uses multiple class-loaders for different parts of your application, it is important that you use the correct classloader for the job. If you place this property manager code in a utility JAR, you probably don't want to use the classloader that loaded the property manager. Instead, you would want to use the classloader of the component that called the property manager. A good way to do this is to use the context classloader:

```
Thread.currentThread().getCon
    textClassLoader().getResour
    ceAsStream();
```

  WebSphere ensures that the context classloader is set appropriately before each call into your application.

- ***Consider building hierarchical property file searches (like ResourceBundle).*** Instead of just appending a single environment token on to the end of any property file searches, consider building your search to navigate up a sequence of extensions, ending with no extensions. This will allow developers to scope their property files as necessary. A simple example would be if your custom property manager is asked to load "myprops.properties" while running in the production environment, it first looks for "myprops.properties.prod" and then "myprops.properties." In a clustered environment, you may want to indicate a specific server in a cluster, along with the environment type itself, such as "myprops.properties.prod.pdserver1."

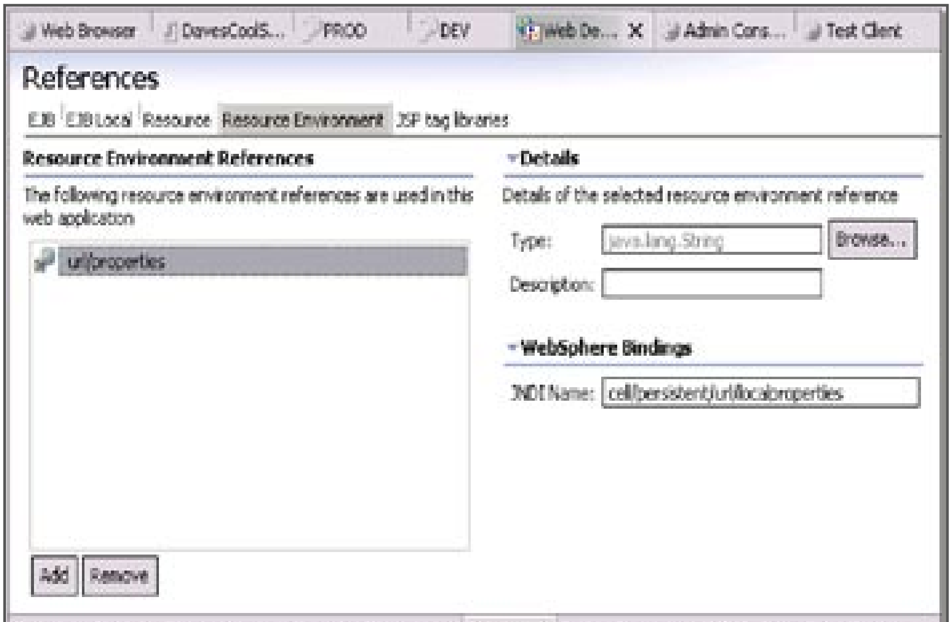


FIG 2: RESOURCE ENVIRONMENT REFERENCES

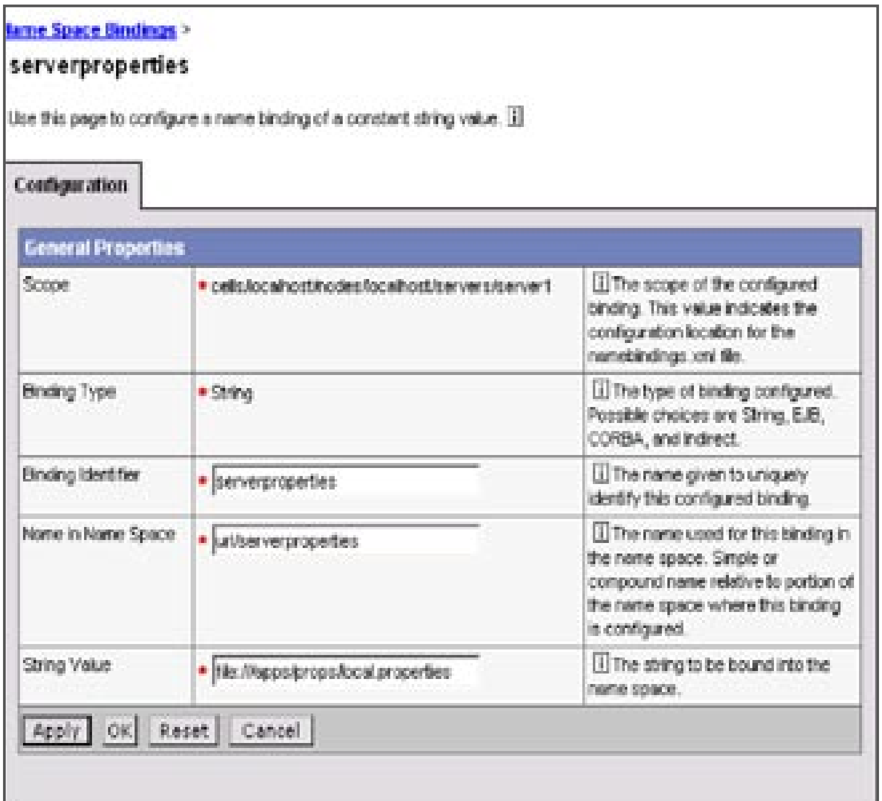


FIG 3: CONFIGURE NAMESPACE BINDINGS

Where to Put Them

A common problem in J2EE applications is deciding on where to place properties files. There are three main options, which will be discussed

briefly here. Once one of these options is chosen, then an appropriate access method will need to be selected (see Table 2).

OPTION	ADVANTAGES	DISADVANTAGES
Properties files external to EAR	Easy to modify at runtime	Additional build and deployment steps required for the properties files since they are not a part of the EAR.
	Easy to implement	May require granting of Java 2 security privileges.
Properties files kept in root of EAR, but not in JAR	Easy to modify at runtime	Relies on feature that is "encouraged", but not required, by J2EE.
	Packaged with EAR file during deploy	Solution is slightly different for WSAD and WebSphere.
		Harder to reuse the module/project that loads the properties file (discussed later)
Properties files in JAR in EAR	Complies completely with J2EE specification (no assumptions about manifest. mf support)	Cannot view or edit properties as easily after the application is deployed.

TABLE 2: COMPARISION OF POSSIBLE PROPERTY FILE LOCATIONS

External to the EAR File

Most early J2EE applications accessed property files that were external to the EAR. This was mainly due to ease of implementation.

The major disadvantage of having any application resources external to the EAR is that there will be additional packaging and deployment steps since all assets are not a part of the EAR.

On the other hand, in environments where environment-specific values are closely managed and guarded by Operations this approach could be beneficial. The development team would define what the property file should look like, as well as how it should be accessed (e.g., direct file system access or via classpath), and the Operations group would create and place the file accordingly.

This method works best with the JNDI Resource References, as the administrators can bind the location of the files at deployment time while the application code can utilize J2EE compliant loading by using the URL class instead of direct java.io package loading.

USING A PROPERTIES JAR FILE

If you are not worried about easily editing the property files on the WebSphere servers themselves after deployment, the easiest way to bundle property files on the classpath is to create a properties JAR file and add it to your application's EAR.

An important step to remember is to add the property JAR to the manifest classpath of any component that will be accessing the properties.

This approach accomplishes two common major goals of having the EAR self-contained and being a fully J2EE-compliant implementation.

PROPERTIES FILES IN ROOT OF THE EAR

Some WebSphere shops have the requirement that properties be packaged in the EAR, yet also be accessible without having to extract them from a JAR. They do this to allow for easy checking or changing of the properties. This solution relies on the classloader's ability to search through non-JAR resources in the EAR.

This solution could be considered "not fully J2EE compliant." It relies

	ACCESS METHODS			
WHERE PLACED	java.io APIs (direct file system access)	ResourceBundle (classpath)	URL	J2EE APIs
External to EAR	X	X	X	
In JAR in EAR		X		
Properties files kept in root of EAR, but not in JAR		X		
J2EE environment entries				

TABLE 3: MATRIX OF COMPATIBLE PROPERTY LOCATIONS TO LOADING METHODS

"Now it's more common for development teams to create property files for each environment and package them all into the EAR file"

on a feature that is recommended, but not mandated by J2EE. The following excerpt from section 8.1.1.2 of the J2EE 1.3 specification explains this recommendation:

J2EE products need not support this mechanism for reference to classes or resources that are not in .jar files included in the .ear file. However, support for such usage is encouraged. Applications are encouraged to make use of such support only when necessary and where available, although currently such usage is non-portable.

WebSphere has implemented this functionality since version 4.0.x, however, since J2EE has not mandated it, IBM could withdraw this functionality at any time.

1. Create a directory named "properties" in the root of the EAR project. This directory will be even with the META-INF directory.
2. Place all properties files in this "properties" directory.
3. Write code to access the properties file. This would normally be done with

```
Thread.currentThread().getContext
  ClassLoader().getResourceAs
    Stream(propFileName);
```

This code should normally be placed in a utility JAR that can be referenced by all other modules of the enterprise application.

4. For each module that will access the properties file, make the following changes to the MANIFEST.MF file:

```
Manifest-Version: 1.0
Class-Path:../<NameOfEarProject/
  properties properties
```

WSAD will show errors on the EAR project due to the entries in the manifest, but the errors can be ignored. There are two entries in the manifest file because WSAD loads code from the workspace differently than WebSphere Application Server loads it at runtime.

The "../<NameOfEarProject/properties" entry is for execution within the WSAD test environment. If you not using the WSAD test environment, this entry in the manifest can be excluded. This tells the classloader to look up one directory (to the workspace root) and in the <NameOfEarProject> directory (the folder for the EAR project). This has to be done because WSAD treats all projects/modules as being at the same level in the hierarchy. In other words, WSAD does not treat the modules and JARs as though the EAR project contains them.

The "properties" entry is for runtime in WebSphere Application Server. In WAS, the EAR is the top level with respect to the modules' classpaths, and each module is one level deeper. The modules are extracted to the disk in the same hierarchy in which they are stored within the EAR file.

There should be no adverse effect of having both entries in the classpath permanently. A conflict could occur in WSAD if there were a "properties" directory in the module itself (unlikely). A conflict in WebSphere Application Server would occur if "../<NameOfEarProject/etc" existed relative to the installed location of the EAR (extremely unlikely). Either of these scenarios would only occur if it were created intentionally.

As you may have observed, problems may occur when reusing the component with the reference to the properties directory. In WSAD, the manifest will reference a directory name that contains the name of the EAR project. This will be a problem when the module needs to be reused in multiple EARs.

Pick Your Location Carefully

Not all property file locations will work with all methods to load properties. Table 3 indicates which methods can work together.

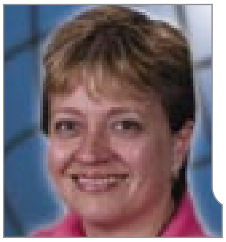
As you can see, there is a variety of approaches for managing environment-specific properties. Most of the companies that use WebSphere are using one of the options mentioned in this article, or a slight variation of one of them. It is nearly impossible to say that one method is always desirable over the others; hopefully this article presented the information in such a way that you can choose the approach that works best for your organization.

IBM WebSphere Remote Server

Do More – Faster

BY PAMELA KLYM
& JERRI TRAFLET

Pamela Klym is market manager for the retail industry for IBM Software Group.



Jerri Traflet is a solutions executive for the retail industry for IBM.

After years of spending their IT dollars to help them cut costs, comply with Y2K requirements, and compete in a world of multiple ways to shop, retail companies are ready to re-invest in their number one channel – their stores.

In fact, they must invest in their stores to continue cutting costs and increase revenue in a relentlessly competitive industry grappling with a dizzying number of products and customers who want more convenience, higher quality, and better service, but not necessarily higher prices.

Technology exists, however, for retailers to bring true business-altering capabilities to their stores. In fact, some retailers are already using technologies such as radio frequency identification (RFID), digital media, transaction log polling (TLOG), and more to improve and update their stores.

These kinds of capabilities can truly transform the shopping experience and are just as important for efficient, cost-effective management of retail companies' processes fully across their businesses, from headquarters to the stores to the Web.

"Retailers must provide a much more dynamic environment for today's customers who expect to have the information and products they want at their fingertips. In a word, they need to Web-ify the store," said Pamela Klym, market manager for the retail industry for IBM Software Group. "In addition, they need to manage processes more at the store level – to monitor, update, and act on sales and customer data, and learn about issues early for quick problem solving."

To do so, retailers need an IT infrastructure able to manage new capabilities along with existing systems. IBM's Store Integration Framework provides a structure for deploying customized combinations of middleware solutions, specialized applications from independent software vendors (ISVs), and industry-specific services to allow retailers to be more responsive to business opportunities and customer demands.

"Store Integration Framework is built on open standards for efficient integration with existing systems, multiple platforms, and ISV applications, yet it is configured specifically for retail store operations," said Jerri Traflet, a solutions executive for the IBM retail industry.

A key component of the Store Integration Framework is the IBM WebSphere Remote Server, which leverages the reliable infrastructure IBM delivers at the enterprise, but configures and sizes it specifically to meet the requirements of the store environment. It allows retail customers to extend and integrate their enterprise technology into their stores so they can fully manage retail operations as a single entity.

The IBM WebSphere Remote Server provides the infrastructure that enables retailers to respond as needed to remove cost from their businesses, increase employee productivity, and create a unique shopping experience for their customers based on new technologies. It is built from IBM's Middleware for Retail Solutions, which includes WebSphere, DB2, Tivoli, Lotus, and Rational components plus a J2EE-compliant application server, an assured message delivery component, a standard relational database management system, and management and monitoring agents.

This infrastructure, coupled with IBM enterprise solutions, can provide retailers with real-time access to customer, product, and sales information, standard access to applications from anywhere, and improved deployment and management of resources. "It puts critical information where it is needed most – in the store – so that employees can respond more efficiently to customers' and suppliers' needs," said Klym. "Getting the customer information on products, pricing, and promotions right then and there encourages purchasing and increases loyalty."

The IBM WebSphere Remote Server also enables the use of ISV solutions in stores, including:

- Mobile shopping devices that deliver personalized information to customers
- RFID technology to speed the checkout process and locate

“Retailers must provide a much more dynamic environment for today’s customers who expect to have the information and products they want at their fingertips”

merchandise in the store and elsewhere in the supply chain
- Digital merchandising that provides dynamic product information and comparisons on demand
- Up-to-date inventory information to help prevent erroneous reordering of merchandise

“The IBM WebSphere Remote Server delivers a standards-based platform for integrating current store systems, such as existing IBM point-of-sale (POS) applications, new POS solutions provided by IBM Business Partners, and the infrastructure to deploy new technologies,” said Traflet. “It links the Web, the store, and the retail enterprise together to help the retailer determine whether it has the right goods, in the right location, at the right price with the level of service and the unique shopping experience consumers demand,” added Klym.

Many retailers begin implementation by updating their POS systems. Since the server is based on Java 2 Platform, Enterprise Edition (J2EE), there is flexibility in choice of operating system. “Currently, most retail companies’ enterprise-level operations have all the data and the store doesn't,” said Traflet. “They can now update to link all operations, so they can act on data and add solutions. It’s like they are extending electricity throughout the house and will then plug in a television, lights, computer, etc.”

Typical users of the IBM WebSphere Remote Server are retailers who are often global in size with hundreds, or even thousands, of remote retail stores. Because the solution can run unattended in a store, it can also be used by innovative service message block (SMB) retail clients who are positioning themselves for growth in the industry.

The second package in the Store Integration Framework is the IBM WebSphere Systems Management Accelerators for Retail. This package includes sample files and examples for remotely deploying, managing, and monitoring applications that will run on the remote retail servers installed in the stores. Before its planned availability in the fourth quarter of 2004, IBM is making the solution available to ISVs and IBM Business Partners.

There is also a variety of extensions for specific solution implementations available separately through services engagements. They include the RFID extension for managing RFID readers and RFID tags; the J2EE extension, which provides for management of business objects and application interaction within the store; the retail extension package, which delivers WebSphere Business Integration adapters for TLOG and bulk data transfer; and the digital media extension, which supplies tools to provide dynamic digital merchandising and digital surveillance solutions.

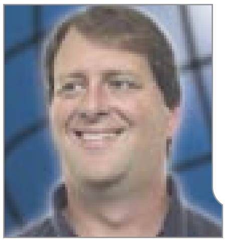
In addition, if a customer runs an IBM 4690 application such as General Sales Application, Supermarket Application, or IBM SurePOS Application Client Environment, the 4690 extension product delivers IBM Store Integrator and Data Integration Facility products for extending the POS environment. These solutions help integrate IBM POS applications with a standard API interface for other POS providers to access data and business logic. They can enable transaction integration at the fuel pump, kiosks, mobile POS, and consumer shopping solutions.

This year, IBM has worked with more than 20 retail customers who are piloting the retail solutions. Klym described these customers as “innovators – visionary customers who want to make the most of the Web and want to deliver the store of the future.” Those customers include entities in Europe and Asia, as well as the Americas. “According to our customers and the key analysts, investments in the retail store will increase significantly over the next 10 years. The middleware we are delivering will enable our customers to do more, faster,” said Traflet. “Many customers have had this vision of middleware in the store, but because they were going to have to integrate all of the various products together (application server, messaging, database, systems management) and then integrate it into their enterprise, it was going to be a challenge. Often their architectural vision matches the architecture we are delivering with this Store Integration Framework.”

Speed up and simplify

Rapidly Deploying Applications to WAS

BY TIM FRANCIS



Tim Francis is a senior technical staff member and development manager of the WebSphere Tools team in the IBM Canada Toronto Lab. He has been one of the lead architects for WebSphere Studio since it was first conceived and has played a key role in its evolution and development. Tim is a senior member of the WebSphere Architecture Board and a core member of the Rational Desktop Tools Development Council. He is coauthor of *Professional IBM WebSphere 5.0 Application Server*, Wrox Press.
francis@ca.ibm.com

Developing J2EE applications can be time-consuming and error prone, and deploying them for execution on an application server only adds to the complexity. Even a simple "Hello World" Enterprise JavaBean (EJB) requires multiple files to be created and maintained in the correct locations. To the developer, this can mean frustrating complexity during the iterative development and debugging of the application. To the business, this complexity can cause delays in mission-critical or customer-facing applications.

For IBM WebSphere users, many of these problems have been addressed in the latest release of WebSphere Application Server, Rational Web Developer for WebSphere Software (RWD) and Rational Application Developer for WebSphere Software (RAD), announced in October. These products contain a new feature called WebSphere Rapid Deploy, which accelerates application deployment and simplifies system testing on WebSphere Application Server version 6. If you're unfamiliar with RWD and RAD, they are the newest versions of WebSphere Studio Site Developer and WebSphere Studio Application Developer, respectively. An impressive set of new features helps enhance the quality of your code and simplifies the development of J2EE and portal applications. These products are now part of the IBM Rational division that took on their development staff and mission over a year ago. Despite the name change, RWD and RAD continue to be optimized for deploying applications to WebSphere, as shown by the addition of WebSphere Rapid Deploy.

WebSphere Studio Application Developer (WSAD) provides advanced wizards and editors to help you create and manage the artifacts within the development environment. The complexity of the J2EE infrastructure is still evident. Even a simple stateless session bean requires a bean implementation class, a home and remote interface, and a deployment descriptor, all of which must be kept in sync during development. WSAD also contains an integrated test environment that can automatically deploy applications to the embedded WebSphere Application Server with a single menu item click. This test environment has a number of limitations though, including the inability to deploy your application to a "staging server" that is shared among a number of developers. These issues are all addressed in the new products via the "WebSphere Rapid Deploy" (WRD) feature.

WRD introduces a new annotation-based programming model that greatly simplifies development. The annotations take the form of javadoc style comments in the code, allowing complex component-based applications such as EJBs to be represented within the development environment using only a single Java class. This class contains annotations that indicate the actual J2EE artifact type that is desired (e.g., a stateless session bean) and then generates the infrastructure (e.g., the remote and home interface). The resulting application is fully-formed and J2EE-compliant, so you only need to maintain one simple class and all of the complex aspects become derived deployment artifacts.

Similar capability is available in an open source project known as "XDoclet," and the annotations it supports can be processed by WRD. However, WRD is tightly coupled into the RAD development environment; for example, it provides code assist when you hit control-space, and changes are processed incrementally when a change is made to the Java source file. It is possible to annotate a class, method, or Java field, and the resulting code is simple to read. Figure 1 is a screen shot showing all that is required to define an EJB – everything else is now generated and maintained as a derived deployment file. Figure 1 also shows the code assist available within the annotation.

No explicit action is required to process the annotation, as the generation of the code occurs automatically in a background task. Changes can also be made to the annotated bean at any time and the deployment artifacts are kept in sync – and the output of the entire process is a normal EJB project that is completely accessible using the rest of the development tools. For example, Figure 2 shows the same EJB displayed in the deployment descriptor editor.

You can see a green "@" symbol on the file icon in the navigator view on the left indicating that this bean came from an annotated class. Complete UML (unified modeling language) visualization support is also available for annotated beans, as shown in Figure 3. As above, the only file that actually needs to be maintained by the developer is the original annotated JavaBean; all other files are maintained automatically.

Once development of the bean is complete, the application needs to be deployed to a WebSphere Application Server for testing – and WRD plays a substantial role here as well. The test environment in WSAD generates a complete server configuration and then starts a new instance of the server process. This approach provides a very simple user experience but is somewhat slow because the entire application server process has to be started. In the past, generation of the complete server configuration meant that the resulting server process was limited to the execution of the applications in the workspace. There was no way to install a new application into a running server process or to share a server instance with another user – a requirement that is quite common for "staging" or integration test servers.

WRD solves these problems by integrating the normal WebSphere "publish" mechanisms into the development environment. This means that when the application is executed a new server process does not need to be started; on the contrary, the application is published to and installed into an existing application server. The developer is no longer required to wait for the application server process to start, which in turn means the test experience can be dramatically faster. This change also allows the use of a shared server from within the test environment: as the application is being correctly installed into the server, the development environment no longer owns the complete server configuration, and interaction with shared

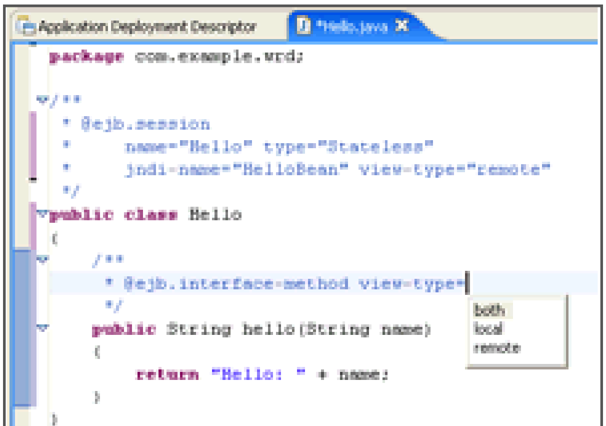


FIG 1: DEFINE AN EJB

WebSphere Journal Advertiser Index...

COMPANY	URL	PHONE	PG
EV1 SERVERS	WWW.EV1SERVERS.NET	800-504-SURF	13
H&W COMPUTER SYSTEMS	WWW.HWCS.COM/WSJ19.ASP	800-338-6692	21
IBM	WWW.IBM.COM/MIDDLEWARE/AGILITY	800-IBM-4YOU	2
IBM	WWW.IBM.COM/MIDDLEWARE/BUILD	800-IBM-4YOU	25
INFERDATA	WWW.INFERDATA.COM/WDJ	888-211-3421	3
ITEMFIELD	WWW.ITEMFIELD.COM/WBI	800-838-1055	49
ITKO	WWW.ITKO.COM	877-BUY-ITKO	31
MOTIVE	WWW.MOTIVE.COM/WITHIN1	512-531-2527	52
NASTEL TECHNOLOGIES	WWW.NASTEL.COM/UNCOMPLICATE	800-580-2344	51
PROLIFICS	WWW.PROLIFICS.COM	800-675-5419	17
TANGOSOL	WWW.TANGOSOL.COM	617-623-5782	29
VISUAL PARADIGM	WWW.VISUAL-PARADIGM.COM	+852 2744 8722	27
WEB SERVICES EDGE 2005	WWW.SYS-CON.COM/EDGE	201-802-3045	41
WILY TECHNOLOGY	WWW.WILYTECH.COM	888-GET-WILY	5

WebSphere Journal Coming Next Month...

PART II
The Support of Composite Applications in WebSphere Portal

AN INTERVIEW WITH BUELL DUNCAN
GM, Developer Relations IBM Software Group

AUTOMATED DEPLOYMENT OF ENTERPRISE APPLICATION (EAR) UPDATES
Key concepts and operational considerations

TELECOMMUNICATIONS COMPANIES FACE MANY CHALLENGES
IBM's Middleware solutions save time and money

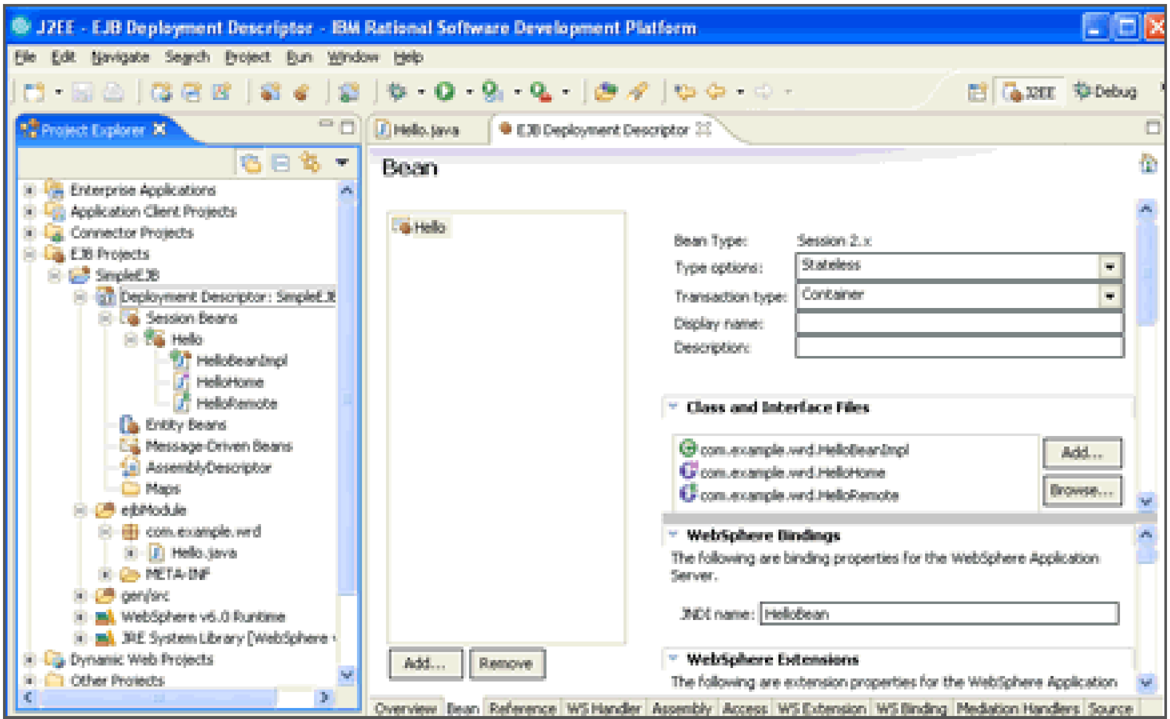


FIG 2: EJB DISPLAYED IN THE DEPLOYMENT DESCRIPTOR

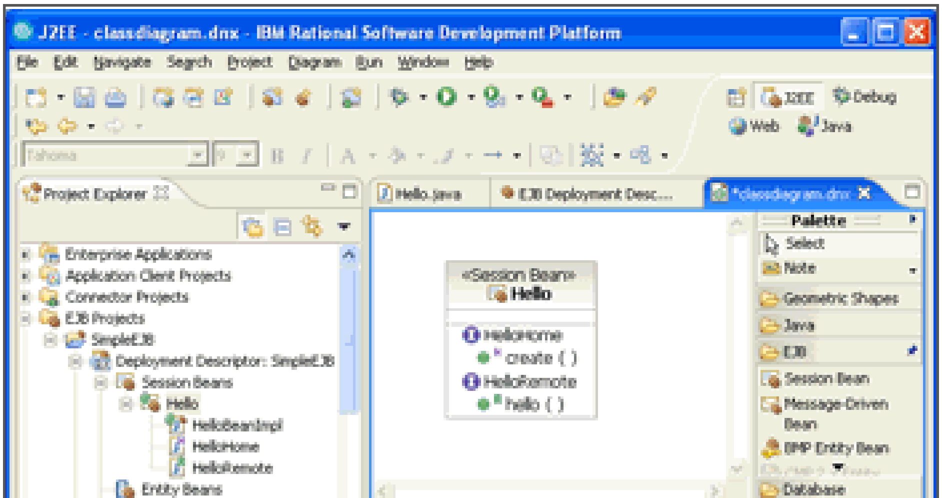


FIG 3: COMPLETE UML VISUALIZATION SUPPORT

or “staging” servers running on any platform is now possible.

Another new feature of WebSphere Application Server v6 is called “fine grained application update.” This capability is exploited by WRD to reduce the number of files published when modifying an application. In turn, the number of times the server needs to be restarted can be minimized. The WRD infrastructure is aware of the modifications being made to an application and automatically makes the minimum changes necessary on the server to reflect those modifications correctly. These changes are all contained within the same simple mechanism for invoking a server that has always been part of WSAD – you can select any artifact within the workspace and invoke a “run on server” action on that from the pop-up menu (see Figure 4).

Another aspect of WRD affects how server configuration changes are edited and maintained for a particular application. WSAD maintains the entire server configuration within the workspace and provides an editor for making changes to that configuration. This approach is very simple to use, but can cause a problem with the entire test experience: there is no mechanism to ensure that the server configuration used during the unit test is replicated correctly when the application is deployed to another test or production of WebSphere Application Server.

The WRD approach to this problem exploits another new function in WebSphere Application Server v6 known as the “enhanced EAR” support. This allows key aspects of the server configuration to be included within the application itself as data in the EAR file. To support this, the server configuration editors that were part of WSAD have been replaced in the new Rational development tools by a “deployment” page in the EAR application editor.

This page allows you to specify common settings such as data source definitions, the classloader mode, and virtual host information. These settings are then carried with the application and applied to any WebSphere Application Server v6 instance into which the application is installed. This is optional behavior (it can be disabled if desired to prevent inadvertent changes being made to a production server), but it helps you ensure that a consistent configuration is used for your application regardless of where it is installed. In turn, this means the good results you obtained while unit testing your application are now much easier to reproduce when the application is installed on other servers.

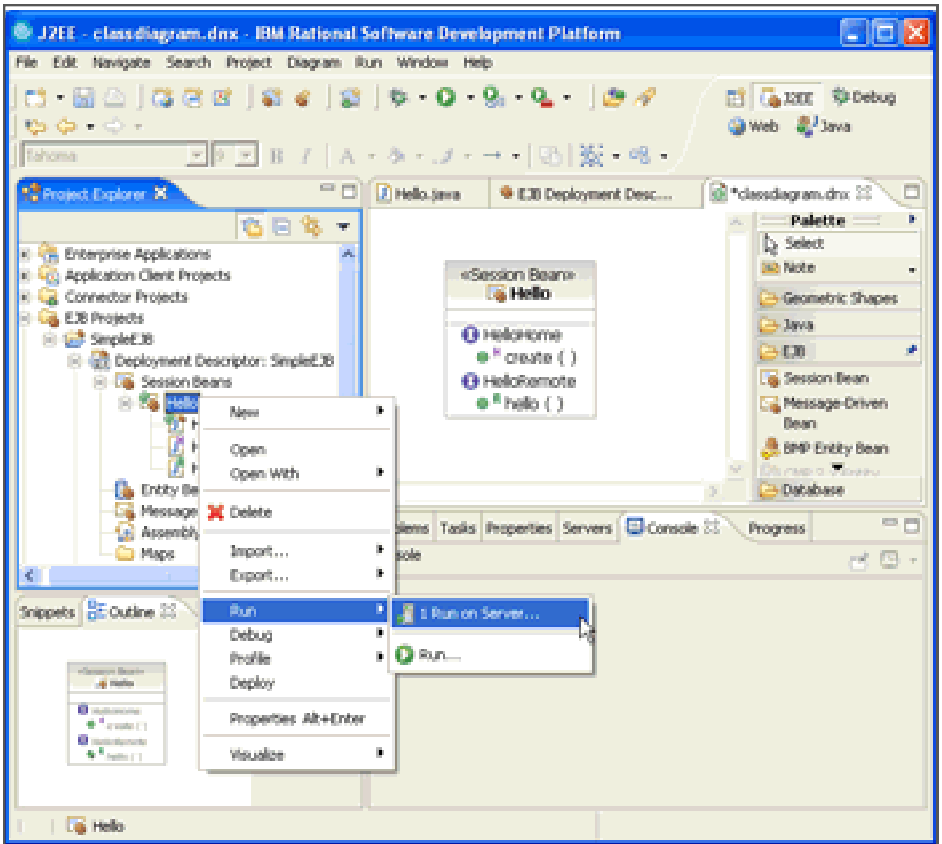


FIG 4: RUNNING AN ANNOTATED EJB

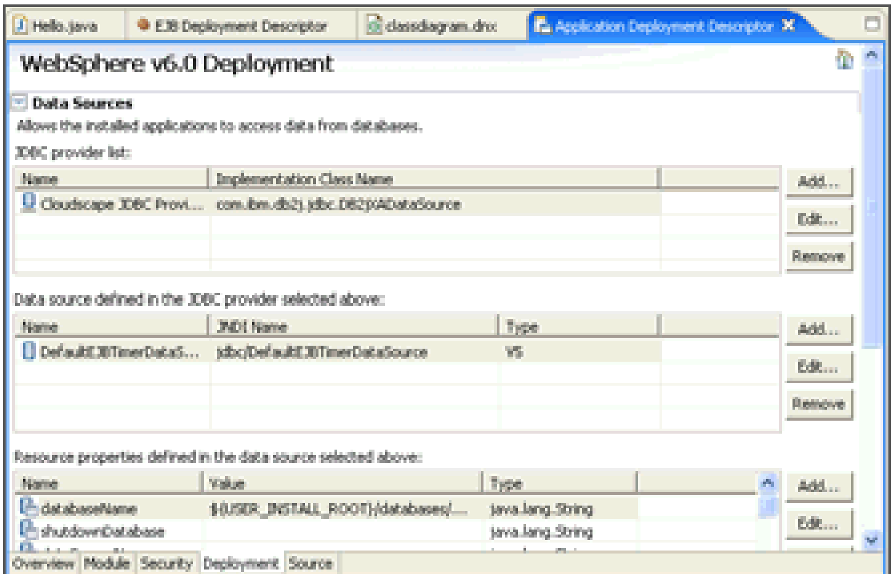


FIG 5: DEPLOYING INFORMATION IN THE ENHANCED EAR EDITOR

Summary

The WebSphere Rapid Deploy capability that is exploited by Rational Web Developer and Rational Application Developer improves the experience of developing and deploying applications for WebSphere Application Server v6. The use of annotated beans allows you to reduce the number of artifacts that need be produced and maintained, simplifying the process and allowing for a more efficient development experience. The automated installation of applications into WebSphere makes the test and deployment process simple. System testing applications across multiple servers is more capable and consistent than ever before. These innovations highlight the integration between WebSphere and Rational products, and the efficient developer experience that is available when they are used in combination.

A vision of Web services for remote portals

Plug-and-Play Remote Portlets

BY KULVIR SINGH BHOGAL & ANDREW SWEET

Kulvir Singh Bhogal works as an IBM Software Services for WebSphere consultant, devising and implementing WebSphere-centric solutions at customer sites across the nation. He has over 70 patents pending in a myriad of technology areas.
kbhogal@us.ibm.com

Portlets constitute interactive Web application components whose presentation markup is aggregated and displayed by a portal server like WebSphere Portal. In a previous WebSphere Journal article, we introduced you to the Java Specification Request for the portlet specification (JSR 168), which lays out the plans for a standard for portlets that will enable them to be deployed to any JSR 168 compliant portal.

In this article, we will further our study of portlet portability. In particular, we will focus our discussion on the Web services for remote portals (WSRP) standard. The vision of WSRP is to allow portlets to be exposed as Web services. The resulting Web service will be user-facing and interactive. Unlike traditional data-oriented Web services, a WSRP-compliant Web service will carry within its payload the presentation logic necessary to display and interact with the portlet; consequently, portlets can be readily plugged into remote portals.

How Things Are Done Without WSRP

Before WSRP, there was simply no concept of being able to access a remote portlet as if it was a local portlet. Portlets had to be physically installed at a portal. As a result of this architectural limitation, enterprises wanting to bring remote content into their own portal typically faced a significant programming effort. They ended up having to run a portlet locally in their own portal infrastructure, which often required a painful development and integration effort involving the redevelopment of a presentation layer.

Let's say, for example, that we are an enterprise that wants our employees to be able to access an employee 401K retirement portlet. The 401K services are provided by a third-party financial institution. Without WSRP we'd end up having to install the 401K portlet into our own portal. The installation effort would probably involve having our portlet access data sources housed in the financial institution. In short, we would have a potentially daunting application development effort on our hands, especially if we are shorthanded in the area of portlet developers. In some cases, "we" might not even have portlet developers on our staff, e.g., if we are a small nontech firm. Then we end up having to worry about security issues and cringe at the reality that the financial institution wants to provide periodical updates to the portlet to better serve its constituents (a.k.a. your employees). Wouldn't life be much easier if the financial institution could provide us with the portlet remotely?

"Wouldn't life be much easier if the financial institution could provide us with the portlet remotely?"

Plug-and-Play Remote Portlets

WSRP is being commandeered by the Organization for the Advancement of Structured Information Standards, OASIS (www.oasis-open.org/). The vision of OASIS is for a portal administrator to be able to choose to incorporate pluggable, remote portlets wrapped as Web services into their own portal. The process of incorporating remote portlets will require no programming effort; OASIS envisions a discovery and installation process involving "a few clicks." The portlets will be user-facing and interactive. The main offering – the portlets – will not be housed locally, *they are remote* (residing on different parts of an enterprise's intranet or the Internet). Because they are Web services, they will be loosely coupled to the consuming portal server. Unlike traditional Web services though, Web services for remote portals will carry both application *and* presentation logic that can be displayed by a consuming portal. To the end user, remote portlets will look like and interact with the user just as local portlets would.

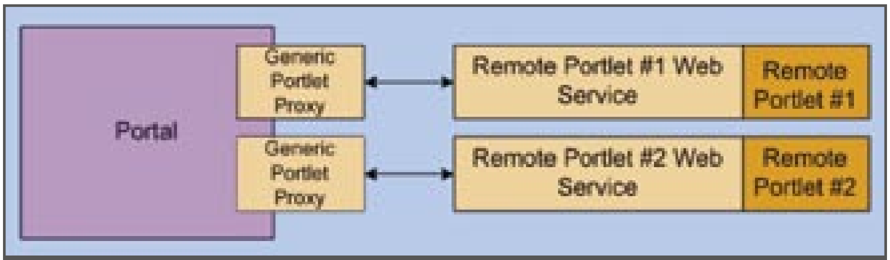


FIG 1: PORTAL USING GENERIC PROXIES TO INVOKE AND CONSUME REMOTE PORTLETS

Consuming WSRP Services

WSRP-compliant portals will not have to be trained in how to consume a particular remote Web service. WSRP-compliant portals will use a single generic adapter through which they can integrate with any WSRP-compliant portlet. The generic adapter is actually portlet proxy code that knows how to consume a remote portal Web service as if it were a local portlet (See Figure 1).

WSRP taps into the advancements made by the Web services community. It is a presentation layer built on the Web services stack. The interfaces of WSRP are laid out in WSDL (Web Services Description Language). Portlets compliant with the WSRP standard can publish metadata to describe themselves to portlet registries from which they can be discovered and integrated into remote portals.

User-Facing Web Services versus Data-Oriented Web Services

Those savvy with Web services might be wondering what the difference is between the Web services they are used to dealing with and presentation-oriented, user-facing, interactive Web services offered by WSRP. Data-oriented Web services typically return data objects encoded in XML as responses to SOAP (Simple Object Access Protocol) requests. Though our encoding is SOAP, each data-oriented service might be quite different (expecting different request parameters and returning a different response) from the semantics of the interaction with the Web service being described in WSDL. Due to the variety of Web services and associated descriptions, data-oriented Web services require service-specific proxies.

Juxtaposing user-facing Web services in our payload that we return to our Web services client, we include markup fragments that can be aggregated by a consumer.

Backing the Standard

As with practically any standard, the standard is rendered worthless unless it has buy-in from the major players of the industry. The WSRP 1.0 specification is a collaborative effort of nearly 25 OASIS member companies and has been approved by key industry front-runners like:

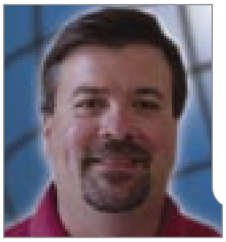
- IBM
- Apache Foundation
- BEA
- Microsoft
- Oracle
- Sun
- Plumtree
- Vignette

IBM in particular has played a major role in promoting the WSRP standard. Big Blue initiated the Apache WSRP4J project to catalyze adoption of the WSRP standard. At the time of this article's writing, the WSRP4J project was still an "incubated" subproject under the Apache Software Foundation's Web services project. Nevertheless, WSRP's vision is gaining the attention of huge audiences who see the user-facing pluggable Web services offerings of WSRP to mean cost savings.

IBM WebSphere Portal Version 5.0.2 currently provides support for WSRP through a WebSphere portal technology preview in the form of IBM WebSphere Portal Version 5.0.2 Cumulative Fix 2 (5.0.2.2). With the cumulative fix, you can enable WSRP in WebSphere Portal and provide your portlets as WSRP services, and also integrate other WSRP-enabled services into your portal as remote portlets. For more information about WebSphere Portal's WSRP support, see "Using WSRP Services with IBM WebSphere Portal Version 5.0.2 Cumulative Fix 2", http://publib.boulder.ibm.com/pvc/wp/5022/ent/en/standards/wsrp.html

JSR 168 Relationship with WSRP

You may begin to wonder what the relationship is between JSR 168 and WSRP. The two standards are not competing. In fact, in some cases, they might be considered complementary. JSR 168 portlets can be deployed to any JSR 168 portal. These portlets are local to the portal. Portlets exposed as Web services (using WSRP) run remote to the portal. They may or may not be JSR 168 compliant portlets. JSR 168 portlets might be exposed as pluggable Web services for other portals to consume. As discussed earlier, the consuming portal interacts with the remote portal service through a proxy.



Andrew Sweet serves as a senior solution architect for IBM's WebSphere Services and is a frequent speaker and mentor around enterprise best practices and standards-based technologies such as J2EE and Web services.
asweet@us.ibm.com

"WSRP's vision is gaining the attention of huge audiences who see the user-facing pluggable Web services offerings of WSRP to mean cost savings"

Conclusion

The WSRP vision will allow for enterprises to consume and publish portlets as Web services. These portlets will carry with them the user experience intended by the creator of the portlet. WSRP aims to allow enterprises to save time and costs by being able to leverage portlets created by others. Since we are still talking Web services, we get the added benefit of being able to code our remote portlets in any programming language, given that we adhere to the interface laws laid down by the WSDL interface description. It is the job of a remote portlet Web service to deliver HTML, WML, VoiceXML, or whatever content an end portal client might expect.

In reality, what we will probably see in the future are WSRP-compliant portals displaying a heterogeneous mix of local and remote portlets to better serve end users. As mentioned in this article, WSRP adds a presentation layer to the Web services stack. As the Web services realm is itself evolving (with standards emerging in key areas such as security and transaction support), it will be interesting to see how or even if the WSRP standard will need to change. Oasis has already begun working to define a new WSRP 2.0 standard.

The JSR 168 specification addressed in the previous article and the WSRP standard have the support of key players in the portal arena including IBM. IBM WebSphere Portal 5 provides support for both WSRP and JSR168. The ability of these technologies to lower costs for companies leveraging portals makes the JSR 168 specification and the WSRP standard key architectural and design considerations for your enterprise.

HOUSE events

HOU

eve

USE
ents

HOU
eve

USE
ents

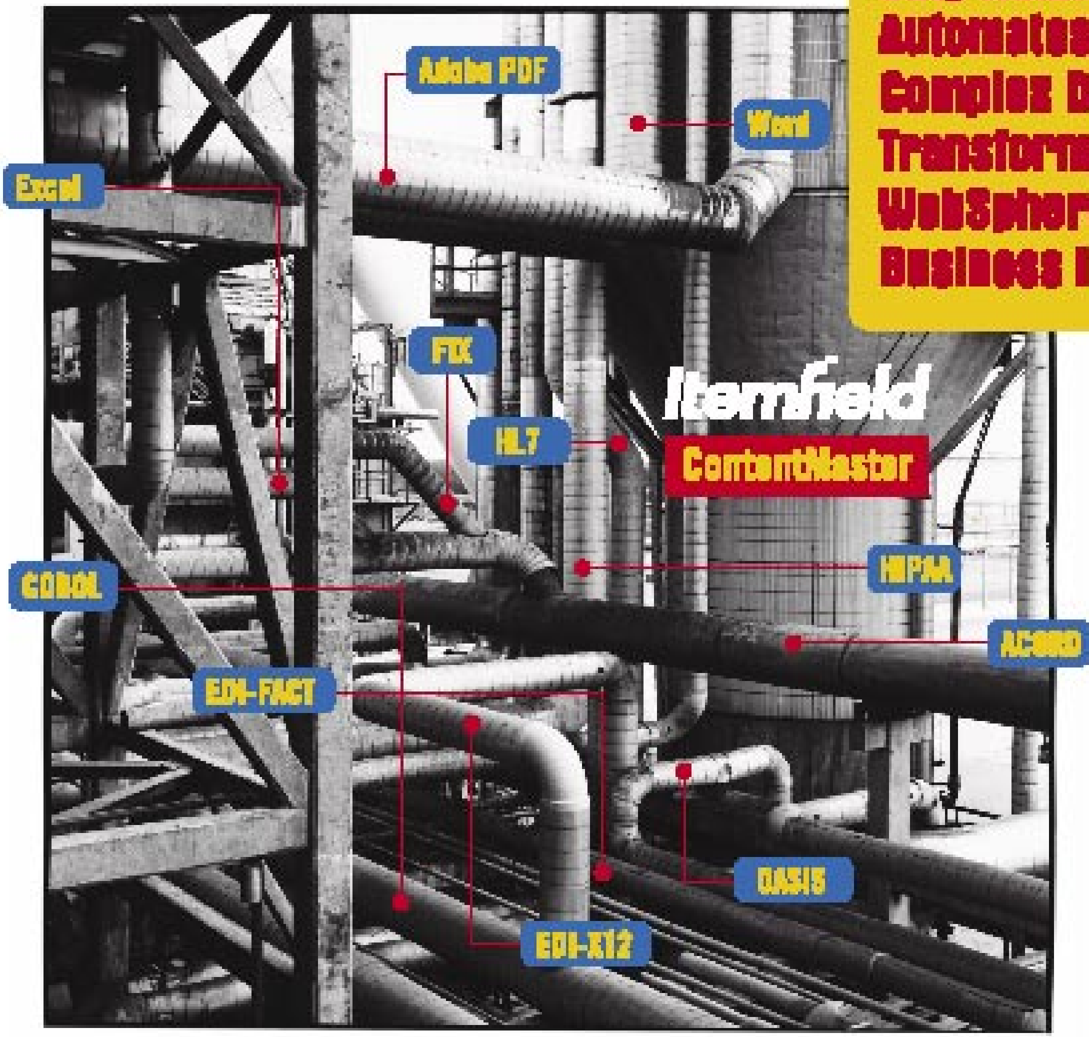
HOU

eve

USE
ents

HOUSE events

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

Striking Gold vs Toxic Liability

BY JON STANSBERRY

With the rush to take advantage of the benefits brought about by upgrading to WebSphere 5.1, corporate America has created an abundance of surplus hardware. When deciding to take advantage of the latest WebSphere upgrade, it is often necessary to refresh your hardware. One thing companies usually let slip past the planning stage is figuring out how to retire excess or obsolete computer hardware. Not only do corporations need expert advice in the upgrading of software applications and hardware selection, they also need a clear indication of what to do with the obsolete or excess hardware.

From privacy issues to the EPA, when a computer leaves your company it can become a serious risk for your business. There are several important factors to consider when planning the disposal of computer hardware. Environmental issues must be among the most important, with data protection and return on investment close behind. Steering through the myriad of options for hardware recycling can be difficult to say the least.

Because of the overwhelming cry to dispose of hardware in an environmentally-friendly way, corporations cannot afford to take any chances. According to the EPA, some computers that are donated or sold to scrap dealers and recyclers may end up being disposed of improperly. To minimize potential liability, businesses must use due diligence to ensure that their computers are disposed of properly.

The Electronic Recycling Act of 2003 is now law in some states (e.g., California) making it a felony to dispose of covered electronic devices in undesignated areas. An estimated 63 million personal desktop computers in America will become obsolete this year. A computer monitor or television set (better known as cathode ray tube or CRT) contains roughly four pounds of lead. These items also contain highly toxic cadmium and mercury, both of which leach into the ground water table. These are just a few of the environmental issues affected by the onslaught of outmoded computer hardware disposal, making it more important than ever for today's responsible companies to have solid recycling plans.

In today's information age, it is ever more important to ensure the protection of sensitive corporate data. Eventually the time comes when each of us is faced with having to return or trade our leased computers, or resell or dispose of our own computers. Many of these systems have hard drives containing data that corporations do not want seen by others. A simple delete, as executed by all operating systems, makes files only nonaccessible on the OS and application program level (the file names are gone). However the fact is the complete contents of all those files are still there, able to be completely recovered as long as they are not overwritten by new files. To be fully protected, companies must ensure that they or their recycling partner are using a disk-wiping program that completely overwrites and destroys all existing data on a hard disk. Every physical byte of the disk must be overwritten. Some corporations even require that data destruction be performed in a manner that conforms to U.S. Department of Defense requirements.

Most companies have some method of selling off or disposing of their excess computer hardware. What we're learning is that it usually costs a company much more (as high as $400 per unit) to manage its own asset disposal/resale program. An IT asset remarketing partner with recycling capabilities can save money and help maximize return on investment.

There are hundreds of companies whose core business is retiring and recycling excess computer equipment, but finding the right one that meets your business's specific requirements can sometimes be quite difficult.

With so many choices available when seeking a partner, it is important you select one that has the marketing and technical capabilities to deliver a flexible solution. The company you select should have a wide range of asset recovery services that enables you to maximize the liquidation value of obsolete or nonproduction current life-cycle IT equipment, coupled with a responsible approach to removing your data and disposing of unusable computer hardware. One way to be sure you are getting the most value for your unnecessary hardware is to talk in depth with the asset recovery company. Are they using creative means of maximizing the return? Do they have far-reaching and established channels for redistribution? Can they handle multiple manufacturers and platforms? Do they know what they are doing when it comes to things like data erasure, recycling, and reconfiguration and remarketing? These are only a few of the questions you should ask to help you decide if you are dealing with the right company. If they cannot readily answer your concerns, you may want to find another partner.

Some creative remarketers of used IT hardware are issuing credit instead of paying for old computers, if requested. This way, the money does not go back to the company immediately, but can be accessed readily in the form of new or reconditioned hardware, effectively increasing the fiscal year budget for IT. Sometimes it makes more sense to donate equipment to a nonprofit organization. In some cases companies can actually get more in write-offs than they can in resale value. However not many nonprofits know what to do with an IBM AS/400. Again, a creative remarketer will help you find a way to donate the equipment to the nonprofit organization, which can in turn trade it in to the remarketer for equipment they really need.

Whether your need is to liquidate surplus equipment or to remove proprietary data from your servers or storage, asset recovery and liquidation services eliminate the hassle of disposal and allow you to focus on more important business functions.

With a 14-year background in the IBM mid-range marketplace, Jon Stansberry has joined the HiTech Assets team to help expand their capabilities into a one-stop shop for IT Asset disposition. HiTech now deals in IBM, HP, Dell, Sun, Cisco, PC/laptop and recycling of other IT assets. jstans@hitech-assets.com