APP DEVELOPMENT • APP MANAGEMENT • APP SERVER

WebSphere® JOURNAL

www.WebSphere.SYS-CON.com

The World's Leading Independent WebSphere Resource

OCTOBER 2005 VOLUME 4 ISSUE 10

PRESORTED
STANDARD
US POSTAGE
PAID
ST. CROIX PRESS

NEW!

CodePro AnalytiX™

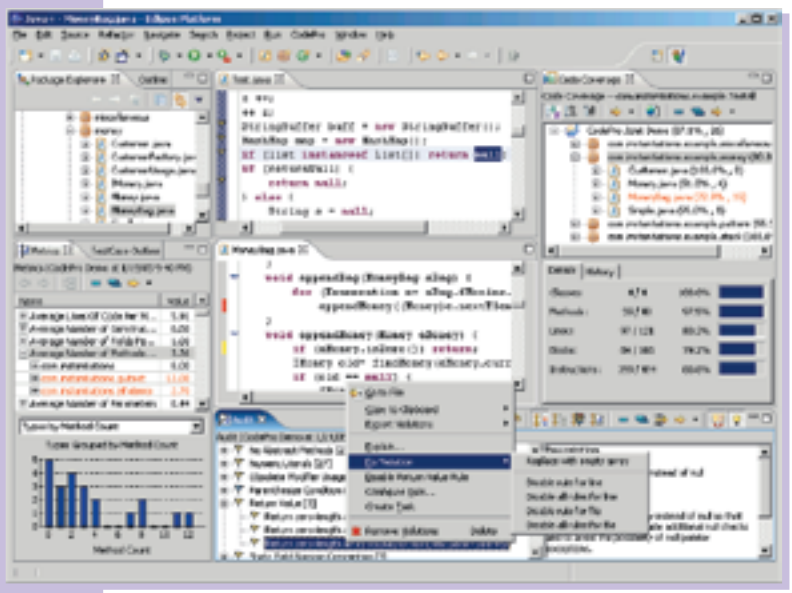
Continuous Collaborative Code Analysis for Rational®, WebSphere® & Eclipse

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

Sound like a fantasy?

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

Your money back.



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

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

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

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

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

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

Put our experience to work for your team.

www.instantiations.com

1-800-808-3737

Key Product Capabilities

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

Risk Free Trial Copy

www.instantiations.com/analytix

eclipse READY. IBM Business Partner eclipse FOUNDATION MEMBER

instantiations

Movin' On Up

time to migrate to the next generation of WebSphere

Ask the Experts

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

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

From J2EE App Servers to WebSphere

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

From WebSphere v3.5/ v4 to v5

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

Taking You Step by Step

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

Prolifics

Winner of 2003 IBM BUSINESS PARTNER LEADERSHIP AWARD

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

Prolifics®

Is IBM Standing Still?

BY ROGER STRUKHOFF

Random encounters can provide unexpected illumination. If you are the type of person who just starts talking to strangers anywhere anytime, as I am, the conversations that ensue can relate to your work a suprisingly high percentage of the time.

And so it went with a conversation I had with at a recent college football game in San Diego. The game itself wasn't very good, with the local lads getting a sound beating from the opening kickoff, so our conversation quickly turned from discussions of the size of today's offensive linemen and defining the modern-day intentional grounding rule to a sudden segue into the future of IBM, prompted by the question, "what do you do when you're not watching college football?"

It turns out that my seat mate was a sales manager for IBM Global Services, "the other IBM" as it's billed these days. "We've got to do something," he said. "Oracle announces a new acquisition almost every week, it seems, and (IBM is) just standing still."

But there's been a flurry of recent announcements by Blue, I countered, including news of a passably major acquisition, that of DataPower. "Not enough," he said. "(Oracle's) Larry Ellison has said the industry is in a major consolidation phase, and he's doing everything he can to make sure this is so. But IBM is just standing around, waiting for the market to come to it."

I found this viewpoint to be quite interesting, given that IBM has been among the more active companies in recent weeks, even though it hasn't tried to swallow a truly major company, despite endless rumors that it will do so some day. The trouble with these rumors, besides the obvious fact that they're rumors, is that IBM's history is not one of major acquisition.

Other than the Lotus exception (which probably proves the rule) many, many years ago, Blue is not known to go out and try to rip the heart out of the industry by buying a DEC, say, or a Compaq, a JD Edwards, a PeopleSoft. Instead, the company consistently tries to be all things to all people and win the game by sheer size and by industry-leading support.

Most recently, IBM has been promoting itself as a friend to the open-source community, through the Apache Foundation and with major announcements aimed at the Eclipse Foundation and at the storage community. The company's efforts are seen by many as nothing more than the big wolf trying to control thing by donning some open-source sheep's clothing, but nevertheless, IBM is opening itself to tremendous potential levels of input (and criticism) from all quarters with this activity, which is hardly the behavior of the secretive, closed culture associated with the company for many years.

IBM is by far the leader in web services, and has an advantage in the burgeoning SOA segment within that market. That is the topic of primary importance to this magazine and our readers. So a breakdown of how the company is faring in each of its business aspects is not totally relevant in these pages. But WebSphere developers need to remain confident that the company behind the technology will remain very strong in coming years.

Unbelievably, IBM looked a lot like a dying company a decade and a half ago. But it has rebuilt its image, remained strong across the board. It's good to know that at least one IBM exec is restless, that he wants the company to do more. This spirit is what keeps great companies great. But the company is hardly standing still. It seems as if the recent tide of announcements will keep rolling in over the months to come.

Now, if we could just do something about that lack of pass protection in San Diego...

ABOUT THE AUTHOR

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

roger@sys-con.com

WebSphere® JOURNAL

ADVISORY BOARD

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

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

WRITERS IN THIS ISSUE

Chris Lockhart, Serge Lucio, Tilak Mitra, Svetlana Petrova, Ade Rixon, Roger Struckhoff, Linfeng Yu

SUBSCRIPTIONS

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

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

EDITORIAL OFFICES

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

WORLDWIDE NEWSSTAND DISTRIBUTION
CURTIS CIRCULATION COMPANY, NEW MILFORD, NJ

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

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

SYS-CON MEDIA

Why is it that some Java guys are more relaxed than others?

These days, with everything from customer service to sales and distribution running on Java, keeping your enterprise applications available can be pretty stressful.

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

So put Wily to work. And keep performance problems from pushing you over the edge.

Get Wily.™

1 888 GET WILY | wilytech.com

wily technology

A Blueprint for Change

The IBM Rational Approach to SOA

BY JIM PALISTRANT

The need to respond to changing business demands with flexible IT solutions has led many businesses to service oriented architectures (SOAs). An SOA is an IT framework that combines individual business functions and processes, called services, to implement sophisticated business applications and processes. SOA is an approach to IT that considers business processes as reusable components or "services" that are independent of the applications and computing platforms on which they run.

ABOUT THE AUTHOR

Jim Palistrant is a Market Manager at IBM Rational that has been working on development and test tools for most of his IBM career. He started working on IBM's Java and Web development tools in 1995, helping to bring the WebSphere Studio tools to market. In 2004 he joined the Rational team and helped bring the Rational Developer products to market. In 2005 he became the Market Manager for Rational SOA Solutions. He is a frequent speaker at technical conferences and to customers around the world.

SOA solutions are designed as assemblies of services in which the description of the assembly is a managed, first-class aspect of the system, and hence amenable to analysis, change, and evolution. A system is then viewed as a choreographed set of service interactions, so enterprise solutions appear as federations of services connected via well-specified contracts that define their service interfaces. The ultimate goal of adopting an SOA is to achieve flexibility for the business and IT. To bridge the gap between business and IT processes, companies are taking a business-driven approach to SOA development. The following practical guide will address how businesses can develop an SOA to meet their specific IT and business challenges.

Business-Driven Development for SOA

Business-driven development for SOA addresses two of the key challenges facing business leaders – improving responsiveness and aligning IT with business goals. When business requirements are used to drive all downstream development activities, the gap between IT and business closes. In the context of service-oriented development, what results are exceptionally flexible solutions that can be rapidly adapted to meet changing business needs and priorities.

The IBM Rational Software Development Platform for SOA

The IBM Rational Software Development Platform is a complete, open, modular, and proven solution for developing software. It provides development teams with a powerful foundation for business-driven development of SOA solutions, and supports many other styles of development as well.

The IBM Rational Software Development Platform supports key capabilities for building SOA applications including: requirements and analysis; design; construction; software quality; process and portfolio management; and software configuration management.

The following overview provides a brief look at each of these capabilities, how they fit into an SOA approach, and the IBM Rational Software Development Platform tools that support them. This overview will set the stage for a more in-depth exploration of how all of these tools work together in the development lifecycle of an example SOA project.

The IBM SOA Programming Model

The IBM programming model for Service Oriented Architecture separates development and deployment activities into separate phases that can occur at different times and can be done using different skills, by different individuals. This yields a true separation of concerns, enabling the repurposing of software components.

Key elements of the programming model being driven by the IBM Rational Software Development Platform include:

- ***Service Data Objects (SDOs):*** Now in the standardization path at the Java Community Process. SDOs provide a simplified data access programming model for various resources and complement the core Web Services standards XML, WSDL, and SOAP.
- ***Business Process Execution Language for Web Services (BPEL4WS):*** A service orchestration and component scripting standard that supports workflow and business process integration.
- ***JavaServer Faces (JSF):*** A Java framework that speeds Web application development for developers who are not expert at J2EE.

- ***Customization of applications using external policies and rules:*** A series of emerging standards are in development for policy definition and enforcement, including Web Services Policy and OMG Business Semantics of Business Rules (BSBR).

Business Modeling and Requirements

Effective business-driven development starts by analyzing business requirements, needs, and existing processes. This analysis is then used to drive requirements, and all subsequent stages of development.

IBM WebSphere Business Modeler enables analysts to model, simulate, and analyze complex business processes quickly and effectively. Analysts use WebSphere Business Modeler to model "as-is" and "to-be" business processes, allocate resources, and perform "what-if" simulations to optimize and estimate business benefits. Later these models can be transformed into Unified Modeling Language (UML) and Business Process Execution Language (BPEL) to jump-start design and integration activities.

Service-Oriented Solution Design

An effective approach to designing a service-oriented application requires the modeling and definition of well-defined, well-documented interfaces for all major service components prior to the construction of the services themselves.

IBM Rational Software Architect is an integrated design and construction tool for creating service-oriented applications. By leveraging model-driven development with the UML and unifying all aspects of software application architecture, Rational Software Architect enables software architects to design flexible services architectures and automatically apply design patterns for SOA from analysis and design to implementation.

To assist with solution design in a services-oriented world, IBM Rational Software Architect can be specialized with domain-specific notations and languages relevant to defining, assembling, and managing services. General mechanisms are available to create customized notations

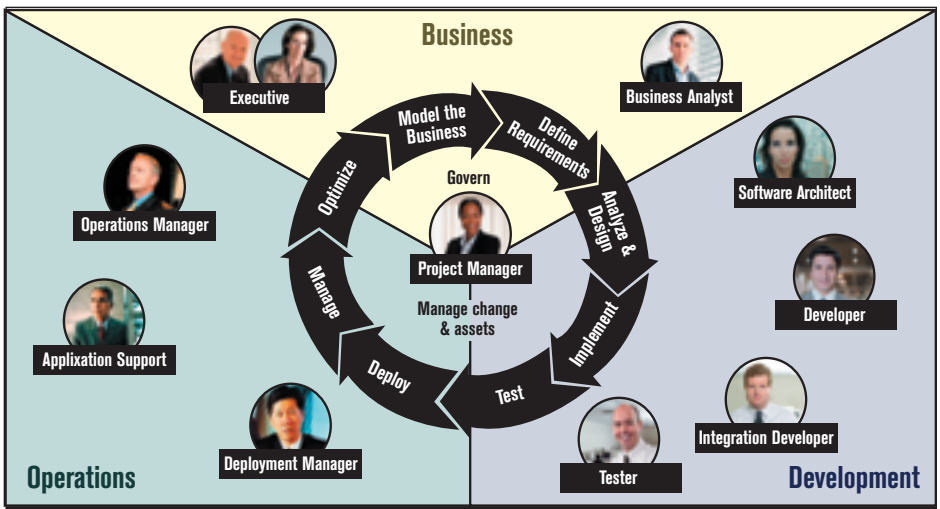


Figure 1: Business-Driven Development for SOA transforms business processes into composite applications that automate and integrate the business

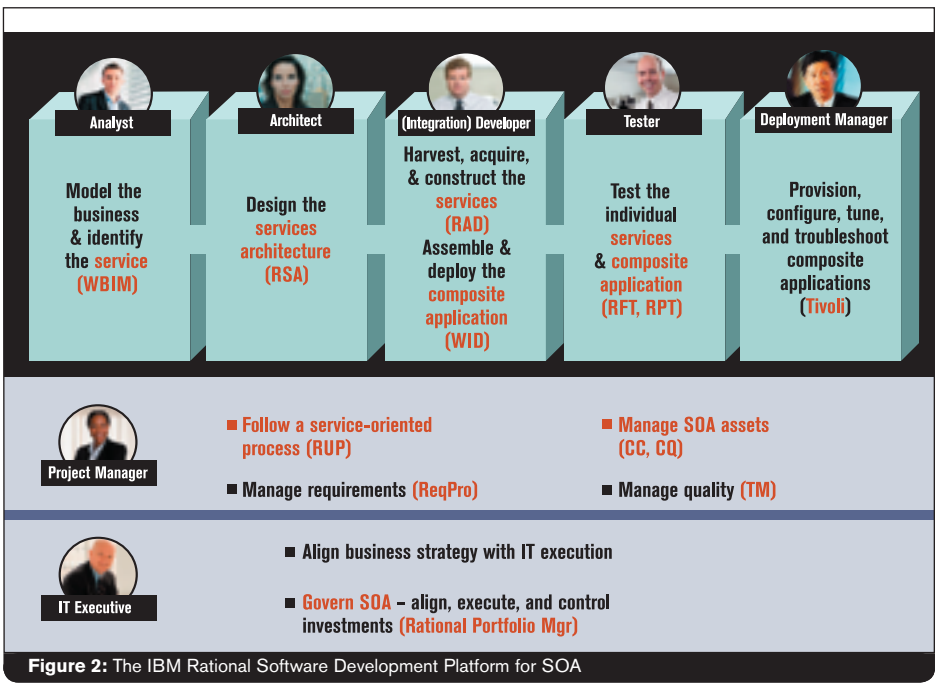


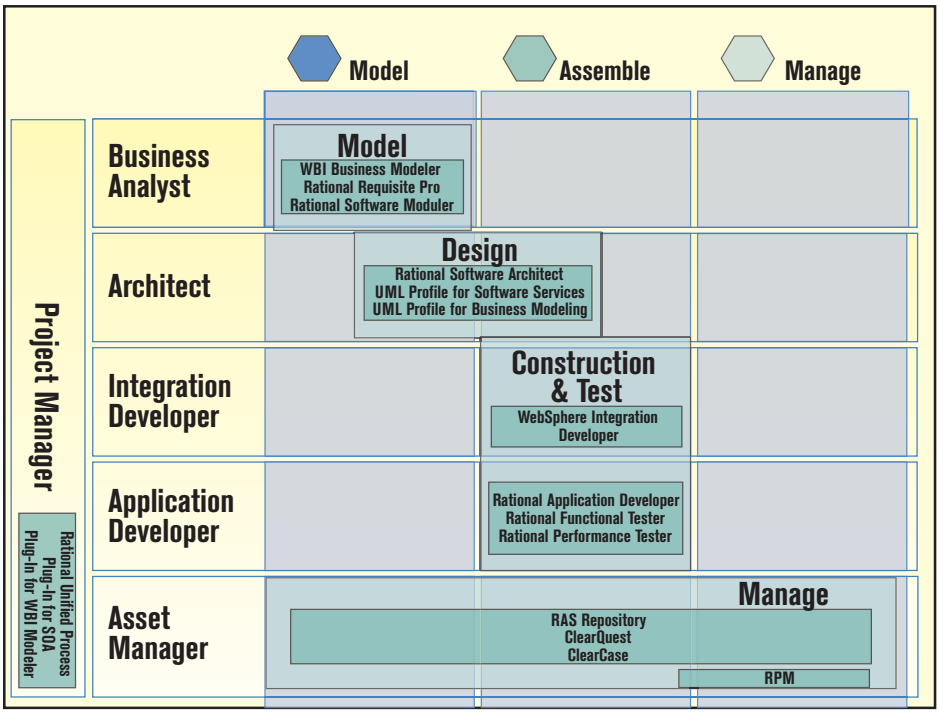
Figure 2: The IBM Rational Software Development Platform for SOA

using built-in product extension mechanisms. Examples of domain-specific languages created using these mechanisms include:

- ***The UML Profile for Software Services:*** Accelerates the transformation of business processes into Web Services by providing a common language for describing services, helping software architects to model, map, and partition services.
- ***The UML Profile for Business Modeling:*** Presents a language for capturing business models in UML and is supported by the Business Modeling Discipline in IBM Rational Unified Process.

Service-Oriented Solution Construction

Java, J2EE, and Web developers are often tasked with creating the core ele-



ments supporting SOA, typically Web Services. Manually creating Web Services – or even developing systems that consume existing Web Services – requires a substantial amount of tedious and error-prone work.

IBM Rational Application Developer is a comprehensive integrated development environment for service-oriented development that automates many of the tasks commonly performed in the construction and consumption of Web Services so developers can focus on writing the business logic code, and rely on Rational Application Developer to automate everything from WSDL file and code generation, to test proxy generation and WS-I conformance verification..

Rational Application Developer greatly simplifies the process of creating a Web Service top-down based on requirements specifications and UML models. Developers save time and reduce errors by using Rational Application Developer to automatically generate WSDL files that contain an XML schema to describe Web Services as well as skeleton JavaBeans or Enterprise Java Bean (EJB) files. Rational Application Developer also includes an XML Schema Definition (XSD) editor for specifying the format of messages. Finally, Rational Application Developer's J2EE Component Architecture tools can be used to repurpose existing CICS and IMS transactions as Web Services.

Rational Application Developer's Enterprise Generation Language support enables business-oriented and procedural developers, who may not know Java, to exploit Java to develop, test, and debug data-driven Web applications, Web Services and business logic using procedural programming constructs. EGL's comprehensive Web Services support handles most of the hard work in developing or consuming Web Services. Very little coding is required to consume existing Web Services or to enable existing assets and this leads to a shorter learning curve and high productivity for business-oriented developers.

To enable developers to build in quality early in the development cycle, Rational Application Developer includes automated code review tools to validate coding best practices, component test tools to automate the creation of test stubs, harnesses and input data based on WSDL file analysis and performance profiling tools information to detect and diagnose bottlenecks, deadlocks, and race conditions that enable developers to build quality early in the development cycle

Rational Application Developer also automates the process of discovering and consuming existing Web Services. Developers use a built-in UDDI Explorer tool to browse UDDI registries, find Web Services, and import them into a project. And, Web designers can invoke a Web Service by using the Page Designer feature to drag and drop Web Services on to a Web page.

Service Composition and Assembly

Integration developers create and deploy composite applications by refining business process models – created in WebSphere Business Modeler – into an executable set of choreographed services. WebSphere Integration Developer is an IDE used by integration developers to assemble composite applications that deploy to the IBM WebSphere Process Server. Using the visual BPEL editor, integration developers can view the business processes designed by business analysts. They then use WebSphere Integration Developer to choreograph the services, wire them together into a composite application, optionally test them with a built-in test environment, and deploy directly to a runtime environment.

WebSphere Integration Developer provides integration developers with the visual software development tools they need to specify, test, and deploy executable business processes that integrate Web Services, enterprise applications, human tasks, and other service components into effective SOA-based business solutions.

Process and Portfolio Management

IBM Rational Unified Process, or RUP, is a configurable software development process platform that delivers proven best practices and a configurable architecture. RUP is well suited to the needs of SOA development initiatives because it's founded on software engineering best practices,

Motive brings an enlightened approach to application management, building management intelligence into the application itself. Only Motive transcends the fragmented, disjointed reality of traditional application management, to deliver the visibility, insight and automation that support and development teams need to keep applications running smoothly.

This is no mere vision for the future. It's here now in application management products from Motive. Learn more about Motive's breakthrough approach in a new white paper about built-in management at www.motive.com/within1.

motive

www.motive.com

offers a configurable process framework, and is scaleable to support enterprise initiatives.

RUP can be applied to support the development of a complete SOA composite application as well as each individual service in the solution. RUP also provides a systematic approach for bridging the gap between business and IT.

IBM offers RUP Plug-ins – downloadable assets containing a variety of process components customized for specific tools, technologies, or domains – including plug-ins tailored to SOA development:

- ***The RUP Plug-In for SOA:*** Integrates support for service-oriented architecture and service-oriented solutions into the RUP framework, with SOA-specific concepts, guidelines, activities, artifacts, and tool mentors.
- ***The RUP Plug-In for WebSphere Business Modeler:*** Updates the Business Modeling discipline in RUP to leverage WebSphere Business Integration solutions and provide a unified approach for business modeling based on the essential capabilities of the IBM Rational Software Development Platform.

From an SOA perspective, applications are dynamic entities – combinations of services to meet a particular set of business requirements. One effect of taking this view is the need for enterprises to explicitly design, implement, and manage the set of services in the organization as a portfolio of available capabilities. Executives and project managers use IBM Rational Portfolio Manager to gain insight into the business benefits, costs, and risks of the portfolio of SOA services. With Rational Portfolio Manager, they can prioritize proposed, existing, and under-construction services, track service level financials, manage SOA project-team dependencies, forecast demand for service creation, and better understand the cost of SOA creation, operations, and maintenance.

Software Quality

Besides the developer testing capabilities in Rational Application Developer and WebSphere Integration Developer, the IBM Rational Software Development Platform includes tools to help testers continuously assess the quality of Web Services and ultimately SOA composite applications. Test plans and individual test cases can be linked to project requirements in Rational RequisitePro to ensure complete requirements-based test coverage.

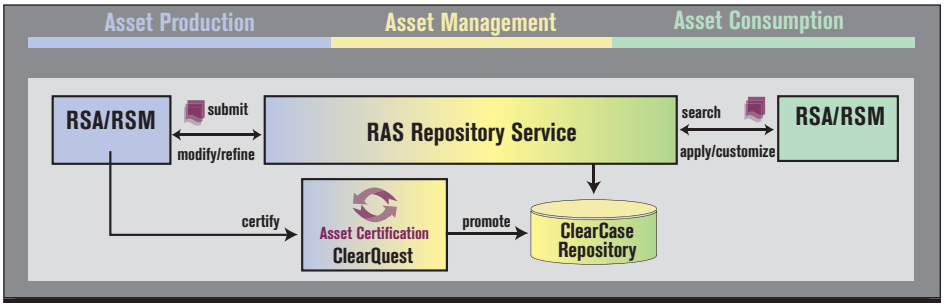


Figure 3: Rational ClearCase and Rational ClearQuest support RAS and asset lifecycle management

In general, test plans should be developed early during the life cycle. The best approach consists in formalizing detailed sequences of steps and verifications for every manual tasks defined in business processes. These different sequences are then "orchestrated" into test scenarios targeting multiple objectives. Once test plans are complete, a review with business analyst is critical to ensure proper coverage of business goals.

A number of tools can be leveraged to do such planning. Word processors tend to be the predominant ones. Despite their attractiveness, word processors don't foster reuse among multiple test scenarios. A lot of content is generally duplicated making the maintenance of the scripts tedious. Rational Manual Tester can be easily leveraged to overcome this challenge

IBM Rational Manual Tester: With Rational Manual Tester, each sequence of steps defined for a manual task is captured as a test script. These elementary test scripts are then easily woven into longer test scenarios up to an entire business process without any duplicating content. Test scripts maintain references between each other greatly simplifying the maintenance of the test scenarios. The only update required when a manual task changes consists of updating its associated script: All the test scenarios are automatically up-to-date, reflecting the change.

IBM Rational Functional Tester: Rational Functional Tester strikes a good balance of UI technology coverage and openness. In particular, it offers scripting using non-proprietary languages (Java or VB.NET) making it flexible enough to accommodate most conditions. In addition to that, test scripts can be easily recorded on a combination of different user interfaces, potentially built with different technologies. This makes the orchestrated verification of an end-to-end business process virtually seamless. Furthermore, test execution can be triggered automatically from the command line, making it easy to integrate as part of an end-to-end build/deploy/test process.

Functional testing is definitely the core of any testing effort, but a number of testing dimensions have to be considered above and beyond that, especially in the context of SOA applications. Performance and security are probably among the most important ones.

IBM Rational Performance Tester is a test creation; execution, and analysis tool that validates the scalability and reliability of applications under multiple user loads. Performance tester enables teams to pinpoint system bottlenecks before application deployment. Tests are recordings of a user's activity in a Web browser; teams need little to no programming knowledge to understand and modify these tests. Just a few mouse clicks engages a data pooling capability that ensures unique data for each emulated user. Using an intuitive graphical test scheduler, teams can then organize their tests to accurately simulate the different types of users and user activities the application under test will support.

TANGOSOL™

A Cure for Downtime.

Eliminate the single points of failure, add fault tolerance and enable transparent failover for your J2EE applications.

TANGOSOL COHERENCE™ has no single points of failure, ensuring high availability and reliability for your mission-critical J2EE applications. Application state, HTTP sessions and data caches are never at risk. Applications using Coherence keep it up without interruption, even when servers go down.

Try before you buy.
Download a free evaluation license at

www.tangosol.com

Coherence™ is pure Java software that reliably and cost-effectively scales J2EE applications across any number of servers in a cluster. Coherence provides reliable coordinated access to in-memory data that enables safe data caching and user session management for applications in a clustered environment.

Sales	+1.617.623.5782 or sales@tangosol.com
Support	support@tangosol.com or http://www.tangosol.net
Pricing	US$1,995 to US$4,995 per Coherence production CPU Development licenses are offered free of charge.

Asset Lifecycle Management

One of the key drivers for the new way in which organizations are looking at their next-generation solutions platforms is a focus on asset management and reuse. A new emphasis is placed on creating, harvesting, applying, and managing assets across the lifecycle. For practical reasons, this is accompanied by strong governance practices for reusable assets tied to a flexible asset lifecycle management system.

As organizations create SOAs to leverage existing assets, the ability to accelerate and manage change becomes more critical. Using IBM Rational ClearCase and IBM Rational ClearQuest for software configuration management, teams can automate the software lifecycle and establish a consistent process across distributed environments.

IBM Rational software configuration management products help teams improve project collaboration and release coordination; increase development responsiveness and agility; and enhance operational efficiency.

In addition, reusable SOA-related development assets can be packaged and consumed in the IBM Rational Software Development Platform through direct support for the Object Management Group's Reusable Asset Specification (RAS) standard.

The IBM Software Development Platform includes several other tools that enable the team to more effectively manage asset workflow. WebSphere Studio Asset Analyzer provides support for the Asset Identification workflow. Rational Software Architect provides a RAS client for Asset Production. This tool also includes another RAS client, called the Asset Explorer, to facilitate Asset Consumption. Developers and architects use Asset Explorer to connect to one or more RAS repositories and issue searches that examine the RAS XML manifest in each of the RAS assets.

An Example of Business-Driven Development for SOA

Developers: Constructing the Service

A Java developer designs and implements the architect's specification and constructs the service using IBM Rational Application Developer.

With many development tasks automated, the developer can focus on implementing the appropriate business logic needed to fulfill the service's functional requirements in Rational Application Developer. The developer can then deploy and test the code in the WebSphere Test Environment included with Rational Application Developer.

Alternatively, a developer can repurpose existing assets, and use Rational Application Developer to automate the creation of a Web Service from existing Java classes or EJBs. Both Rational Application Developer and Rational Software Architect can take advantage of the UDDI browser included in the tools to find and reuse existing services in the development of a solution.

Using Rational Application Developer, the developer can register the service's availability in a UDDI registry and create both a WSDL description of the service and an Enterprise Archive (EAR) file to be used by the Integration Developer. Once these assets are created, the developer can then package them and store them in a RAS-compliant repository, such as one based on Rational ClearCase.

Getting Started with IBM and SOA

Making the transition to SOA development doesn't require an enterprise to completely overhaul its IT infrastructure and development processes all at once. Often, the transition takes place in incremental steps. A business can decide to rework an existing business process using an SOA, or implement a single new SOA application. Another organization can begin by adopting SOA-oriented development methodologies and tools for a particular phase of the software development lifecycle.

Summary

Flexibility is fundamental to companies seeking to react to a rapidly changing landscape that includes emerging competitive threats, shifting compliance and regulatory requirements, mergers, acquisitions, and evolving technology. Just as important is the ability to align a core businesses process – software development – with business needs and priorities.

Achieving flexibility and better alignment of business and IT objectives requires executing IT projects with a high level of coordination, accuracy, and clarity. Business-driven development for services-oriented solutions helps businesses create solutions that truly meet an organization's needs today and are readily adapted when those needs change in the future.

The IBM Rational Software Development Platform plays an important role in enabling organizations to create a set of services capable of realizing these goals. It combines market-leading tools, best practices, and other resources to create a rich, integrated environment for service-oriented solutions development.

"When business requirements are used to drive all downstream development activities, the gap between IT and business closes"

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

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

To get your free copy of the informative new article

"3 Steps to Uncomplicating Business Process Management"

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

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

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

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

NASTEL™ TECHNOLOGIES, INC.

Put Your Application Infrastructure On AutoPilot

Best Practices for .NET/WebSphere Interop

Managing Mixed-Mode Deployments

BY LAURENCE MORONEY

Mixed-mode deployments where the data center has a mixture of different technology platforms, hardware, and software and where those platforms interoperate together to deliver software applications is the norm rather than an exception.

Experience shows that it's a challenge getting applications to run across a diverse array of software and hardware platforms with acceptable availability, performance, scalability, and security. As such, several methodologies and standards of integration or interoperability have evolved, the most common being Web Services, which despite their power and flexibility, aren't magic bullets to easy integration, and should be used only where appropriate.

This article will review several strategies for designing mixed-mode deployments and how they impact manageability. A low-level interoperability strategy from Mainsoft, called Platform Unification, will also be introduced to solve many of the problems of interoperability technologies, without adding layers of complexity for development, assurance, deployment, and management.

I'll share best practices on how to design applications to make the most of existing assets, and minimize the difficulties brought about by technology diversity. In particular, I'll focus on the interoperability needs of an application that runs partially on the Microsoft .NET framework using the Windows operating system and Intel-based hardware, and partially on WebSphere running on any operating system and IBM's z9, zSeries, or other eServer platforms.

Emergence of Mixed-Mode Systems

Mixed-mode systems exist for myriad reasons, including:

- ***Mergers and Acquisitions:*** Corporate executives considering a merger or acquisition focus first on aligning their businesses. It's therefore not uncommon for a company that uses WebSphere to suddenly find it needs a system that's open to .NET.
- ***High License Costs:*** With the advent of Linux as a 'low-cost' or 'no-cost' operating system, a company heavily invested in .NET may try to reduce costs by deploying new assets to Linux, which will then need to interoperate with existing systems.
- ***Successful Departmental-Level Applications:*** Companies that use J2EE at the strategic corporate level, but allow other technologies at the department level, may require interoperability at certain function points. Management may also want to implement company-wide department-level applications that require interoperability or migration to a J2EE architecture.
- ***Desire for Service Orientation:*** Many companies want to simplify their interoperability to become service-oriented either within the enterprise and/or to external customers. Integration directly with business partners or customers often necessitates a cross-technology interoperability solution.
- ***High Development Costs:*** The business driver of cutting costs may also impact development costs. Many companies have diverse pools of talent across technology realms, but need them focused on a specific task, thereby cutting training, licensing costs, etc., and possibly reducing "wasted" manpower.

Strategies for Interoperability

There are several strategies that allow for interoperability between.NET- and WebSphere-based software platforms. These include Web Services, bridging solutions, cross-platform implementations of .NET, and an innovative Platform Unification approach from Mainsoft where ASP.NET code runs on WebSphere, using the J2EE runtime, and allowing for much lower-level interoperability than achieved with other solutions. Platform Unification allows for .NETFramework-based applications to make the most of IBM software and hardware solutions, so that a unified data center architecture, based on zSeries and running Linux and WebSphere can handle your entire application domain.

ABOUT THE AUTHOR

Laurence Moroney is the Director of Product Evangelism for Mainsoft. He joined Mainsoft from Reuters, where he was a Senior Architect in their CTO Office, specializing in Enterprise Architecture and Web Services Interoperability. Laurence has also worked extensively in the financial services and security fields. He is the author of several computer books, including *Expert Web Services Security in the .NET Platform* and *ASP.NET 1.1 with VB.NET* as well as several dozen technology articles.
lmoroney@mainsoft.com

Web Services

Web Services are abstract entities that encapsulate business logic using XML for discovery, description, and communication. In the example of a J2EE backend and a .NET frontend, a Web Service layer can be introduced that wraps the business logic of the WebSphere backend and makes it accessible to the .NET frontend. Communication between the two should be transparent. This form of architecture is called loosely coupled. A major selling point and strength of the WebSphere Application Development environment is just this – easing the service orientation of the business assets that are deployed on WebSphere.

However, it should be noted that while a loosely coupled, standards-driven architecture should make interoperability easy, in practice varying interpretations of standards have meant that Web Services written in one language on one platform don't always understand those in another language on another platform. Also, Web Services incur a large performance overhead, so while it may be tempting to wrap high-performance EJBs running on a zSeries platform in Web Services to expose them to other applications, you may lose the benefits of the technology and the platform with the overhead that you incur. From the management point of view, the following observations are made:

- ***Fault Management:*** Web Services are widely adopted and are associated with many standards so that if the web Service is implemented using a standard, then it can be trusted to meet the appropriate criteria. In the case of management and fault management, WS-Management and MOWS standards encapsulate this. If the service is written to these standards, then many tools that allow for fault management and analysis are available.
- ***Configuration Management:*** Standards such as WSDM and MOWS address the issues of configuration management such as service changes, deployment, and lifecycle management.
- ***Accounting Management:*** MOWS address the functionality of metering services as well as auditing and integration with modules such as Service Level Agreement (SLA) management.
- ***Performance Management:*** Here is where Web Services break down from a management point of view. They introduce significant overhead in needing to process intricate XML documents for SOAP and WSDL. When the service adheres to specifications such as WS-Security and WS-Management the complexity of these documents, and thus the performance hit can increase exponentially.

Bridging Solutions

Bridging solutions are tightly coupled solutions providing a messaging transport and translation layer between components running on diverse systems. Reviewing a hypothetical example system where the frontend presentation tiers are developed using ASP.NET and the backend business logic and data management tiers are implemented on WebSphere, a typical bridging solution would provide a layer that the ASP.NET could call using remoting or other familiar semantics. The bridging solution would handle the low-level details of the communication allowing developers to call remote objects passing data, state, and the like. This may appear at first to be an elegant strategy, but it suffers from challenges insofar as management is concerned.

These include:

- ***Fault Management:*** Bridging solutions don't integrate tightly with the underlying management APIs of either J2EE or .NET. Standard or integrated metering and logging of either of these may not be effectively leveraged.
- ***Configuration Management:*** Commercial solutions for the above scenario include Js.NET and JNBridge. These require runtime configuration by means of text files containing the relevant information. These files have to be deployed carefully and managed on both sides of the system, and in many cases may have to be regenerated on small changes to the system.
- ***Accounting Management:*** Generally bridging solutions don't integrate with the underlying management APIs, and don't offer the facility for unified accounting.
- ***Performance Management:*** Bridging solutions bring an overhead to the performance of the system that is significant, and in some cases major. The runtime data marshalling can reduce overall performance.
- ***Security Management:*** Security and security context management handling can vary from implementation to implementation, but when using a bridging solution, a new channel of communication is introduced tht may or may not allow for a compatible security context with the rest of your J2EE or .NET architecture.

The Enterprise Service Bus

The Enterprise Service Bus (ESB) is an evolving concept, having grown from the concept of Web Services, but stretching beyond the simple request/response model offered by Web Services into the entire realm of application integration. The concept is modeled around a common communications pathway, a bus, to which all types of application (request/response, asynchronous, message driven, legacy, etc.) are connected via adapters. Thus all information flows on a common information bus, using a common format: SOAP.

The concept is interesting, and in its infancy, so ascertaining its manageability is difficult. However – one can deduce that since it's an amalgamation of all forms of application integration, using adapters to set a common means of communication, it should fare like the Bridging methodology when it comes to gauging its manageability.

EAI Applications

There are many EAI applications designed for application integration.

Examples are WebSphere Business Integration, Microsoft BizTalk, TIBCO, and BEA products. These follow similar principles to the Enterprise Service Bus, whereby adapters provide a translation layer between different standards to a single internal data format that is then managed by the EAI application and dispersed where appropriate. It appears to be very similar to the Enterprise Service Bus, and it is, but for one major difference – in EAI applications the communication and data representation format is generally a proprietary format, whereas with the ESB it's standards-based. From the point of view of management, it should fare like the Bridging methodology.

Mono: A Cross-Platform Implementation of .NET

Mono is an Open Source project, sponsored by Novell, that aims to bring the power of the .NET framework and the flexibility of development in C# and VB.NET to platforms other than Windows. It encapsulates an implementation of the CLI runtime as well as implementations of many of the .NET Framework APIs and system classes. It has been successfully used in migrating large applications from Windows to Linux – for example, a recent project by the city of Munich used Mono to migrate their ASP.NET applications to a large-scale deployment of over 300 Linux servers.

Mono implements a subset of the .NET management APIs, but doesn't provide an integrated management framework across platforms, nor does it integrate them with native management functionality that may be present on the operating system. The implementation of the key management elements such as Fault Management, Configuration Management, Accounting Management, and Performance Management would therefore need to be implemented by the application developer.

It's important to note that this isn't necessarily a limitation of Mono, since Mono isn't intended to be an interoperability solution, but a multi-platform one. It is, however, open, extensible, and continually adding new features, so with a little effort it can be a very effective solution. Therefore interoperability between Mono and a WebSphere-based application can still be implemented and managed using technologies built on Mono such as Web Services.

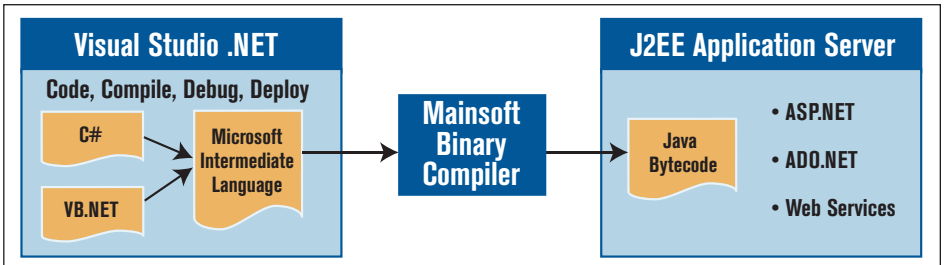


Figure 1: The .NET-to-J2EE compilation process

Platform Unification

Managing mixed-mode systems brings enough challenges of its own without adding more through an interoperation technology such as Web Services or Bridging. This is part of the rationale behind the Platform Unification strategy designed and implemented by Mainsoft.

Visual MainWin for J2EE establishes compile time conversion from .NET Framework-based code such as C# or VB.NET to Java bytecode that will run on WebSphere. As such, .NET applications will interoperate with WebSphere applications by running directly on WebSphere. Given the Open Source nature of Mono, Mainsoft can convert the supported .NET Framework classes to Java, providing a .NET Framework runtime that runs on WebSphere. In a symbiotic fashion, Mainsoft then returns significant amounts of code and QA to the Open Source community. In addition, Visual MainWin is implemented as an add-in to the Visual Studio.NET IDE, enabling developers to use a familiar coding environment. Developing, deploying, and debugging applications on J2EE is seamless and virtually identical to their experiences on the Windows/IIS platform.

This solution removes the need for additional interoperability technologies are such as Web Services, or Bridging, although the former is certainly an excellent technology for presenting a consumable API frontend to customers should it be required.

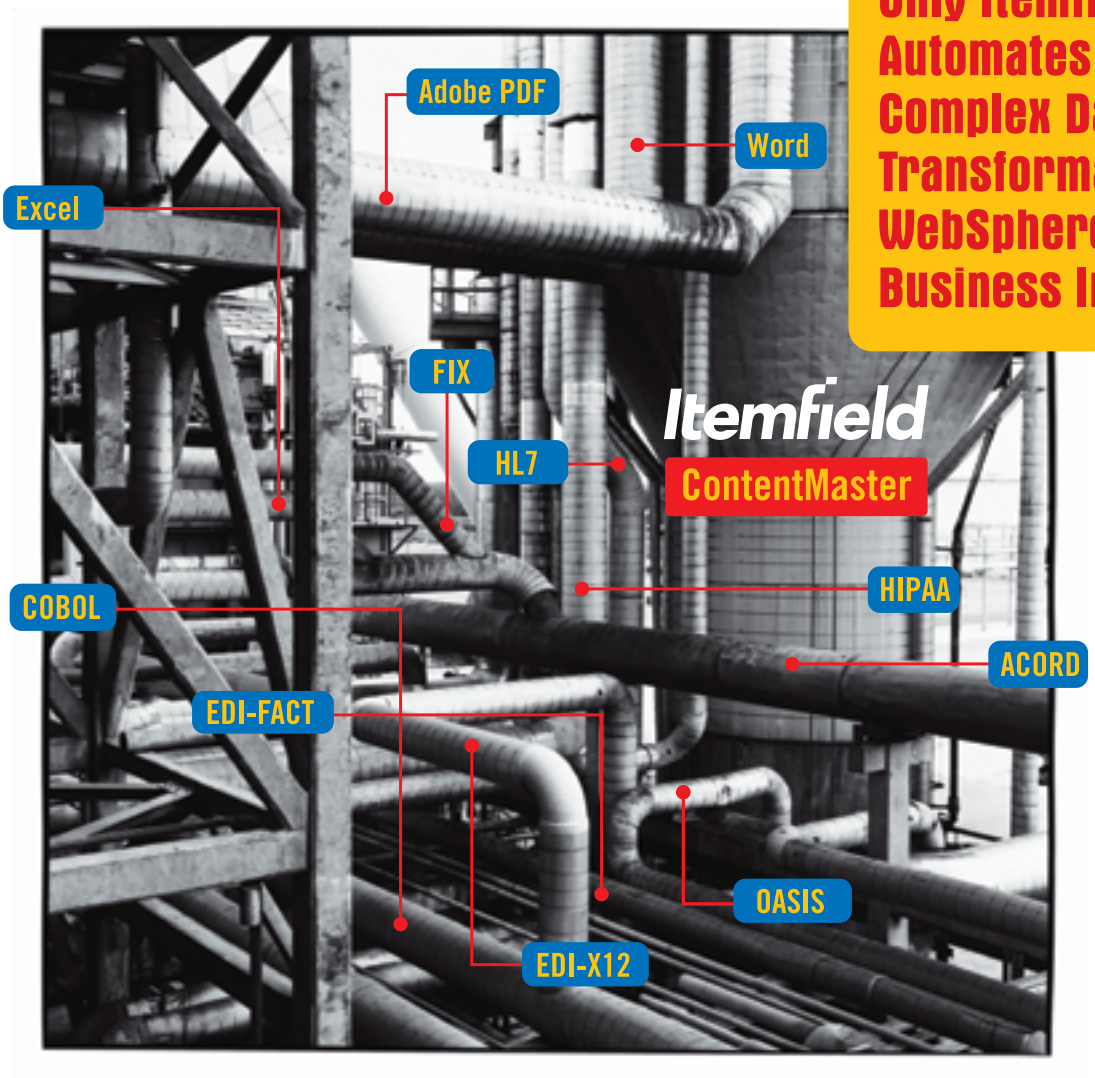
From a management point of view, the platform unification strategy is a strong approach.

- ***Fault Management:*** Monitoring, event handling, logging, etc., are all implemented using the underlying J2EE mechanisms, and are fully compatible with the WebSphere environment and any existing management systems, such as Tivoli.
- ***Configuration Management:*** Overall systems management is greatly simplified due to the homogenous nature of the platform. Tuning, updates, and patch management are performed for a single system configuration since all components are deployed as J2EE modules. Application management and update deployment are also greatly simplified. All resources can be managed uniformly through the J2EE management console.
- ***Security Management:*** Under the platform unification model, security is provided through J2EE. This homogenous model is used across the application, and allows managing users, authorization, and authentication in a single domain. This simplifies the management of security, and eliminates the need to synchronize user and authorization stores across multiple domains. It also decreases security risks of credential information being passed between domains and platforms on wire protocols such as HTTP or HTTPs. Finally it lets .NET components benefit from the J2EE security model, allowing the use of J2EE declarative security. .NET code can access J2EE APIs and this allows the implementation of programmatic security.

Understanding Visual MainWin

Visual MainWin converts Microsoft Intermediate Language (MSIL) into Java bytecode then executes it on WebSphere. This conversion handles the difference between .NET and Java semantics while

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

producing efficient and fully compliant Java bytecode. The key elements of the conversion are:

- ***Direct Mapping:*** The compiler directly maps object structure, classes, interfaces, and types as well as the flow of control semantics from .NET to the equivalent Java semantics.
- ***.NET Semantics:*** The compiler handles .NET semantics not present in Java such as .NET unsigned types and .NET value types. .NET delegates by generating Java bytecode with the equivalent functionality.

The binary compiler resolves differences between the .NET and Java platforms and produces the Java bytecode as an output. This is then packaged as a Java JAR or WAR file for deployment, along with all the supplementary .NET Framework runtime support classes.

The .NET runtime environment is hosted on J2EE by an implementation of the .NET Framework in J2EE, provided by Mainsoft and based on the Mono project.

Figure 2 shows how this works through the Visual MainWin for J2EE technology stack. .NET-based code runs on the .NET Framework class library, implemented by using the Mono source code, compiled into Java bytecode, and hosted on WebSphere as well as the core Data and .NET classes from mscorlib (a fundamental part of the .NET Framework) implemented directly by Mainsoft on Java. These then run using standard Java packages such as JDBC. For example, .NET code built using the System.Data namespace will convert into Java bytecode using the Java-based implementation of System.Data that runs on top of JDBC.

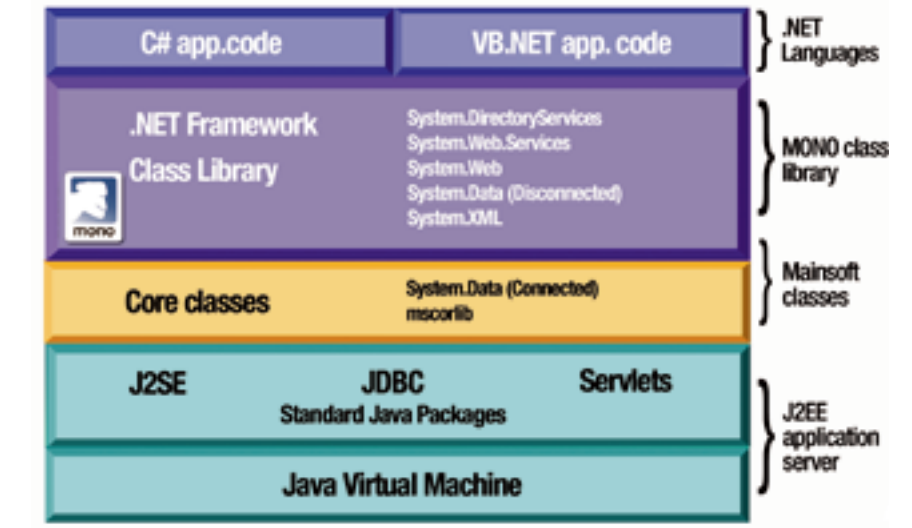


Figure 2: The Visual MainWin for J2EE technology stack

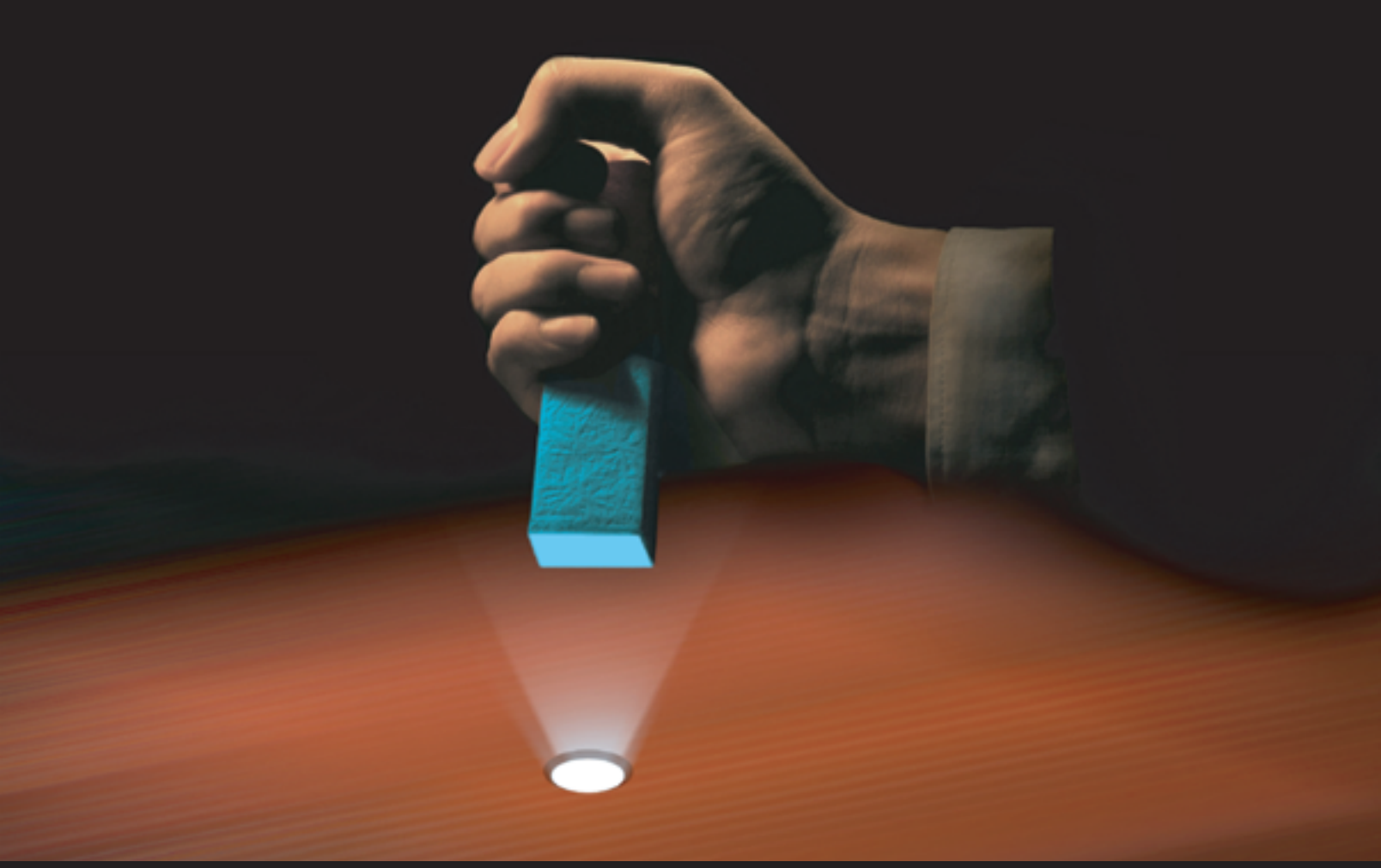
Interoperability is achieved through native support. Since the runtime for all compiled code is WebSphere, the .NET programmer and applications can access Java classes, EJBs, and namespaces transparently from the .NET environment as though those classes were .NET classes. The JDK and J2EE classes are all available to the .NET developer should the need arise. It works both ways – .NET classes built on the unified platform are also available to the Java programmer and applications.

In a distributed environment, .NET code can reference Java code that's deployed as an EJB or Web Service. Since the entire environment is Java, point-to-point integration with EJB assets is available over RMI. The developer experience is simple – they create a proxy to the EJB in a way similar to that used for consuming Web Services. Additionally the .NET programmer can access the J2EE server context, which is useful, and necessary for manipulating JNDI objects such as data sources.

Conclusion

Managing mixed-mode systems is an intricate affair, made even more difficult by the additional layers of complexity that are added for interoperability. In this article we reviewed the various methodologies that are available to allow for interoperability and their impact on the complexity of the system. We explored the various solutions in the context of their impact on fault management, configuration management, accounting management, security, and interoperability. In addition, Platform Unification was shown to have a great impact on reducing the complexity of mixed-mode systems and positively affecting the management of the system.

> **"Managing mixed-mode systems is an intricate affair, made even harder by the layers of complexity added for interoperability"**



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

Think JReport.

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

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

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

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

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

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

JReport
JINFONET SOFTWARE

Performance Considerations for Custom Portal Code

Ensuring Optimization

BY RAINER DZIERZON, KLAUS NOSSEK, & MICHAEL MENZE

(This article was first published on developerWorks WebSphere at http://www.ibm.com/developerWorks/websphere.)

(This is the second part of a two-part article. The first part appeared in the September 2005 issue of WebSphere Journal.*)*

Large session objects decrease the JVM memory available for creating and executing application objects. As a result, performance can degrade as the decreased available heap memory leads to more frequent garbage collection.

Another factor is that the in-memory lifetime is always longer than the required usage lifetime, and so the number of sessions occupying space in the Java heap is usually greater than the number of active users. A session expiration time is configurable in WebSphere Application Server and is actually required to avoid a case when a user has to log in again after only a few seconds of inactivity. The release of a session is the responsibility of WebSphere Application Server and the portlet container.

The serialized session size should be smaller than 4KB because WebSphere Application Server can store such sessions with an acceptable database performance overhead, and it takes less time to transfer such sessions over the network. If the session size grows beyond 32KB, the database must use table cells configured for binary large objects that require physical disk access (for most supported databases) if such a session is retrieved or written to the database.

As a first consequence, the creation of sessions should be avoided wherever possible from an application point-of-view. On most public and unauthenticated pages, sessions aren't usually required. Interacting with a portal on such a page is possible via so-called render links that, by definition, don't change the server-side state. Render parameters are maintained by the portal for each portlet for all subsequent requests to that page. To avoid having a JSP create a session by default, the page session directive in the JSP should be set to false:

```
<@ page session="false"%>
```

Otherwise, this JSP will create a session if one doesn't exist.

ABOUT THE AUTHORS

Rainer Dzierzon is team lead of the WebSphere Portal performance team located in the IBM Development Laboratory in Boeblingen, Germany. Since 1990, he has worked on numerous software projects for several companies in the area of database performance tools, text search, SyncML, online banking, and financial services architecture and standards. He and his team work as part of the WebSphere Portal development community to provide developers with the insights into performance behavior and to consult customers and solve their performance issues. He holds a diploma in computer engineering from the University of Cooperative Education in Stuttgart.

Klaus Nossek is a performance analyst on the WebSphere Portal team at the IBM Development Laboratory in Boeblingen, Germany. He received a diploma in computer science from the University of Stuttgart in 1996. After working for several years on various software projects, he joined the IBM WebSphere Portal Division in 2003. Currently, he belongs to the WebSphere Portal performance team. His main areas of work are developing performance scenarios for WebSphere Portal and Java code profiling for detailed performance analysis.

Michael Menze joined IBM development in 2002 after studying with IBM for three years. He works in the Boeblingen development team for WebSphere Portal and is responsible for WebSphere Portal performance analysis, and the introduction of performance improvements.

The following Java code fragment shows how you can make sure that an incoming request joins an existing session, rather than unconditionally creating a new one:

```
PortletRequest.getPortletSession(false)
```

With the parameter value of false, a session isn't created if no session existed before. If a session didn't exist before, it's probably not appropriate to create one in a portlet just for the sake of storing data in it. As a second consequence, the session should not be misused as an all-purpose data store mechanism. Remember that the goal is to keep the session size as small as possible. If keeping some data in memory is advantageous due to the design of a portlet, then a cache might be the right answer. Cache entries can be scoped with the session ID to keep a relationship between the session and the data that will be kept in memory. Keep in mind that this kind of cache won't be cluster-aware in case of a failover; this is sometimes an acceptable tradeoff. If the data is re-creatable from other data available to the portlet, then the session scope requirement of cached entries is questionable.

In many cases storing large objects in the session can be circumvented by just storing a key in the session and using this key as a reference to look up a larger object in some other data structure. Another option would be to use a more compact representation of the same information and put that object into the session.

Furthermore, the portlet design has to carefully consider what is actually stored in a session. The session is generally intended only for storing the conversational state of the user interaction with the portal application (for example, the contents of a shopping cart in a Web shop portlet). This kind of data is user-specific and can't be recreated by any other means. In WebSphere Portal, this kind of data handling is called a session state.

If session state isn't really required, there are other data storage options available for portlets:

During the action phase of a portlet, render parameters can be set for the portlet's subsequent render phase. A portlet uses render parameters to render its view specific to a specific set of values. Render parameters are maintained by the container request to request, even if interaction occurs with another portlet. In WebSphere Portal, this kind of data handling is called a navigational state.

The PortletPreferences API can be used for storing data for a portlet if such data will be kept across user sessions. Keep in mind that this API isn't intended to replace general-purpose databases. In WebSphere Portal, this data-handling concept is called persistent state.

The PortletConfig API lets a portlet read its configuration, which is provided by the developer using the portlet deployment descriptor; this is valid for all portlet users. The PortletContext API enables the storing of attributes that other portlets in the same application can also access.

Consider other choices than the session for storing data created and used by a portlet. Avoid replicating data into the session that can be re-created from sources other than through user interaction.

Render Links & Action Links

There are advantages to using render parameters other than just to address a specific portlet view.

If an action parameter for a portlet is detected by WebSphere Portal then special action phase handling must be invoked, making it advantageous to avoid using action parameters. However, be aware that processing render links mustn't change the server-side state of a portlet. The only sanctioned way to change the server side state is to use action links – and for a transactional type of request, action links are the best choice.

There are many instances where render links can be used instead of action links. For example, consider a newspaper portlet that can show specific pages with the use of the Previous and Next buttons. Stepping through the newspaper pages doesn't necessarily change the server-side state, which in this case is the entire information contained in the newspaper. To address the next page of the newspaper, it would be sufficient to encode the next page number into the render link for the shown button. The portlet can decide which page to render based on the page number given in a render parameter.

Also, using render links over action links enhances the chance of leveraging a cache infrastructure, be it a browser cache or a proxy cache, since each rendered view is addressed by a separate URL. The URL is the only key used to access a specific generated view in such a cache infrastructure.

Portlet Features

The next sections discuss some of the portlet-tuning features available with WebSphere Portal that should be considered by developers, and could influence the implementation technique chosen. Some required settings have to be provided with the portlet's deployment descriptor, and since these items are also supplied by a portlet developer, they're considered custom code.

Enabling Portlets for Parallel Rendering

WebSphere Portal offers the option of having portlets on a page rendered in parallel. This feature isn't completely "for free" because computing resources are required to maintain and manage the different threads that are used to render each single portlet.

Parallel portlet rendering can be advantageous in cases where many backend systems are involved that produce their own latency while rendering a single page. For example, consider a portal page that contains a number of portlets, each of which accesses a different backend system. In serial rendering mode, the overall latency for retrieving the required data from all the backend systems would have to be calculated as the sum of the individual latency times. In parallel rendering mode, the latency would be the maximum of all individual latency times.

If portlets don't use a backend system too often, the overhead for enabling parallel portlet rendering can become greater than the benefits gained from this feature. If portlets on a page can be rendered independently of a backend system, they only need CPU resources local to the portal

server machine. In this case, the page render response time won't be improved.

Parallel portlet rendering can be enabled for each portlet separately with the graphical UI, or with the deployment descriptor, or with WebSphere Portal's XML access interface. On top of that, there's a global property value that generally turns parallel portlet rendering on and off.

To properly answer the question of whether or not to enable parallel portlet rendering for a portal, there are several things to consider; for example, the number of backend systems involved for rendering a page, typical page structure, the average number of portlets on a page that exploit parallel portlet rendering, and so on. Such questions may not necessarily be answerable in advance by a portlet developer, but certainly the developer can make sure that a portlet is enabled for parallel portlet rendering if that makes sense up-front.

Caching in the Portlet Container

Portlet-based Web pages are aggregated dynamically because of their ability to deliver dynamic content in a personalized manner. This flexibility comes at a cost. Web site response time increases because of the additional work that has to be done to generate these pages on request.

New caching technologies improve the performance of dynamic page generation and reduce system load. WebSphere Portal supports fragment caching (also known as servlet caching) using the WebSphere Application Server dynamic cache to keep portlet output in the cache. Requests for a cached portlet retrieve the content from the cache instead of the portlet. The invalidation of the fragment cache can be accomplished by specifying the expiration time in the deployment descriptor. Further, the fragment cache entries are invalidated during the action phase of the portlet.

There's no time-consuming installation and integration work needed to activate fragment caching. The cache is enabled and disabled using simple XML deployment descriptor files and the WebSphere Application Server administrative console. (See the WebSphere Portal Information Center for details on enabling servlet caching in WebSphere Application Server.)

To make use of expiration-based caching, portlets must define the duration of the expiration cache in the deployment descriptor portlet.xml (for standardized portlet following JSR 168 specification):

```
<expiration-cache>300</expiration-cache>
```

A positive number defines the number of seconds a cache entry exists in the cache.

- A value of -1 indicates that the portlet cache never expires.
- A value of 0 indicates that caching is disabled for the portlet.

The cached entries must not be shared across all users of the same portlet. This caching is per-portlet per-user.

For a JSR 168 portlet that has defined an expiration cache in its deployment descriptor, the portlet window can modify the expiration time at runtime by setting the EXPIRATION_CACHE property in the RenderResponse as follows:

```
renderResponse.setProperty(
   RenderResponse.EXPIRATION_CACHE,
   String.valueOf(numberCrunchingCalculation())
);
```

This approach will work for complex portlets that experience high computation time while calculating their response or request data from a backend, such as from EJB components or a database. In the case of simple portlets, fragment caching shouldn't be enabled. WebSphere Portal uses extra execution resources to calculate the internal cache key for the fragment cache. Performance can regress for simple portlets because cache key calculation becomes more expensive than recalculating the portlet response again.

Fragment caching isn't useful for portlets that are truly dynamic in nature; for example, real-time-based portlets that have to collect current data from other data sources on each request, or portlets that change their response markup on every request. This would result in a high number of cache invalidations and hence there would be no gain in performance. Therefore, the portlet should be enabled for caching only if the output of the portlet will be valid for some period of time before it's updated.

Caching in Remote Caches

With the unique adaptive page-caching feature, WebSphere Portal offers the possibility of dynamically caching generated pages in caches external to a portal server (so-called remote caches), if all the page components indicate that they're cacheable. If completely rendered pages are served from remote caches, a roundtrip to the portal server can be avoided and response times for such pages can potentially be as fast as if they were served from static Web sites.

For more complete details on remote caching. see the article Develop high performance Web sites with both static and dynamic content using WebSphere Portal V5.1.

Portlets (as well as themes) can contribute their specific remote cache information to the overall remote cache information for a completely rendered page. The remote cache information is a data structure consisting of the information about the cache scope (whether it's cacheable in a shared or non-shared cache) and the expiration time (how long the content is considered valid). The portlet's remote cache information can be provided via the deployment descriptor or the WebSphere Portal GUI. Beyond that, a portlet can also provide remote cache information at render time for each portlet window, as illustrated in the following code:

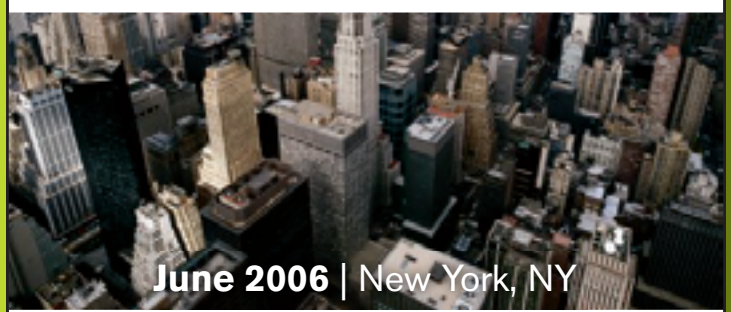
```
. . .
import com.ibm.wps.util.RemoteCacheInfo;
import javax.portlet.RenderResponse;
. . .
/* Do rendering */
public void doView(RenderRequest renderRequest,
RenderResponse renderResponse)
   throws PortletException, IOException {
   /* Some code might happen here */
   . . .
   /* Publish a dynamic expiration time during
rendering */
   renderResponse.setProperty(
      RenderResponse.EXPIRATION_CACHE,
      String.valueOf(numberCrunchingCalculation())
   );
   /* Publish a cache scope value of "shared"
```

ENGAGE AND EXPLORE...

The Technologies, Solutions and Applications that are Driving Today's Initiatives and Strategies...

CALL FOR PAPERS NOW OPEN!

SOA WebServices Edge 10th International conference+expo 06



The Sixth Annual SOA Web Services Edge 2006 East - International Web Services Conference & Expo, to be held June 2006, announces that its Call for Papers is now open. Topics include all aspects of Web services and Service-Oriented Architecture

Suggested topics...

- *Transitioning Successfully to SOA*
- *Federated Web services*
- *ebXML*
- *Orchestration*
- *Discovery*
- *The Business Case for SOA*
- *Interop & Standards*
- *Web Services Management*
- *Messaging Buses and SOA*
- *Enterprise Service Buses*
- *SOBAs (Service-Oriented Business Apps)*
- *Delivering ROI with SOA*
- *Java Web Services*
- *XML Web Services*
- *Security*
- *Professional Open Source*
- *Systems Integration*
- *Sarbanes-Oxley*
- *Grid Computing*
- *Business Process Management*
- *Web Services Choreography*

CALL FOR PAPERS NOW OPEN!

2006 ENTERPRISE> OPENSOURCE CONFERENCE+EXPO

The first annual Enterprise Open Source Conference & Expo announces that its Call for Papers is now open. Topics include all aspects of Open Source technology. The Enterprise Open Source Conference & Expo is a software development and management conference addressing the emerging technologies, tools and strategies surrounding the development of open source software. We invite you to submit a proposal to present in the following topics. Case studies, tools, best practices, development, security, deployment, performance, challenges, application management, strategies and integration.

Suggested topics...

- *Open Source Licenses*
- *Open Source & E-Mail*
- *Databases*
- *ROI Case Studies*
- *Open Source ERP & CRM*
- *Open-Source SIP*
- *Testing*
- *LAMP Technologies*
- *Open Source on the Desktop*
- *Open Source & Sarbanes-Oxley*
- *IP Management*

Submit Your Topic Today! www2.sys-con.com/events

Sponsored by

WebServices JOURNAL · XML JOURNAL · .NET DEVELOPER'S JOURNAL · eclipse developer's journal · WebSphere JOURNAL · information STORAGE+SECURITY journal · wldj · JDJ · LINUXWORLD MAGAZINE · MX developer's journal · asp.netPRO · SD Times · CoDe · Software Test & Performance

Attention Exhibitors:
An Exhibit-Forum will display leading Web services and OpenSource products, services, and solutions

*Call for Papers email: jimh@sys-con.com

For Exhibit and Sponsorship Information - Call 201 802-3066

Produced by SYS-CON EVENTS

```
during rendering */
   renderResponse.setProperty(
      RemoteCacheInfo.KEY_SCOPE, RemoteCacheInfo.
Scope.SHARED_STRING );
   /* Some other code might happen here */
   . . .
}
```

How you set remote cache information depends on the "freshness" requirements and the scope of the rendered view. Keep in mind that if rendered pages are served from caches, the request might not even get to the portal server.

Custom portlet developers should consider exploiting remote caching if caching is available in the infrastructure.

Themes and Skins

In portal terminology, themes are sets of JSPs that determine the look-and-feel of your portal application. Since themes are made of JSPs, the tips offered in the JSP section also apply here. This section details some possible performance pitfalls with the set of JSP files that gets combined into a theme.

Typically, a theme consists of many different JSP files, each delivering the content for a certain area of the screen. While it's possible to dynamically include JSPs, it's common – and generally recommended – for JSPs to be statically included in other JSPs.

Since many JSPs might be included in others at compile time, the resulting Java source and servlet bytecode files can potentially be very large. In general, there's no performance problem with having a large class file, but it's possible that compiling the JSP sources to a class can fail due to size restrictions incorporated in the Java programming language. For example, methods in Java can't be larger than 64KB. Large and complex themes can easily reach this limit and will then no longer compile. In this situation, you have three options:

- ***Substitute dynamic includes for some (but not all) static ones.*** As mentioned in the JSP section, this trades performance for being able to compile the JSPs at all. From a performance standpoint, this is the least preferred resolution, though it's the easiest to implement.
- ***Try to restrict the use of scriptlets in the JSPs.*** WebSphere Application Server can apply optimizations to code that only calls tag handlers, which can help maintain the 64KB limit.
- ***Clean up your JSP code.*** Very often these files contain more code than is necessary. Even removing HTML comment lines or white spaces or moving JavaScript code into separate files can save sufficient space.

Themes sometimes take over complex tasks in an application. However, you have to be careful here. Remember that the theme will be rendered on every single request to your portal, so don't introduce expensive computations that put a high burden on the system.

Be especially careful with mimicking portal functionality. For example, themes could iterate over large numbers of pages in the portal application and then filter these and only display a navigation structure to the user that includes just a few of the pages the theme requested from the portal APIs. In this case, much of the processing that happens inside the portal will be lost since the results are discarded afterwards. Filtering based on portal access control or personalization rules would be more efficient here.

Furthermore, try to limit the number of links to portal resources from your portal pages. Each URL that the portal has to generate puts additional load on the system. If you need application themes with huge numbers of links, try to cache the contents of some of these pages so that it is not necessary to recalculate all the links on every request.

Themes are also part of the remote caching infrastructure in WebSphere Portal. The remote cache information of a theme is a set of specifically named metadata that can be set via XML access, as shown in the following example:

```
<!-- Theme "shared" scope and 40 seconds cache
expiration -->
<theme action="update" active="true"
objectid="xmplTheme" uniquename="wps.theme.
example">
   <parameter name="remote-cache-scope"
type="string" update="set">SHARED</parameter>
   <parameter name="remote-cache-expiry"
type="string" update="set">40</parameter>
</theme>
```

A theme can't provide any render-time remote cache information.

WebSphere Portal supports the notion of high-performance skins. These skins are special insofar as they aren't generated based on JSPs; their output is created from precompiled Java classes. Of course, this kind of skin is less customizable; you can only modify the stylesheet information and included images. However, if performance is your most critical factor, you should think about enabling high-performance skins for certain elements on your page or for certain portlets. (See the Information Centers in Resources for more information, including various hints that can help you program high-speed skins and themes.)

Tools

Tools are available to assist you at all stages of WebSphere Portal application development and verification. This section describes different categories of tools that you can use during different development cycles and provides a few examples to help you get started with developing and analyzing your custom code.

Development Environment

Technically, you can use any kind of text editor to write portlets, themes, and skins, but it's a lot easier using an integrated development environment like IBM Rational Application Developer together with the IBM Portal Toolkit. Portlet code samples and basic portal code fragments are also available to help you achieve your first results very fast; the development environment is integrated with a portal server so you can immediately deploy and test your code.

Performance Analysis Tools

When your code is ready for deployment, you have to understand its performance implications in detail. There are several steps

Visit the *New*

www.SYS-CON.com

Website Today!

The World's Leading *i*-Technology News and Information Source

24/7

FREE NEWSLETTERS
Stay ahead of the i-Technology curve with
E-mail updates on what's happening in your industry

SYS-CON.TV
Watch video of breaking news, interviews with industry leaders, and how-to tutorials

BLOG-N-PLAY!
Read web logs from the movers and shakers or create your own blog to be read by millions

WEBCAST
Streaming video on today's i-Technology news, events, and webinars

EDUCATION
The world's leading online i-Technology university

RESEARCH
i-Technology data "and" analysis for business decision-makers

MAGAZINES
View the current issue and past archives of your favorite i-Technology journal

INTERNATIONAL SITES
Get all the news and information happening in other countries worldwide

JUMP TO THE LEADING i-TECHNOLOGY WEBSITES:

IT Solutions Guide
Information Storage+Security Journal
JDJ
Web Services Journal
.NET Developer's Journal
LinuxWorld Magazine
Linux Business News
Eclipse Developer's Journal
MX Developer's Journal
ColdFusion Developer's Journal
XML Journal
Wireless Business & Technology
Symbian Developer's Journal
WebSphere Journal
WLDJ
PowerBuilder Developer's Journal

you can take, and they are summarized below, but there's one general rule that always applies to performance: In most programs, about 80% of the execution time is spent in 20% of the application code. This 20% of code is on the "critical path" and it's these areas that are worth optimizing for performance. For example, the render method of a portlet is much more performance-critical than its init method, since it's called on in every request.

- Code profiling should be done in the early stages of development, or as a first performance test after development. Profiling means that execution time information is collected at the method level, often using the JVMPI interface. Profiler results help you identify the critical path of your application; that is, the code that's executed most of the time. Profilers also often give information on object creation rates and memory consumption.
  - IBM Rational Application Developer comes with a profiling agent you can use to test your programs.
  - Eclipse Test and Performance Tools Platform is an Eclipse project that provides Open Source profiling support based on Eclipse.
- Once your portlet has been deployed into a portal, you should test for the behavior of the portlet under load. Stress or load generators, like Rational Performance Tester, Rational Robot, Apache JMeter (as a cost-effective alternative), and others, are load-testing solutions to help you accurately simulate the system behavior under production load. These tools collect much information to help you determine whether your system is in good performance shape, including data on request response times, and processor utilization.
- During load testing, you should monitor several performance parameters in your portal environment. IBM Tivoli Performance Viewer (which is shipped with WebSphere Application Server) can be helpful to monitor resource utilization in the application server.
- Many problems with portal environments are memory-related. JVM implementations provide two kinds of information for tools to analyze for performance:
  - Output from the garbage collector, verbose:gc.
  - Heap dumps, which are helpful when hunting for memory leaks.

Check out IBM alphaWorks for analysis tools for the garbage collector output; heapRoots, on the other hand, is a powerful heap dump analysis aid. The IBM Java Diagnostics Guides also provide helpful information for dealing with portal-related performance issues. See Resources for links to these references.

Very often you won't need this complete set of tools when developing code for WebSphere Portal, but for larger portal rollouts in your production environment it's essential to understand your portal code from a performance point-of-view.

Conclusion

When creating custom portal code, there are a number of areas that the developer must consider to ensure that portal performance is optimized. To summarize:

- Focus on improving the critical code path. A critical code path is one that takes a long time to process or is frequently executed. Find out which methods of which classes are on the critical path. Outside the critical path optimizations are rather useless.
- Consider both execution performance and memory allocation.
- Use appropriate tools to measure and profile your code for the most typical user interactions.
- Solutions to a coding problem may differ in orders of magnitude with respect to performance.
- A specific implementation to solve a discovered performance problem has to be understood in detail.
- Consider the backend access pattern while designing your custom code.
- Don't misuse the session as an all-purpose data store for a portlet. There are better ways to handle data for various implementation requirements.
- Consider exploiting the special features provided by WebSphere Application Server and WebSphere Portal to optimize portlet performance, provided the target environment is also exploiting the same feature(s).

Resources

Learn

- *JSR 168: Portlet Specification*
- *Best practices: Developing portlets using JSR 168 and WebSphere Portal V5.02*
- *Meet the Experts: Marshall Lamb on WebSphere Portal Programming*
- *Portlet Development Best Practices and Coding Guidelines*
- *WebSphere Application Server Development Best Practices for Performance and Scalability*
- *IBM WebSphere Portal for Multiplatforms Version 5.0 Performance Tuning Guide*
- *WebSphere Application Server 5.1 Information Center*
- *WebSphere Business Integration Server Foundation 5.1 Information Center*
- *WebSphere Portal 5.1 Information Center*
- *Redbook: IBM WebSphere V5.1 Performance, Scalability, and High Availability WebSphere Handbook Series*
- *Redbook: WebSphere Application Server Enterprise V5 and Programming Model Extensions WebSphere Handbook Series*
- *Java Performance Tuning by Jack Shirazi, O'Reilly & Associates (2003), ISBN: 0596003773*
- *Java Performance and Scalability by Dov Bulka, Addison Wesley (2000), ISBN: 0201704293*
- *Java Platform Performance: Strategies & Tactics by Steve Wilson, Addison-Wesley (2000), ISBN: 0201709694*
- *Performance Analysis for Java Websites by Stacy Joines, Addison Wesley (2002), ISBN: 0201844540*
- *IBM Rational Application Developer*
- *Eclipse Test & Performance Tools Platform*
- *JVMPI documentation*
- *IBM Java Diagnostics Guides*
- *IBM WebSphere Developer Technical Journal: Develop high performance Web sites with both static and dynamic content using WebSphere Portal V5.1*

Get Products & Technologies

- *IBM Pattern Modeling and Analysis Tool for Java Garbage Collector*
- *Diagnostic Tool for Java Garbage Collector*
- *heapRoots*

Extending JAAS

Advanced Authentication in WAS

BY KEYS BOTZUM, BILL HINES, PAUL ILECHKO, MESSAOUD BENANTAR

The advanced authentication features in IBM WebSphere Application Server V6 support a more flexible authentication model with a new highly customizable authentication framework that's based on – and extends – Java Authentication and Authorization Service (JAAS).

ABOUT THE AUTHORS

Keys Botzum is a Senior Technical Staff Member with IBM Software Services for WebSphere. He holds a Masters degree in Computer Science from Stanford University and a B.S. in Applied Mathematics/Computer Science from Carnegie Mellon University.

Bill Hines is a Senior Certified Consulting IT Specialist with IBM Software Services for WebSphere. His expertise includes installation, configuration, tuning, dynacache, security, troubleshooting, and design/architecture of enterprise J2EE WebSphere applications.

Paul Ilechko is a Senior Solutions Architect with IBM Software Services for WebSphere. He has been involved with WebSphere and J2EE technology almost since their inception. He has a B.Sc. in Mathematics from the University of London.

Messaoud Benantar is a senior software engineer with IBM Software Services for WebSphere. He spent most of his career as a software developer working in IBM Large Systems Division, Software Group/Tivoli and IBM Global Services. His focus is on middleware and computing security.

In the past, IBM WebSphere Application Server had a rigid authentication model that made it challenging to support complex or unusual requirements. This situation was recently addressed with a new highly customizable authentication framework based on Java Authentication and Authorization Service (JAAS), extended with features that specifically address the requirements of managing user authenticity and privileges in a distributed application server environment.

This new framework defined for JAAS support in WebSphere Application Server includes:

- well defined interfaces for altering the user subject
- enhanced Trusted Association Interceptor (TAI) support
- explicit documentation for the WebSphere Application Server login process
- the ability to assert complete user credentials to WebSphere Application Server (including group information)
- replication of subjects in a distributed environment
- plus a number of beneficial extensions to the security programming model.

This article will describe these new features in detail. The next sections provide some background on the fundamentals of the WebSphere Application Server authentication process and an overview of JAAS. The rest of the article will describe the most important aspects of these features as they relate to authentication.

Authentication Overview

First, let's look at the way in which WebSphere Application Server supports authentication:

During a WebSphere Application Server login, some information has to be provided to prove the user's authenticity. This can be information like a user ID and password, an X509 certificate from an SSL session, or a single sign-on token from a browser. Once this information is either authenticated or validated (credentials are authenticated; tokens are validated), a JAAS subject is generated with a WebSphere Application Server credential and principal.

The WebSphere Application Server credential is a WSCredential implementation in the public credential set of the subject.

The WebSphere Application Server principal is a WSPrincipal implementation in the principal set of the subject.

The custom types are, of course, created to contain custom information that WebSphere Application Server uses to track user identity.

As resources are being accessed on the login thread, the subject is used in the server where the authentication took place to make authorization decisions.

Besides authentication, which requires a requesting entity to provide proof of its identity, WebSphere Application Server also supports identity assertion. This is a relaxed form of authentication that doesn't actually require proof of identity, but rather accepts the identity based on a trust relationship with the entity that vouches for the asserted identity.

Figure 1 illustrates the authentication methods available in WebSphere Application Server, in which we distinguish between two main authentication cases:

- Web client authentication.
- EJB client authentication.

Web Authentication

The most common way for Web clients to authenticate is by providing a user ID and password that can be accepted using either HTTP basic authentication or form-based

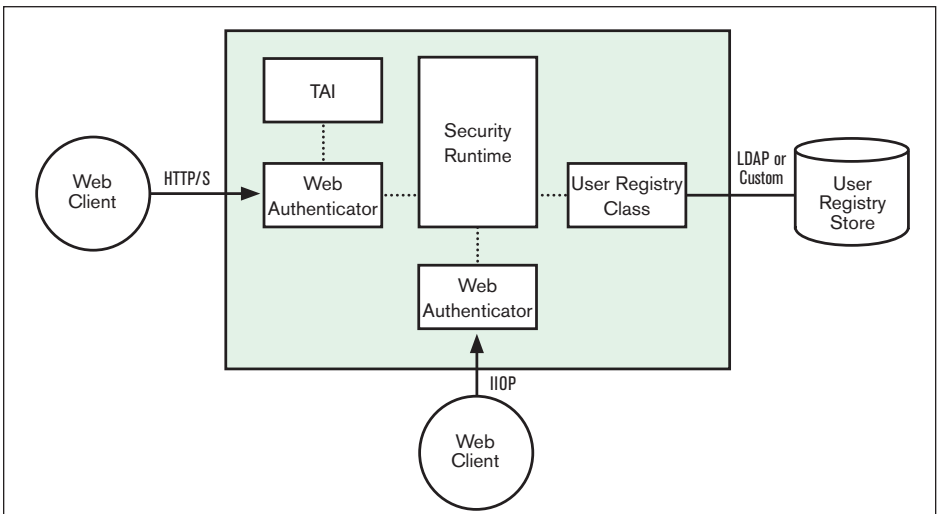


Figure 1: WebSphere Application Server authentication methods

authentication. WebSphere Application Server takes this information, looks up the user's unique ID (for example, a DN for LDAP) in the registry, and then verifies the password against the registry. In the case of LDAP, an ldap_bind is performed.

Web clients can also authenticate using client certificates. As with any SSL system, client certificate authentication is done at the termination of the SSL connection. Thus the Web server – rather than WebSphere Application Server – is responsible for performing the client certificate authentication. When the certificate authentication is complete, the WebSphere Application Server Web server plug-in forwards the client certificate information to the application server, which then extracts information from the certificate and looks up the user in the registry (so this is actually an identity assertion from the Web server to WebSphere Application Server). Be aware that when the Web server does client certificate authentication, the Web server plug-in simply asserts that information from the Web server to the application server. Therefore, the connection from the Web server to the WebSphere Application Server should be protected by mutually authenticated SSL. Otherwise, any client with direct access to the application server Web container can easily forge certificate information and compromise security.

The information used for the registry lookup is customizable and can be made totally flexible if a custom registry is developed using the WebSphere Application Server custom registry interface. Once authentication has occurred, a single sign-on (SSO) token is created and sent back to the browser as a cookie – this is equivalent to the LTPA token from prior releases of WebSphere Application Server – and the security credential is cached by the security runtime.

EJB Client Authentication

EJB clients can authenticate using passwords or certificates. In the case of password-based authentication, the client runtime is responsible for obtaining the user ID and password and sending them to the server to be verified against the registry. In either case, if the authentication is determined to be valid, a CSIv2 session is established and used for future requests. As with Web client authentication, a JAAS subject is created. Unlike Web clients, the subject is associated with the CSIv2 session rather than being placed in the security cache.

By default, the WebSphere Application Server client runtime prompts for user ID and password (if they are needed) using a graphical dialog box. This behavior can be controled by editing the sas.client.props file in which you can even specify a user ID and password. However, after obtaining the user ID and password it's recommended that clients use the JAAS login APIs to authenticate in some appropriate way under application control.

Trust Association Interceptor

HTTP clients can also pass identity information to WebSphere Application Server by using a trust association interceptor (TAI). The TAI interface provides a mechanism by which WebSphere Application Server enables an external component to authenticate the user and then assert the identity to the WebSphere Application Server Web container.

You can develop a custom TAI or use one of several that are already commercially available. These TAIs are typically used in conjunction with a Web authentication proxy server, such as IBM's Tivoli Access Manager or Netegrity's SiteMinder. These products authenticate the user and then simply inform WebSphere Application Server as to the end-user's identity. Typically this is done by the proxy server sending the user's ID and some additional verifiable information to the application server.

The TAI extracts this information and then returns the user's ID (or optionally a subject) to WebSphere Application Server, which then queries the registry as it normally would, but doesn't validate the user's password. (If the user ID isn't found in the registry, the assertion will, of course, fail.) This provides a powerful mechanism for enabling WebSphere Application Server to participate in a single sign-on domain.

Advanced TAIs can assert more than just the user's ID. In fact, they can assert an entire identity including all the group memberships required to support authorization. In such a case, WebSphere Application Server won't have to query the user registry.

Internal Authentication

Application code, as well as WebSphere Application Server itself, can also authenticate from within the process, essentially creating an authenticated subject on-the-fly. To do this, the standard JAAS login APIs are used following the same approach used in other scenarios: the provided user ID and password are validated against the registry and if validation is successful, a JAAS subject and the authentication token are created. Though it may not be obvious, this implies that when WebSphere Application Server servers authenticate themselves, they use the same registry for authentication as user-level authentication.

JAAS Overview

This section provides some basic introductory material on JAAS concepts, specifically the way in which login modules are used to provide pluggable authentication. Readers should become familiar with the JAAS specification.

Introduction to JAAS

WebSphere Application Server makes heavy use of the JAAS programming model. JAAS is a standard Java framework for performing a number of security-related tasks, including login, custom authentication, and (through Java 2 extensions) authorization. WebSphere Application Server supports the use of JAAS for login and customized authentication (with the restrictions stated in this article). JAAS exposes an application-level programming interface (API) for use by applications and a service programming interface (SPI) for the providers of its functionality. This model accomplishes the goal of insulating applications from service providers, thereby enabling portability across system platforms. Furthermore, JAAS follows the Pluggable Authentication Module (PAM) authentication model and so the service provider is completely pluggable through administrative configuration procedures.

Authentication sometimes involves more than one authentication system. To support this, JAAS isn't only pluggable, it's also stackable. This means a sequence of one or more mechanisms can be configured to drive the authentication process. Figure 2 is a high-level representation of the JAAS model.

JAAS uses the concept of a subject to define a user. A subject is created at initial authentication time, and is really just a container for user information; it includes principal and credential data. Authentication is performed by login modules that are grouped into login configurations, providing the stackable authentication mechanism described above. Login modules have the ability to populate data into the subject. In WebSphere Application Server, this will include WSPrincipal and WSCredential objects. WSPrincipal is basically a Java principal, which is used to define an entity in Java such as a user, organization, or login ID. The WSCredential defines security information that will be used for authorization, such as group memberships.

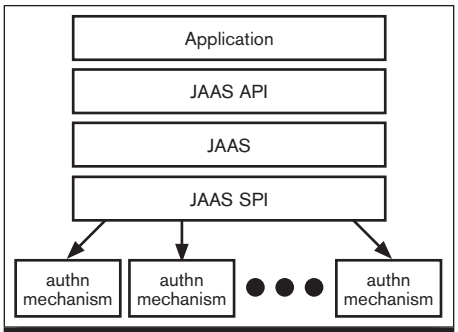


Figure 2: The JAAS authentication framework

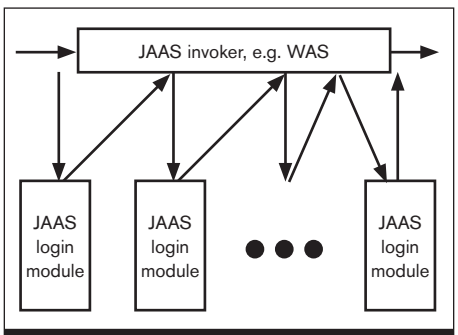


Figure 3: The JAAS invocation model

JAAS Control Flow

As discussed above, the JAAS login process provides access to a series of login modules orchestrated into a login configuration. The JAAS invocation model consists of the caller (for example, a WebSphere Application Server container runtime) alternately calling JAAS components until all configured modules are invoked or a failure is encountered. Figure 3 shows this seesaw-like pattern of alternating control flow between the invoker and the JAAS modules.

The effect of each JAAS login module is driven by a configuration attribute with one of following values:

- ***Required:*** The LoginModule is required to succeed. Irrespective of whether it succeeds or fails authentication still continues with the subsequent LoginModule list.
- ***Requisite:*** The LoginModule is required to succeed. If it succeeds, authentication continues down the LoginModule list. If it fails, control immediately returns to the application (authentication doesn't proceed down the LoginModule list).
- ***Sufficient:*** The LoginModule isn't required to succeed. If it does succeed, control immediately returns to the application (authentication doesn't proceed down the LoginModule list). If it fails, authentication continues down the LoginModule list.
- ***Optional:*** The LoginModule isn't required to succeed. If it succeeds or fails, authentication still continues to proceed down the LoginModule list.

(By default, WebSphere Application Server login modules have a required semantic.)

We will now expand on the flow of Figure 3 by showing the calling sequence for JAAS modules and then discuss the data constructs exchanged via the controller/invoker. This will involve manipulating the core JAAS objects, which are:

- Subject
- LoginContext
- LoginModule

The Subject and the LoginContext objects are visible to the JAAS invoker while the LoginModule isn't; that is, it's subject to configuration only. To authenticate an entity (which we'll call the subject) the following steps are performed:

- ***An application instantiates a LoginContext.*** The LoginContext consults a login configuration to load all of the login modules that are part of that configuration.
- ***The application invokes the LoginContext's login method.*** The login method invokes the loaded login modules in accordance with the semantics mentioned above. Each login module attempts to authenticate the subject. Upon success, login modules associate relevant principals and credentials with a Subject object that represents the subject being authenticated.
- ***The LoginContext returns the authentication status to the application.*** Upon success, the application retrieves the subject from the LoginContext.

Refer to the JAAS Reference Guide for detailed constructor and method signatures of the Subject, LoginContext, and LoginModule classes.

Data Flow Across Login Module Invocations

As described above, authentication of the subject in the JAAS programming model is done by instantiating a separate LoginContext object and its corresponding LoginModule objects. These two constructs encapsulate authentication data for the subject being authenticated.

Before each login module is invoked, it's initialized with the subject, a shared login module state, login module-specific options, and possibly a CallbackHandler. With successful authentication, the subject is updated with relevant credentials. Data encapsulated by the subject therefore represents the first construct passed across configured login modules, while the shared state represents the second such data element. These two objects allow for the effect of one login module to become visible to the next module in the list.

The optional CallbackHandler is used by the login module to gather authentication information about the environment and the user, including an identity and a proof of possession of that identity. It's possible for this process to involve an interaction with the end user; however, this isn't feasible for server-side callbacks. On the server side, this information is obtained from an incoming client request or the environment itself. Callback handlers for WebSphere Application Server are discussed in the JAAS usage section.

Authentication Plug Points

The WebSphere Application Server authentication process is fully pluggable. By providing plug points for custom code at most key steps, it's possible to heavily customize the WebSphere Application Server authentication process. Code can be developed to add custom information to a subject, require additional authentication information as part of a login process, or even bypass the normal registry use by asserting complete user credentials to WebSphere Application Server.

There are two ways in which these extensions to the WebSphere Application Server authentication model can be achieved:

First, most of the authentication process is built around JAAS login modules, and so it's possible to plug in custom login modules before, after, or between the login modules provided by IBM (however, the IBM modules mustn't be removed).

Second, the TAI interface enables the assertion of complete credentials for Web-based authentication.

WebSphere Application Server provides a set of standard login configurations with login modules and callbacks that are used in various situations to achieve authentication. These modules and callback handlers are defined and available depending on the specific authentication situation. The major configurations that will support most needs are for Web inbound, RMI inbound, and RMI outbound scenarios. We'll discuss each in some detail, but first we'll look at the standard WebSphere Application Server login modules.

Standard Login Modules

Of interest to us here are three login modules that are pre-defined in WebSphere Application Server for use with WEB_INBOUND, RMI_INBOUND, DEFAULT, and RMI_OUTBOUND configurations:

- ***com.ibm.ws.security.server.lm.ltpaLoginModule***
  Creates credentials, either using a user-provided hashtable or default registry access.
- ***com.ibm.ws.security.server.lm.wsMapDefaultInboundLoginModule***
  Instantiates a subject using either credentials or tokens depending on the login type.
- ***com.ibm.ws.security.server.lm.wsMapCSIv2OutboundLoginModule***
  Converts subject into tokens for outbound transmission over IIOP; this is used only in RMI_OUTBOUND.

You can put your custom login modules either before or after the IBM login modules; where you put them will depend on what you're trying to achieve:

If you're looking to assert identity information and bypass the standard WebSphere Application Server login module behavior of authenticating against a user registry, then you must put your login module before the ltpaLoginModule.

However, if your login module has to use information created by the WebSphere Application Server login modules (for example, you need to see the authenticated user to add extra information to the subject), then you would place your login module after the IBM modules.

It's also possible to put your custom module between the two IBM login modules, but that's rarely required. One such case can be if you write an RMI_OUTBOUND login module, when it would be crucial that your login module execute before the IBM wsMapCSIv2OutboundLoginModule.

System Login Configurations

WebSphere Application Server now defines four system login configurations that are used in specific situations related to security:

- WEB_INBOUND
- RMI_INBOUND
- RMI_OUTBOUND
- DEFAULT

(There are also application login configurations that are used for other purposes, such as J2EE Connector authentication, which won't be discussed here.)

WEB_INBOUND

Figure 4 shows a simplified view of the authentication flow for the WEB_INBOUND configuration.

As its name implies, this configuration is intended to be used to authenticate Web-based (HTTP) traffic. There are three basic scenarios:

- If a user has already authenticated (meaning an SSO token is present in the request) and the subject is available in the local security cache, then the TAI and all login modules are bypassed and control is passed directly into the Web container.
- If an SSO token is available but the subject isn't in the local security cache, WebSphere Application Server will attempt to obtain the user's subject from another server (we'll explain how later). If this succeeds, then what's called a propagation login is done. (Propagation is discussed in a later section.) The login modules will

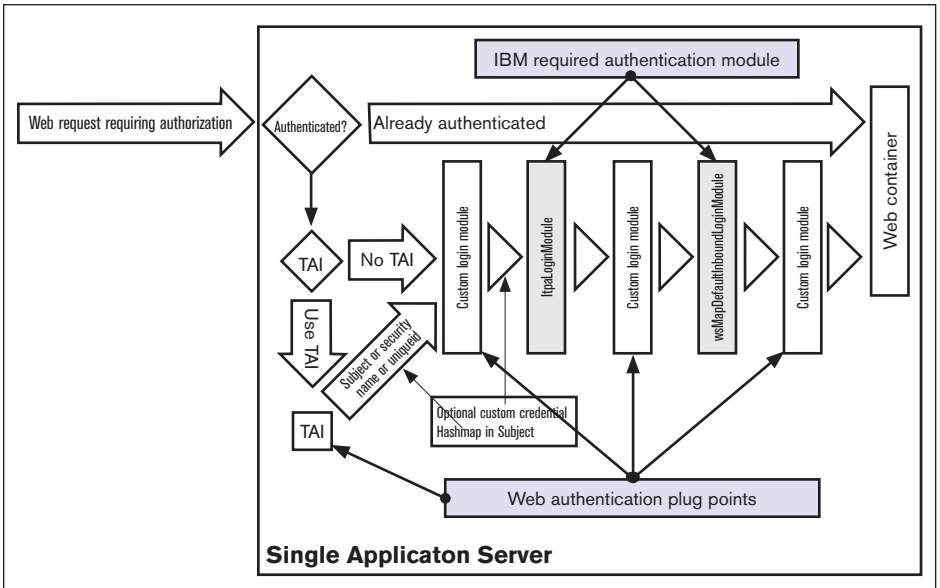


Figure 4: Web inbound flow

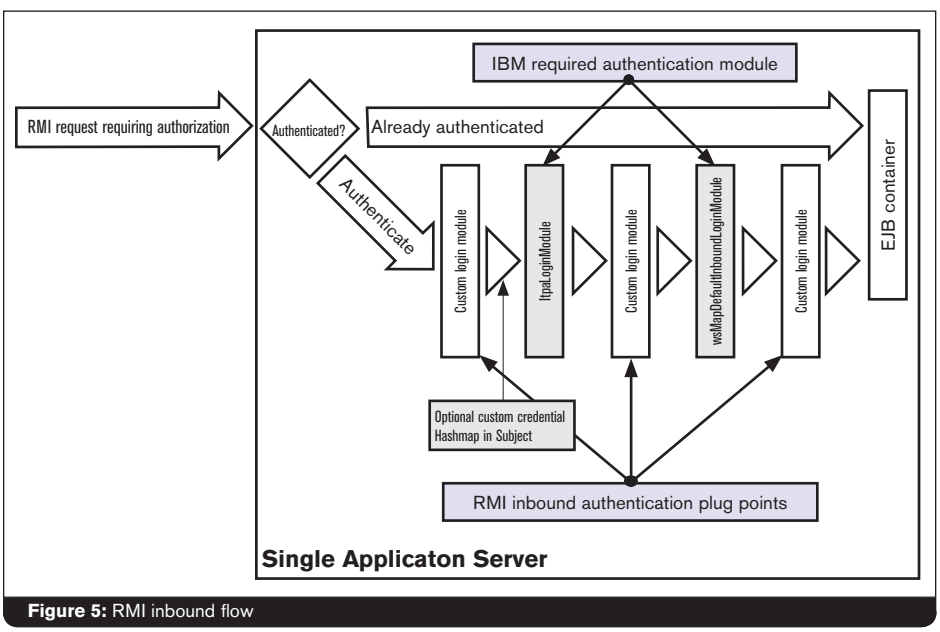


Figure 5: RMI inbound flow

be called and are expected to validate the subject. If WebSphere Application Server is unable to obtain the subject, then things get tricky. By default, WebSphere Application Server will do an initial login (in the next bullet item), but this behavior can be altered.

- If no SSO token is available, then an initial login is done. First, the TAI will be invoked if it exists. After the TAI is invoked, the login modules will be invoked. The TAI can create a subject for use by the login modules.

When the login modules are invoked, both the required login modules and any custom login modules will be invoked. The sequence in which they are invoked is determined administratively via the login configuration. Of course, you can decide to write a custom TAI and custom login modules, but this is unlikely to be a desirable option.

As you look at the Figure 4, notice that the TAI can provide a custom subject to the login modules; custom login modules can also customize the subject. Shortly, we will go into more detail about what login modules and the TAI can do.

RMI_INBOUND

Figure 5 shows a simplified view of the authentication flow in the RMI_INBOUND configuration.

As its name implies, this configuration is intended to be used to authenticate RMI/IIOP requests; that is, requests that bypass the Web container and invoke an EJB directly. These could be requests from a standalone EJB client or from another application server.

There are three basic scenarios:

- If a user has already been authenticated, and so a CSIv2 session established, all login modules are bypassed and control is passed directly to the EJB container.
- If the user already has an authenticated context at the caller, the stack of login modules in the configuration will be executed in a propagation login (the difference between an initial login and a propagation login will be explained later) as the user authentication data (known as tokens) are propagated from one server to the next. Both the required WebSphere Application Server login modules, as well as any custom login modules added to the configuration, will be invoked.
- If the user hasn't authenticated, the login modules will be invoked just as before, but this time in an initial login mode. The login modules are responsible for verifying the user's authentication data and creating the subject.

RMI_OUTBOUND

Figure 6 shows how the authentication flow occurs in the RMI_OUTBOUND configuration.

As its name implies, this configuration is intended to be used for special authentication-related processing on a call out to a remote EJB container, which typically involves some type of credential mapping processing. You can do identity assertion to a remote server without needing a login

module, but if the remote server doesn't share a registry with the authenticating server, you may need to use a login module to map the existing user identity information to a credential that's valid for that server.

If a CSIv2 session is already in place, the remote container is called directly and the login modules aren't executed. Otherwise, the stack of login modules in the configuration will be executed. This will include both the required WebSphere Application Server login module, as well as any custom login modules added to the configuration.

This configuration won't be discussed any further in this article.

Other Configurations

There are other defined login configurations in WebSphere Application Server. The DEFAULT login configuration handles situations when none of the above configurations apply; these include SOAP requests from the admin client and JMX admin authentication requests. There might also be legacy login configurations, such as SWAM, LTPA and LTPA_WEB. However, these should no longer be used as of WebSphere Application Server V5.1.1.

Additionally, there are two login configurations related to Web Services, wssecurity.signature and wssecurity.IDAssertion that this article won't cover.

TAI Use

Understanding JAAS and the basics of WebSphere Application Server authentication, we'll now discuss how to customize the WebSphere Application Server authentication process in more detail. We'll begin with custom TAIs, since these are the easiest to understand and use – and are also the most commonly used since TAIs support Web-based single sign-on.

TAI: Asserting Identity Information to WebSphere Application Server

If the user has already been authenticated by some authentication system other than WebSphere Application Server, it's possible to inform WebSphere Application Server of the user's identity information rather than requiring the user to re-authenticate. This is known as identity assertion.

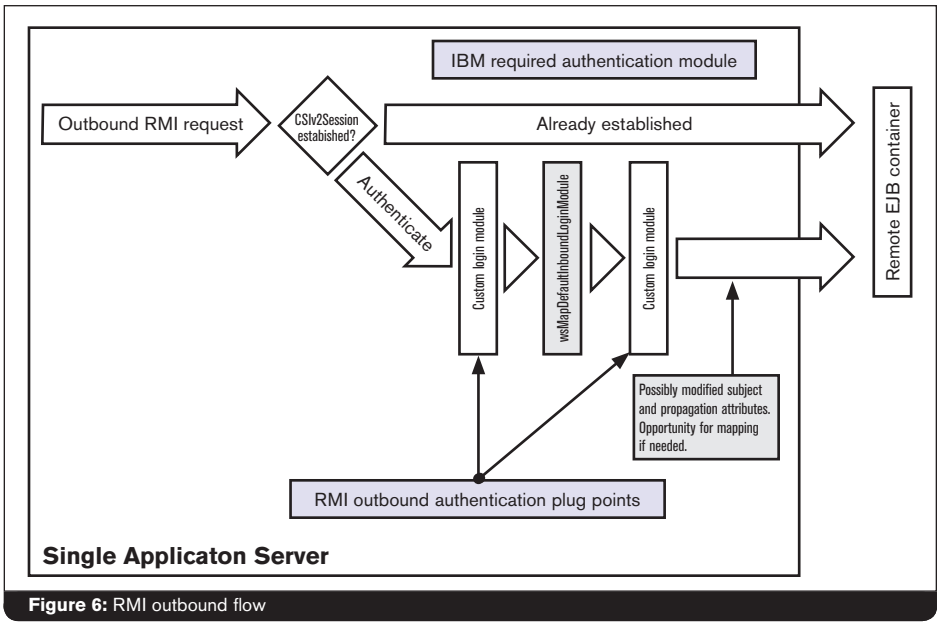


Figure 6: RMI outbound flow

WebSphere Application Server has long supported Web-based identity assertion as part of the TAI interface. As of WebSphere Application Server V5.1.1 there are two TAI interfaces. The legacy TAI method of returning the user ID to WebSphere Application Server is retained, but a new, more powerful method is now available. Custom-developed code can assert not just the user's ID but rather the entire user identity (user ID, unique ID, and group information). If this is done, WebSphere Application Server won't contact the UserRegistry to derive additional user information. This may improve performance in situations where the user's group information is already known, but more importantly, this gives applications the flexibility to determine a user's group memberships by criteria other than the static information in the UserRegistry.

An additional new feature in the TAI is the support for a multi-phase negotiation process, which enables the TAI to request additional information from the client that is asserting identity.

The advanced TAI interface that supports Subject assertion is com.ibm.wsspi.security.tai.TrustAssociationInterceptor. Note that this is different from the older (and still supported) com.ibm.websphere.security.TrustAssociationInterceptor.

As mentioned above, this advanced TAI interface supports a multi-phase negotiated authentication process (for example, some systems require a challenge response protocol back to the client). The two key methods in this new interface are:

- ***public boolean isTargetInterceptor(HttpServletRequest req)***
  This method will return true if this TAI should handle this request; false tells WebSphere Application Server to ignore this TAI.
- ***public TAIResult negotiateValidateandEstablishTrust(HttpServletRequest req, HttpServletResponse res)***
  This method returns a TAIResult object that indicates the status of the request being processed. The HTTP response object can be modified if needed.

The TAIResult class has three static methods for creating a TAIResult, all of which always take an int as the first parameter. This parameter is expected to be a valid HTTP request return code and is interpreted in one of two ways:

The value HttpServletResponse.SC_OK tells WebSphere Application Server that the TAI has completed its negotiation. WebSphere Application Server will then use the information in the TAIResult to create a user identity.

Any other value indicates that WebSphere Application Server should return the TAI output (which has been placed into the HttpServletResponse) and return it to the Web client (or more likely to an authenticating proxy). Typically, this will result in the Web client providing additional information on a future request and making another call to the TAI.

The created TAIResults have the following meanings:

- ***public static TAIResult create(int status);*** Indicates only a status to WebSphere Application Server. The status shouldn't be SC_OK since no identity information is provided. This will typically be used to inform the asserting client that the TAI needs additional information. Most likely, the TAI has modified the HttpServletResponse object to send some useful information back to the caller.
- ***public static TAIResult create(int status, String principal);*** Indicates a status and the user ID or unique ID for this user. WebSphere Application Server will create credentials by querying the registry for all required user information. The status must be SC_OK.
- ***public static TAIResult create(int status, String principal, Subject subject);*** Indicates a status, the user ID or unique ID for this user, and a custom subject. If the subject contains a complete user hashtable (see the next section on creating a custom subject), the principal will be ignored. The contents of the subject will become part of the eventual user subject. The status must be SC_OK.

There are a few additional methods on the TrustAssociationInterceptor interface that are discussed in the Java documentation. These are used for initialization, shutdown, and identifying the TAI to WebSphere Application Server.

Given this framework, a custom TAI can accomplish a number of different things. A number of simple examples follow. For clarity's sake examples don't show error handling.

TAI That Asserts a User Simply

Here is a trivial example of a negotiateValidateandEstablishTrust that asserts a user ID:

Listing 1

```
TAIResult negotiateValidateandEstablishTrust
  (HTTPServletRequest req, HTTPServletResponse
response) {

  String userid = //get from request somehow
  if ( didn't work )
        throw new WebTrustAssociationFailedExcep
tion("a reason why");
  return TAIResult.create(HTTPServletResponse.
SC_OK, userid);
}
```

In this case, only the user's user ID is provided. WebSphere Application Server will access the registry to retrieve all additional user information, such as group memberships.

To indicate authentication failure (and cause WebSphere Application Server to fail the user authentication) you must throw a WebTrustAssociationFailedException.

TAI That Negotiates

This example shows how to indicate that additional negotiation is required, and also how to return a user ID:

Listing 2

```
TAIResult negotiateValidateandEstablishTrust
  (HTTPServletRequest req, HTTPServletResponse
response) {

  //determines phase by examining request
  if (phase one) {
  ... Alter response object appropriately...

        //Indicate not done to WebSphere
Application Server. Return code is client specific
        return TAIResult.create
(HTTPServletResponse.SC_CONTINUE);
  } else {///done
  String userid = //get from request somehow
        return TAIResult.create(HTTPServletRespo
nse.SC_OK, userid);
  }
}
```

"In the past, IBM WebSphere Application Server had a rigid authentication model that made it challenging to support complex or unusual requirements. This situation was recently addressed"

Create a custom subject

WebSphere Application Server also provides the ability to create a custom subject. This can actually be achieved by user code either in the TAI or in a custom login module. Before we get into the details of how this is achieved in a TAI, we need to clarify that there are actually two very distinct things that can be described as "creating a custom Subject." The simpler meaning is merely the adding of custom attributes to a standard subject, created by WebSphere Application Server using its normal method.

One important thing to remember is that if you create custom subjects containing custom attributes, the code to create that subject probably needs to be on every application server. This is because any application server might create the subject as part of an authentication. More importantly, every application server potentially needs to deserialize the subject, and so in order for it to make sense, it has to have access to the subject's Java class definitions. If the target server does not have the custom classes, deserialization will fail and WebSphere Application Server will simply discard the custom information.

The second meaning of "custom subject" is the creation of a subject with custom credentials by providing a custom hashtable that contains user identity information. In this scenario, user code provides all of the identity information required by WebSphere Application Server to generate a subject.

With either type of custom subject, it is important to understand the implications in a clustered environment. This will be covered in detail later in the sections on JAAS usage and propagation, but the important thing to be aware of is that if the subject you create is modified in such a way that WebSphere Application Server cannot recreate it using its own default mechanisms, then you must add a custom cache key to the subject. This key is used during identity propagation to ensure that the correct subject is instantiated on another application server. This is equally true whether you create a custom subject with custom credentials, or merely add attributes to the standard subject created by WebSphere Application Server.

- Add information to a subject

The code below shows how to add custom attributes to a subject. WebSphere Application Server will still create a normal user credential, accessing the user registry as required, but will retain the custom attribute in the subject for future application use during this login session. The custom attribute in the example is added to the public credentials area of the subject (which is probably where you would normally add it). (For this to work in a distributed environment, the SomeType class must be serializable.)

Listing 3

```
TAIResult negotiateValidateandEstablishTrust
   (HTTPServletRequest req, HTTPServletRespose
response) {

   String userid = // get from request somehow

   SomeType somethingextra = new SomeType();
   ... do whatever to somethingextra ...
Subject subject = new Subject();
Subject.getPublicCredentials().add(somethingextra);

return TAIResult.create(HTTPServletResponse.SC_OK,
userid, subject);
}
```

- Create a custom subject with custom credentials

If you want to completely override the authentication process and complete a full custom subject with all required credentials, a hashtable of authentication data must be created and added to the subject, as in this sample code:

Listing 4

```
String userid = //get from request
InitialContext ctx = new InitialContext();
UserRegistry reg  =(UserRegistry)ctx.
lookup("UserRegistry");
String uniqueid = reg.getUniqueUserID(userid);

//define groups
ArrayList groups = new ArrayList();
// add admin group
groups.add(reg.getUniqueGroupId("Administrators"));

// stash in hashtable
Hashtable hashtable = new Hashtable();
hashtable.put(AttributeNameConstants.WSCREDENTIAL_
UNIQUEID,uniqueid);
hashtable.put(AttributeNameConstants.WSCREDENTIAL_
SECURITYNAME,userid);
hashtable.put(AttributeNameConstants.WSCREDENTIAL_
GROUPS,groups);
hashtable.put(AttributeNameConstants.WSCREDENTIAL_
CACHE_KEY,uniqueid+"MyCustom");

Subject subject = new Subject();
subject. getPublicCredentials().add(hashtable);
return TAIResult.create(HTTPServletResponse.SC_OK,
"ignored", subject);
```

In this example, we create our own custom array of groups for the user, containing the single element "Administrators". We add this to the hashtable of information for the user, which you can see we also update with the user ID and unique ID (obtained from the user registry). We also create a custom cache key to ensure that the subject will be recreated correctly on propagation.

As you can see from the sample above, WebSphere Application Server will expect to find specific key information in the hashtable in the subject. The keys are defined in com.ibm.wsspi.security.token.AttributeNameConstants and they are used as follows:

- ***Key: WSCREDENTIAL_UNIQUEID***
  Description: This should be a unique representation of the user. The best way to obtain a unique ID for a user is to call the WebSphere Application Server UserRegistry method: public String getUniqueUserId(String userSecurityName). This will also ensure compatibility with the WebSphere Application Server default implementation for the unique ID. If you need to customize the unique ID for some reason, the best way would be to write a Custom User Registry that was able to generate meaningful and appropriately unique IDs.
  Format: this is a java.lang.String.
  Expected format examples (realm/uniqueUserId):
  LDAP: "ldaphost.austin.ibm.com:389/cn=user,o=ibm,c=us"
  Windows: "MYWINHOST/S-1-5-21-963918322-163748893-4247568029-500"
  UNIX: "MYUNIXHOST/32"

- ***Key: WSCREDENTIAL_SECURITYNAME***
  Description: This is the securityName (commonly called the user ID or short name) of the authenticated user. (WebSphere Application Server uses the securityName attribute for the getRemoteUser(), getUserPrincipal() and getCallerPrincipal() APIs.) The best way to ensure compatibility with the WebSphere Application Server default implementation for the securityName value is to call the WebSphere Application Server UserRegistry method: public String getUserSecurityName(String uniqueUserId).
  Format: this is a java.lang.String.
  Expected format examples:
  LDAP: "user" (ldap UID)
  Windows: "user" (Windows username)
  UNIX: "user" (UNIX username)
- ***Key: WSCREDENTIAL_GROUPS***
  Description: This is an ArrayList of realm qualified groups to which this user belongs. The format of these groups is important as they are used by the WebSphere Application Server authorization engine for group-to-role mapping in the deployment descriptor. The format provided must match what the WebSphere Application Server default implementation expects. If you use a third party authorization provider, then this should be whatever the third party provider expects. The best way to ensure compatibility with the WebSphere Application Server default implementation for the unique group ID value is to call the WebSphere Application Server UserRegistry method: public List getUniqueGroupIds(String uniqueUserId). If the HashMap is present in the subject created by the TAI but it contains no groups, then WebSphere Application Server will create a credential with the user having no groups; it will not query the registry to obtain group information in this scenario.
  Format: this is a java.util.ArrayList of java.lang.String.
  Expected format examples for each group in the ArrayList:
  LDAP: "ldap1.austin.ibm.com:389/cn=group1,o=ibm,c=us"
  Windows: "MYWINREALM/S-1-5-32-544"
  UNIX: "MY/S-1-5-32-544"

In addition to the three keys listed above, you may also need to specify a fourth key, the cache key, which is a unique identifier further defining the uniqueness of these credentials. The cache key defines how WebSphere Application Server internally caches subject information. However, it is important to understand that if you let WebSphere Application Server use the default cache key, it will recreate a default subject as needed, which means that customization of the subject could be lost.

This point is crucial. If you are providing only information that could also be directly derived from the UserRegistry, there is no need to provide the unique cache key. WebSphere Application Server will cache the information provided using the user's unique ID, and if the information must be recreated, then WebSphere Application Server will simply query the registry. However, if the information you are providing is not derivable from the UserRegistry, or is perhaps unique to this particular authentication session, then a unique cache key must be provided. The key must be appropriately unique; that is, it must at least be globally unique across all users. For example, if the subject will always contain the same information for the same user, the user's unique ID plus a simple hard coded constant value is sufficient (as we've done in our example above). On the other hand, if the information in the subject is unique to this specific login session (perhaps it contains the login time or login source), then the cache key must be dynamically generated (again including the user's unique ID) and should be unique across all users and instances of your custom login module or TAI. The cache key definition is:

- ***Key: WSCREDENTIAL_CACHE_KEY***
  Format: this is a java.lang.String

TAI scenario and example

The following code shows an example TAI that incorporates some of the concepts that we have discussed so far. This TAI accepts three parameters, previously set by an unsecured Web login application: the user ID and password of the requestor, plus a flag called AdminPriv, which enables a user who would normally be given administrator privileges (that is, they are a member of the admin registry group) to either accept or decline those privileges. This is a somewhat unrealistic example, but it serves the purpose of showing how to override registry attributes by creating a custom subject.

The TAI will verify the user's password using the UserRegistry, obtain the user's groups, and also verify that the user is actually a member of the admin group if they request admin privileges. If the user does not request admin access and the user is a member of the admin group, the TAI will remove that group from the group list and pass it to WebSphere Application Server. In this case, a custom key must be created because we have modified the subject based on some dynamic, non-registry driven information (in this case the user's preference to be admin or not for this particular session). In all other cases, the TAI will just use the default group list from the user's registry entry, so a custom cache key is not needed (the subject will contain only default information).

TAI installation and configuration

As discussed earlier, to develop a custom TAI you need to implement the TrustAssociationInterceptor interface defined in the com.ibm.wsspi.security.tai package. The WebSphere Application Server library in which this interface is defined is in the JAR file wssec.jar. The implementing JAR file that contains your custom TAI (we have provided loginexamples.jar for this purpose) should be deployed in the WebSphere Application Server environment in a location that is accessible by the security portions of the application server runtime. Your custom TAI implementation should be deployed under the WAS-INSTALL/lib/ext directory for the application server nodes because the entire WebSphere Application Server runtime can access code here. You may encounter problems if you try to place your TAI under a shared library for just the application server.

Assuming that you have enabled Global Security for your server, follow these steps to make sure you properly configure your custom TAI:

1. Install your TAI JAR in WAS-INSTALL/lib/ext. For our example, we simply copied the loginexamples.jar file to this directory for our WebSphere Application Server V6 install.

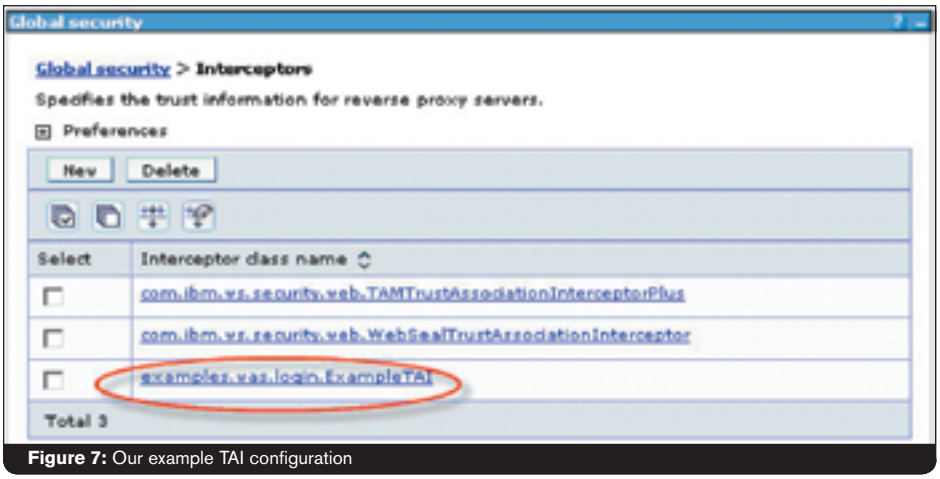


Figure 7: Our example TAI configuration

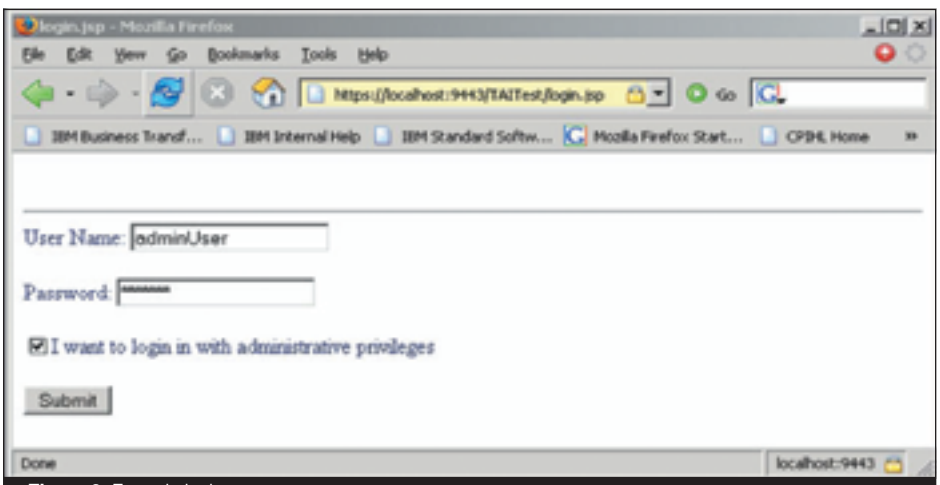


Figure 8: Example login page

2. From the WebSphere Application Server administrative console, navigate to Security => Global Security and ensure that the Active authentication mechanism dropdown is set to Lightweight Third Party Authentication (LTPA).
3. Verify that LTPA is configured for use on your server by selecting Security => Global security => Authentication mechanisms => LTPA and you should see the Password attribute already filled in.
4. Select Security => Global security => Authentication mechanisms => LTPA => Trust association and check the Trust Association Enabled box, then Apply.
5. Select Security => Global security => Authentication mechanisms => LTPA => Trust association => Interceptors. Click on New. Enter the fully qualified classname to your custom TAI class, then Apply. (Figure 7)
6. Our sample TAI does not depend on any custom properties. However, if yours does, then select Security => Global Security => Authentication Mechanisms => LTPA => Trust Association => Interceptors => yourTAIclass => Custom Properties => New to enter the (key, value) pairs for the properties on which yourTAI depends, then Apply.
7. Save your configuration and then restart your server to make your TAI fully operational.

Running the sample TAI

After you complete the TAI configuration and install the TAITestEAR.ear file (which consists of a single WAR file that contains a login JSP and a Servlet called PrintUserInfo), open a browser and navigate to the login.jsp page, as shown in Figure 8.

Our example behaves as follows:

- Case 1
  If we log in with a user that has administrative privileges (that is, the user is a member of the admin group in the user registry), and that user checks the box stating that they desire to log in with those privileges, the following code in our TAI is executed:

Listing 5

```
//determine if user wants to be admin.
String adminPriv = req.getParameter("AdminPriv");
if (adminPriv != null && adminPriv.equals("Y")) {
   System.out.println("User desires admin");
   wantsAdmin = true;
}

//go through groups and remove admin group if
needed.
Iterator iter = groups.iterator();
boolean foundAdmin = false;
while (iter.hasNext()) {
   String gid = (String) iter.next();
   if (gid.equals("admin")) {
           foundAdmin = true;
           if (wantsAdmin == false) {
                     iter.remove();
                     customGroups = true;
           }
           break;
   }
}

//Now, a quick error check (wanting admin when not
an admin)
if ((!foundAdmin) && wantsAdmin) {
   ... Error code not shown ...
}

String key;
if (customGroups) {
   key = uniqueid + "ExampleTAIAdminRemoved";
} else {
   key = uniqueid;
}

Subject subject = createSubject(userid, uniqueid,
   convertGroupsToUniqueIds(reg, groups), key);
return TAIResult.create(HttpServletResponse.SC_OK,
"notused", subject);
```

Manage Costs While Gaining Benefits

WAS for z/OS and zAAP

BY LINFENG YU

Running applications in WAS for z/OS lets you take advantage of the z/OS built-in Quality of Service features such as reliability, availability, scalability, and serviceability. However, the solution could be very expensive.

J2EE applications and WAS for z/OS are very CPU-intensive workloads on z/OS. They consume a lot more CPU cycles than the traditional workloads running on z/OS, especially when processing big XML files.

Normally the CPU utilization of an application running on z/OS is measured in MIPS. An application consuming more CPU cycles means that the MIPS number for the application is higher. To provide more MIPS to run your application, more CPUs are needed. However, adding more CPUs to a zSeries server causes virtually all software license fees to go up because of the zSeries's software license model, which by the machine's CPU horsepower.

This is not a new story. Companies have been using different ways to manage the software cost on the zSeries platform for years. When the J2EE applications start running in WAS for z/OS, it's harder to manage the software cost than before.

To reduce the overall cost of enabling Java on zSeries platform, IBM introduced zAAP for Java workloads on z/OS. The following sections describe what a zAAP is, how it works, and how to use it.

What Is zAAP?

zAAP stands for zSeries Application Assist Processor, also known as the Integrated Facility for Application processor (IFA). It's a specialized processing unit (PU a k a CPU on open systems) available on the zSeries 990 (z990), 890 (z890), and z9. It provides a strategic z/OS Java execution environment for customers who want the powerful integration advantages and traditional QoS of the zSeries platform.

Conceptually, zAAP is just a co-processor like your old PC's floating-point co-processor. Instead of working as a standalone general processing unit (CP), it only assists the general-purpose CPs to execute Java programming under the control of the IBM JVM. For this reason, zAAP's capacity doesn't incur IBM or third-party software charges. So you can buy additional processing power exclusively for Java application execution without affecting the machine model designation that's used to determine zSeries software cost.

zAAP has been designed to operate asynchronously with general CPs to execute Java programming under control of the Java Virtual Machine (JVM). Executing IBM JVM processing cycles on a zAAP is a function of the IBM Software Developer Kit (SDK) for z/OS Technology Edition V1.4, z/OS V1R6, and the Processor Resource/System Manager (PR/SM).

Figure 1 is a z/OS Logical Partition with zAAP. One zAAP can be configured per general processor in a Central Electronic Complex (CEC).

zAAP is enabled by IBM's innovative zSeries PR/SM virtualization technology. It can be virtualized into logical zAAPs and assigned to different LPARs. But zAAPs and general CPs should exist in the same z/OS LPAR. On z990s or z890s, zAAPs are grouped in the ICF/IFL/zAAP processor pool. The ICF/IFL/zAAP processor pool appears on the hardware console as ICF processors. The number of ICFs shown is the sum of IFL, ICF, and zAAP processors characterized on the server.

To exploit a zAAP, the operating system must be migrated to the following levels of software:

- z/OS V1R6
- IBM JDK V1.4 with a PTF for APAR PQ 86689
- For WAS for z/OS Java workloads, WAS for z/OS version 5.1 above

WAS for z/OS version 5.1 and above provide support for IBM's JDK 1.4. It makes

ABOUT THE AUTHOR

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

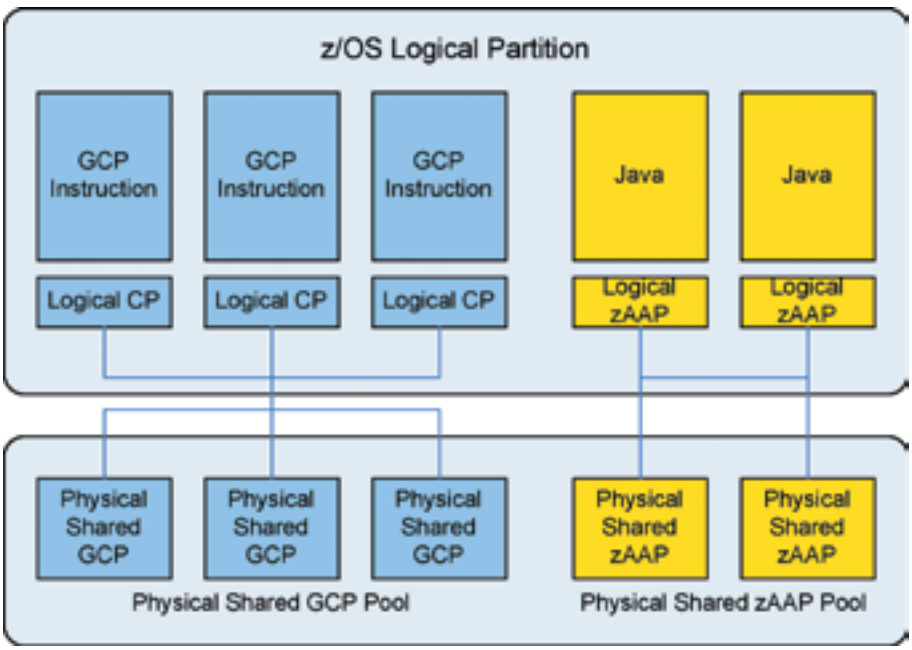


Figure 1: z/OS logical partition with zAAP

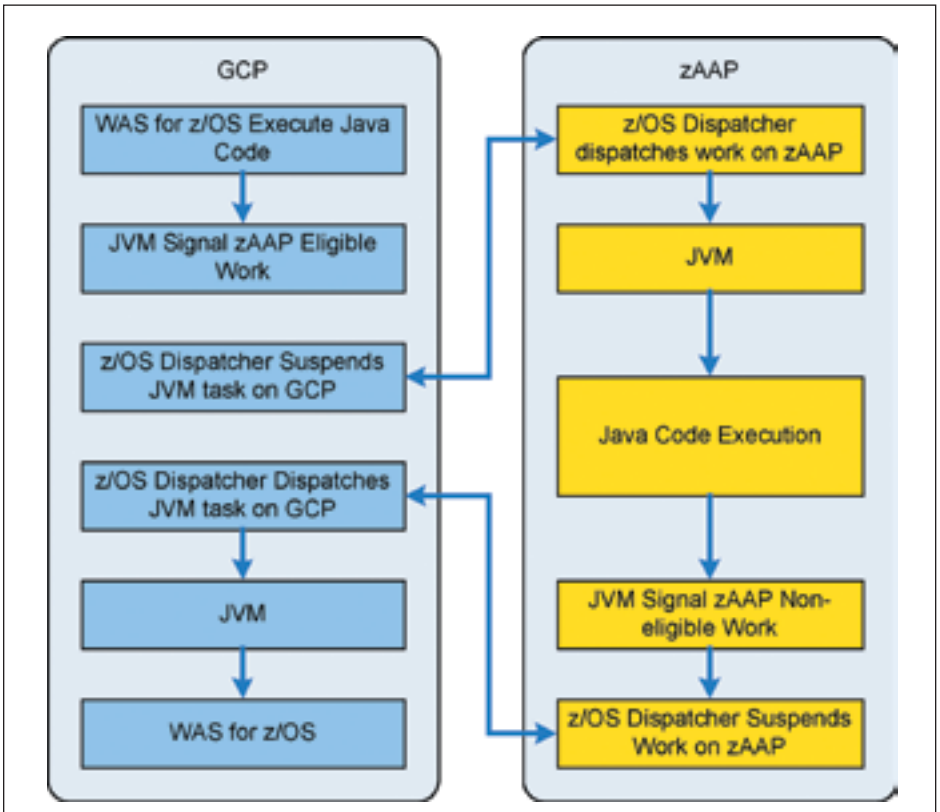


Figure 2: zAAP workflow

WAS for z/OS one of the key workloads that can take advantage of zAAPs.

The z/OS exploitation of zAAP capabilities provides the following added values:

- Simplifies and reduces server infrastructures by integrating e-business Java Web applications next to mission-critical data for QoS.
- Maximizes the value of zSeries investment through increased system productivity, achieved by reducing the demands and capacity requirements on general CPs, which can be reallocated to other workloads.
- With WAS for z/OS, your application can exploit the z/OS Workload Manager (WLM), which can guarantee service levels for specific kinds of customers and workloads defined by business needs.

In summary, zAAP is a special PU on the zSeries server that the Java workload can be off-loaded to. The applications you have running in WAS for z/OS can still leverage the QoS features provided by z/OS.

How Does zAAP Work?

When a z/OS logical partition is configured, both CPs and zAAPs are necessary to support the planned Java and non-Java workloads. Normally a Web application running in WAS for z/OS consists of both Java and non-Java workloads.

Figure 2 is the zAAP workflow. It essentially explains how the zAAP works. The IBM JDK V1.4 JVM, parts of Language Environment (LE) runtime, and z/OS Supervisor are needed to support JVM execution on zAAP.

Some of the JVM tasks are dispatched to general CPs. Basically these tasks do the following:

- Determine if the program code is eligible to run on zAAP
- Signal the z/OS dispatcher of the zAAP work
- Handle the program code that's ineligible to run on zAAP

Other JVM tasks are dispatched to zAAPs. These tasks:

- Determine if the program code is eligible to run on zAAP
- Run the zAAP eligible program code
- Signal z/OS dispatcher of non-zAAP work

Whenever a Java unit of work is executed, it's initially dispatched on a general CP. Before the Java code gets executed in the JVM, the JVM determines if the work is eligible to run on the zAAP. If so, the JVM signals the dispatcher that the current unit of work is zAAP-eligible. Then the dispatcher puts the current unit of work in the zAAP dispatcher queue. When a zAAP processor becomes available, the dispatcher selects the highest-priority work from the zAAP work queue and dispatches it on the zAAP processor.

A zAAP-eligible unit of work can be executed on a zAAP. zAAP work inherits the dispatching priority from the execution on the general CP. When the JVM finishes Java code processing, it signals the dispatcher that the current unit of work is not zAAP-eligible any more. The unit of work release control puts it back in the general logical processor work queue.

If the application is a pure Java application, the entire application should be run on the zAAP. Unfortunately, most applications that run in WAS for z/OS use various native libraries implicitly. For example, the JDBC type II driver, MQ batch adapter, and CTG for CICS Access are all Java code wrapprd around native codes. WAS for z/OS itself has other native code to exploit the z/OS environment. So you might see the dispatcher switch the work back and forth between the zAAP and the general CP. You can see in Figure 3 the zAAP integration at work. Switching the works back and forth causes overhead. Using zAAP reduces the MIPS number on general CPs, but the total MIPS number is higher than before.

I wondered why IBM didn't let the native code called from the JNI run on the zAAP as well. I think that would boost the use of WAS for z/OS and push Java on z/OS. Maybe IBM thinks it would lose it licensing fees.

Exploiting zAAP

Before you decide to use zAAP, you should figure out if you'd really gain anything from it. zAAP has a fixed price of $125,000. The horsepower it provides doesn't incur software charges.

The first thing to figure out is how much of your workload is zAAP-eligible.

Currently the following z/OS subsystems might have zAAP-eligible workloads:

- WAS for z/OS 5.1
- IMS V7, V8 and V9
- CICS 2.3
- DB2 V7, V8

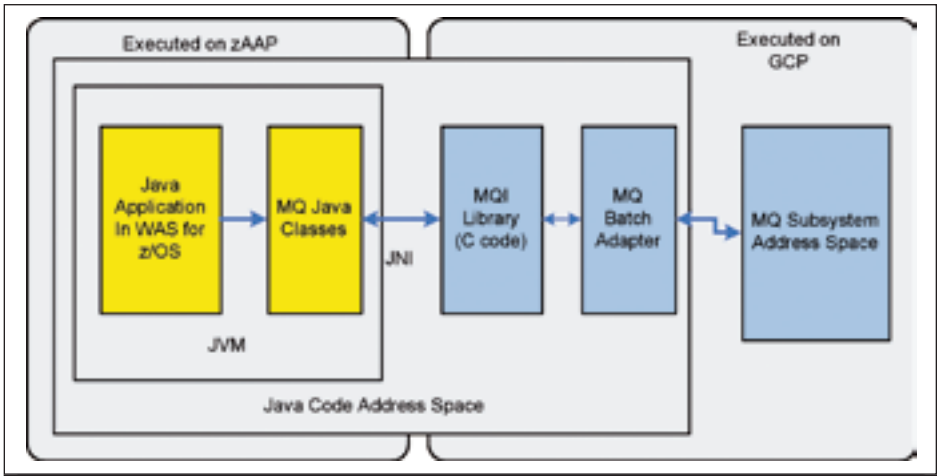


Figure 3: zAAP integration

If you're using WAS 5.1 for z/OS, it's definitely worth exploring zAAP. For the other subsystems, you might not have a Java workload. If you need more information, refer to the product documentation.

A measurement tool, the zAAP Projection Tool for Java 2 Technology Edition, helps estimate the zAAP-eligible content in your workload. It's included in IBM JDK 1.3.1 at the PTF level UQ94379 (S2.4) or later and IBM JDK 1.4 with the PTF (or later) for APAR PQ86689.

The Projection Tool prints out the information on processor time as messages in the STDERR file for each JVM every five minutes, along with statistics and total values at termination. The report consists of:

- ***Java IFA time*** – the time accumulated for Java threads processing zAAP-eligible work;
- ***Java Standard CPU time*** – the time accumulated for Java threads processing zAAP non-eligible work;
- ***Interval address space CPU time*** – the time for working in the address space across all dispatchable units including the Java threads.

Listing 1: Project Tool output

```
IFA Projection data for system id=<SYSE.341> Starting at:
16:25:07 - Current address space CPU: 0.025418 sec.
<SYSE.341> Interval at: 16:28:15 Switches To/From IFA: 17
 Java IFA: 0.411787 sec. Java Standard CPU 0.000059 sec.
 Total address space CPU: 3.266723 sec.
<SYSE.341> Interval at: 16:28:40 Switches To/From IFA: 77
 Java IFA: 3.421488 sec. Java Standard CPU 0.008685 sec.
 Total address space CPU: 0.583209 sec.
<SYSE.341> TOTAL at: 16:28:40 Switches To/From IFA: 94
 Java IFA: 3.833275 sec. Java Standard CPU 0.008744 sec.
 Total address space CPU: 3.849932 sec.
```

Listing 1 is the STDERR file containing the project tool output.

To analyze this data further, an Excel workbook is provided at http://www6.software.ibm.com/dl/zosjava2/zosjava2-p. It processes the STDERR file and stores the zAAP processing information in a spreadsheet. Figure 4 is a sample of the contents of the spreadsheet after executing in the workbook. This Figure is from IBM's technical document.

The workbook combines data from multiple JVMs and multiple address spaces, service classes, and LPARs, aligns data to RMF intervals, adjusts zAAP utilization factoring in the z/OA capture ratios, and expresses zAAP and standard CP time as a percent of the single CP that the data was collected on.

Based on the information provided by the Projection Tool, you can work with your zSeries capacity planning staff to determine how much you can gain from using zAAP and how many zAAPs you should use. I don't want to get into too much detail on this because it involves a lot of factors regarding zSeries capacity planning. Normally Java developers have no interest in it.

Reach Over 100,000

Enterprise Development Managers & Decision Makers with...

Offering leading software, services, and hardware vendors an opportunity to ***speak to over 100,000 purchasing decision makers*** *about their products, the enterprise IT marketplace, and emerging trends critical to developers, programmers, and IT management*

Don't Miss Your Opportunity to Be a Part of the Next Issue!

Get Listed as a Top 20* Solutions Provider

For Advertising Details Call 201 802-3021 Today!

*ONLY 20 ADVERTISERS WILL BE DISPLAYED. FIRST COME FIRST SERVE.

SYS-CON MEDIA

The World's Leading i-Technology Publisher

Using the Projection Tool with WAS for z/OS is quite straightforward. The following steps apply to different versions of WAS for z/OS:

- ***Step1:*** Upgrade the JVM to the service level that supports the Projection Tool.
- ***Step2:*** Change the JVM option and application server configuration to enable the JVM to produce the processor time information.
- ***Step3:*** Test that your application can capture the information.
- ***Step4:*** Disable the Projection Tool.
- ***Step5:*** Analyze the data captured using the Excel workbook.

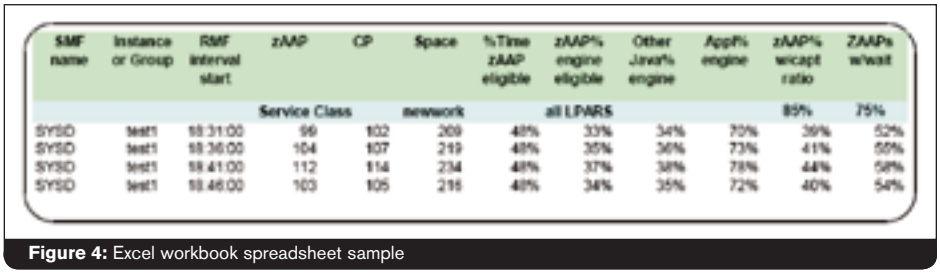
Should you need more information on how to do it, refer to http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100431

Using zAAP

This section describes setting the JVM options for leveraging zAAP, measuring zAAP utilization, and some zAAP-related performance topics.

IBM JDK1.4.1 has a set of options for zAAP processing. If you are on z/OS V1R6, the default values of the JVM options enable you to use zAAP automatically. These are the JVM options:

- ***Xifa:on*** – This is the default option on JDK 1.4.1. It enables a Java workload to run on a zAAP if available. It's only honored on z/OS V1R6.
- ***Xifa:off*** – This option disables the zAAPs.
- ***Xifa:projectn*** – This option is provided so the Project Tool can gather processor time information to help zAAP capacity planning. "n" is the requested time interval to print out the processor time information. A value of "0" indicates that the information will only be written when Java terminates. Messages are normally written after a switch has been encountered for zAAP-eligible work or when returning from that state. This option can be used on all z/OS versions for capacity planning purpose. On z/OS V1R6, SMF and RMF reports have the zAAP capacity planning information. This option disables zAAP use on z/OS V1R6 unless you specify Xifa:on or Xifa:force.
- ***Xifa:force*** – This option forces Java to continue trying to use zAAPs even if none exist. This option is only honored on z/OS V1R6 to collect RMF/SMF data to assess potential zAAP use.



SMF name	Instance or Group	RMF interval start	zAAP	CP	Space	%Time zAAP eligible	zAAP% engine eligible	Other Java% engine	AppI% engine	zAAP% wicapt ratio	ZAAPs w/wait
			Service Class		newwork		all LPARS			85%	75%
SYSD	test1	18.31.00	99	102	209	48%	33%	34%	70%	39%	52%
SYSD	test1	18.36.00	104	107	219	48%	35%	36%	73%	41%	55%
SYSD	test1	18.41.00	112	114	234	48%	37%	38%	78%	44%	58%
SYSD	test1	18.46.00	103	105	216	48%	34%	35%	72%	40%	54%

Figure 4: Excel workbook spreadsheet sample

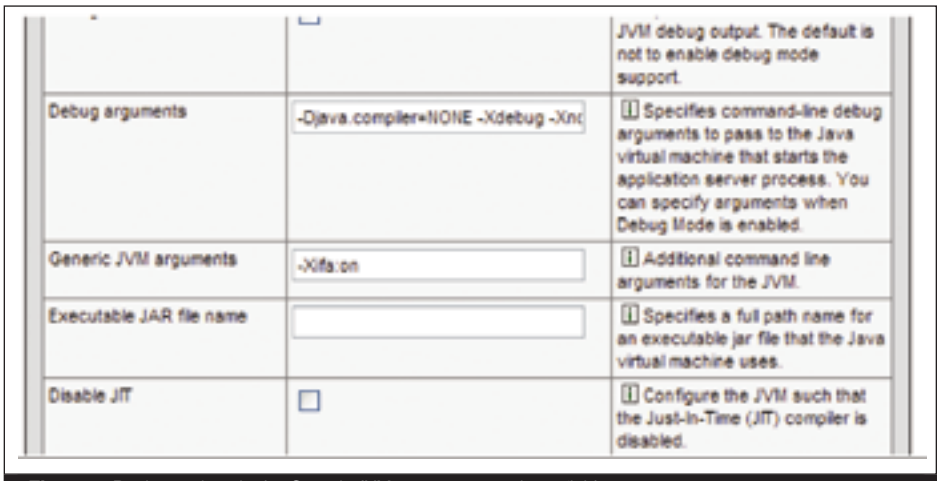


Figure 5: Putting options in the Generic JVM arguments text input field

To set these options for WAS for z/OS you have to access the administration console: Select Servers > Application Servers > Server Name > Process Definition > Control/Servant > Java Virtual Machine > Advanced JVM Settings, and put the options into the Generic JVM arguments text input field as shown in Figure 5.

Figure 5 WAS for z/OS zAAP option set-up

z/OS V1R6 provides the ability to run Java applications on zAAPs. RMF in z/OS V1R6 supports zAAPs by extending CPU Activity, Partition Data, and Workload Activity reports in Monitor I and Monitor III. The reports can be used for zAAP consumption assessment.

The following SMF record types are extended to support zAAPs:

- SMF record 70 subtype 1 (CPU activity)
- SMF record 72 subtype 2 (Workload activity)
- SMF record 79 subtype 1 and 2 (Address Space State and Resource data)

More information on how to interpret these reports can be found in zSeries Application Assist Processor (zAAP) Implementation SG24-6386-00.

I would like to clear up one thing about zAAP performance. Currently zAAP itself doesn't have any built-in mechanism to speed up Java code execution. In other word, the Java code doesn't run any faster than it does on a general CP. (On a z890, the zAAP speed is different from the general CP; the situation is a little bit more complicated.). If your system is short of CPU cycles, you will see the application response time becomes shorter when you off-load the Java workload to zAAPs. However, it doesn't mean that the Java code runs faster. It's just because some of the CPU waits are eliminated. IBM may do something for zAAP to speed up Java code execution in the future.

Once you turn on the zAAPs, the WAS for z/OS performance tuning gets more complicated. I'm going to list some of the zAAP tuning options here to help Java developers get a basic understanding of zAAP tuning to ease the conversation for system people.

The options are specified in the IEAOPTxx member of the SYS1.PARMLIB:

- IFACrossOver=YES|NO
- IFAHonorPriority=YES|NO

The IFACrossOver with a value NO means that there's no zAAP-eligible work that can run on the general CPs. Otherwise zAAP-eligible work can run on general CPs.

The IFAHonorPriority with a value NO means a general CP only selects zAAP-eligible work when the general CP isn't busy. Otherwise a general CP selects the general CP work and the zAAP-eligible work in dispatch priority order.

IFACrossOver=YES and IFAHornor Priority=YES provide the best performance for high-priority work regardless of the workload type. But it probably reduces the estimated amount of work running on the zAAPs.

IFACrossOver=YES and IFAHonor Priority=NO let all processors to run ready work, but give the zAAP a better chance of running Java work and help achieve the estimated offload of Java work.

IFACrossOver=NO provides the maximum offload of Java work to a zAAP but might cause Java work to queue to the zAAP when a general CP is available.

zWLM and LPAR management have been enhanced to handle zAAPs. More information can be found in zSeries Application Assist Processor (zAAP) Implementation SG24-6386-00.

Not all the applications running in WAS for z/OS are good candidates for exploiting zAAPs. Java-intensive applications are good candidates. Low switch-rate applications are good candidates since they minimize the software cost. The Java eligible time per switch might be a good indicator of the application's applicability for a zAAP.

Since using a zAAP is controlled at the JVM level, it implies that it's preferable to deploy applications with similar Java eligible time per switch to the same WAS for z/OS server. If the applications running in a WAS for z/OS server have long Java eligible time per switch, the JVM should be configured to leverage the zAAP. Hence it's necessary to analyze the applications for zAAP utilization using the Projection Tool described above.

However, I haven't discussed the limitations of zAAP in this article because most of them aren't of interest to application development folks. Information about zAAP limitations can be found ar IBM's zAAP resource web site.

Conclusion

zAAP is a specialized processing unit which can be used to lower the overall cost of running WAS for z/OS and other Java workload on z/OS. This article explained the basic concepts of how zAAP works and how to exploit and use them.

Resources

IBM zAAP resource Web site http://www-03.ibm.com/servers/eserver/zseries/zaap/resource.html

DataPower Acquisition

IBM announced it has acquired DataPower, a Cambridge, Mass.-based, privately-held provider of products that help improve security and speed the processing of computer transactions.

IBM acquired DataPower to help companies improve the performance, security and management of business processes built of reusable, open-standards-based software components, which operate independently from the applications and computing platforms on which they run. This increasingly popular approach, called Service Oriented Architecture (SOA), combines business operations with information technology. According to Gartner, an IT industry research firm, SOA will provide the basis for 80 percent of new software development projects by 2008. IBM plans to introduce a family of SOA appliances based on DataPower technology.

The widespread adoption of open standard technology has enabled software applications to better interoperate across business units and externally with third parties. While this has helped companies improve productivity, quickly respond to changing business needs and seize opportunities, it has also placed new demands on IT systems, which can become overburdened from processing high volumes of Web services traffic and resource-intensive malicious attacks.

"With more companies taking a modular approach to running business processes through a Services Oriented Architecture, there's a greater need for technology to deal with the commensurate increases in Web services traffic, which can overburden IT systems," said Robert LeBlanc, general manager, WebSphere, IBM Software Group. "DataPower's products address these challenges."

DataPower is a leading provider of SOA appliance products that enable integration and help provide security at the Web services message level. DataPower products available today include the XI50 Integration Device, which streamlines SOA infrastructures; the XA35 XML Accelerator, which offloads XML processing; and the XS40 XML Security Gateway,


IBM eServer zSeries

which helps provide message-level Web services security. DataPower security features complement the SOA security management capabilities of IBM's Tivoli software.

IBM has a broad portfolio of SOA solutions that enable clients to more effectively model, assemble, deploy and manage business processes. Already integrated with many IBM products, DataPower builds on IBM's existing SOA capabilities, including its WebSphere software, SOA services and consulting. As part of its SOA Business Partner initiative, IBM now has over 150 partners, which can benefit from the new deployment model the DataPower acquisition provides. Data centers can also benefit from DataPower's XML and security technology, including its product designed for IBM eServer zSeries mainframe integration.

"XML Web services have become the dominant protocol for connecting disparate systems," said Jim Ricotta, CEO, DataPower. "DataPower's specialized technology not only provides connectivity, but also message-level security, routing, integration and acceleration. DataPower products nicely complement IBM's SOA capabilities."

Mr. Ricotta will continue to manage DataPower and will undertake additional responsibility within IBM's WebSphere software area. Through this acquisition, DataPower employees will become IBM employees.

LeBlanc Comments on Acquisition

IBM's Robert LeBlanc recently answered the questions from SYS-CON.TV's Roger Strukhoff on IBM's DataPower acquisition.

Roger Strukhoff: The press release says that IBM plans to introduce a family of SOA Appliances based on DataPower technology. When will this be, and can you give us a hint of what these SOA Appliances will be?

Robert LeBlanc: Well, certainly, we're giving you a kind of a directional statement, and now that we've closed the transaction, the next level of detailed work can go on. We will make a set of announcements later this year around more specific road maps around the set of capabilities that we'll deliver but if you look at SOA and all that it entails – security, there's management, there's process management – all of the things that make up an SOA environment are all candidates to be integrated on top of an appliance, so we're going to look at all of those opportunities, to be able to extend out the set of functionality that DataPower provides today. The existing products that are available today provide a lot of value to customers, and in fact if you look at the customer list for DataPower it's pretty impressive with people like UBS, JP Morgan Chase, some of the largest financial institutions that build out pretty substantial SOA deployment and that's where we collectively work together, and where we think that there are opportunities for tighter integration between the set of capabilities that DataPower has and the rest of the IBM portfolio.

Not only web share, but for systems management, for information management and for some of the XML optimization that we're putting in the database in the future.

Strukhoff: Also, on the complimentary aspects - are you saying that with this announcement complimenting WebSphere that individual developers will have a new choice to make within the IBM family, or you're reaching out to developers who are not today prime candidates for WebSphere.
LeBlanc: It's a hard question to answer, I think both – it's clearly aimed at customers making an SOA decision, they're going to look at the breath and depth of the oppor-

tunity and of the technology to deploy, we think that it stands out as the breath of functionality we have and therefore makes WebSphere even more attractive than it already is as the leader in SOA deployment and gives more and more sets of capabilities. We think that it will bring new customers to the WebSphere platform, as well as providing value to the current customers. So, we think that it's actually both.

In answering another reporter's question if IBM will continue to work with other middleware systems, such as BEA and Oracle, LeBlanc said:

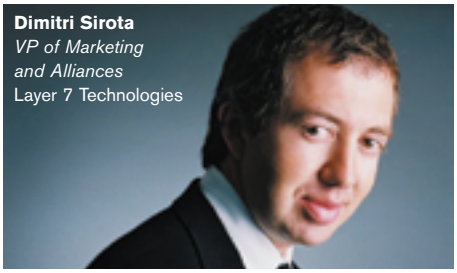
"Absolutely, because we provide value in an SOA environment, and some customers will choose their backend to be BEA or others, and not WebSphere. We provide value to customers, and that's what we're all about. So, the way that we're going to structure the organization is that we will have an SOA, a client's organization that will report directly to me, that is separate from my WebSphere organization, so that we can go after the full customer opportunity. But obviously, the optimization between DataPower and WebSphere is something that will be a huge advantage."

Level 7 Comments on Acquisition

IBM's acquisition of DataPower will no doubt cause commentary on the part of customers, partners, and competitors, as Big Blue continues its recent aggressive marketing and technology moves into the burgeoning global SOA marketplace. One of the early comments came from Layer 7, and a company executive also spoke with Web Services Journal about the deal.

"This acquisition seems to concede that the current generation performance and security solutions offered by the platform vendors is not enough for customers contemplating SOA." says Dimitri Sirota, VP of Marketing and Alliances, Layer 7 Technologies, in commenting on IBM's acquisition of DataPower. "The question for customers will now be how do they implement cross-platform, open solutions that fit their heterogeneous SOA environments. This is an area where Layer 7 has unique strength."

The company had said in an official announcement that the acquisition represented "a major milestone in the SOA sector, and demonstrates growth in the market." The statement added that the acquisition "presents many opportunities to companies


Dimitri Sirota
VP of Marketing and Alliances
Layer 7 Technologies

like Layer 7 that provide standards-based, cross-platform security and governance solutions for XML, web services, and SOA."

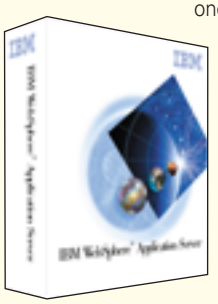
Questioned about Layer 7's existing relationship with IBM's Tivoli division, Sirota noted that "Tivoli has been a good, but not exclusive, partner for Layer 7. We have multiple joint customers and several joint opportunities we will continue to support with Tivoli. We of course also have opportunities generated through other relationships."

"While Datapower and Layer 7 compete on high performance XML appliances, Layer 7 also offers software based gateways, XML VPNs and SOA compliance products not offered by Datapower. We anticipate continuing to work with Tivoli in several of these areas."

WebSphere Portal Dominates

IBM has announced that WebSphere Portal has been named the industry's number one enterprise portal product by a top market analyst firm. According to IDC, IBM has the highest percentage of world wide portal market share for 2004 based on software license and maintenance revenue, which is more than 10 percent higher than the closest competitor

Organizations today are facing an overload of information on multiple yet disparate systems, making it increasingly difficult to identify, access, and disseminate relevant information. Through a security-rich and scalable infrastructure, WebSphere Portal helps businesses organize, combine, and make information and systems easier to use. In addition, WebSphere Portal is a key part of the SOA lifecycle for deploying people based services.



"WebSphere Portal, an integral part of the IBM Workplace family of products and solutions, continues to lead the portal industry with our customer base growing by double digits this year says Ken Bisconti, vice president of IBM Workplace, Portal and Collaboration Products. Support for services oriented architecture (SOA), a high level of security and a clear and reliable product roadmap have been the key drivers in this success."

The following represent a sample of the thousands of customers of all sizes and industries that are using the powerful capabilities of WebSphere Portal technology to address complex collaboration issues.

Ball Memorial Hospital, the flagship hospital for Cardinal Health System, selected IBM WebSphere Portal and Bowstreet technology to develop a physicians' dashboard that aggregates data and functionality from disparate systems. The physicians' dashboard provides access to critical data, such as patient, scheduling, and pharmaceutical information, from a single source.

"We're very pleased with the results we have achieved. The time our physicians save by using the dashboard allows each of them to schedule one additional patient per day. This translates to a potential increase in revenue of approximately $60,000 per month or $720,000 per year," explains Christina Fogle, Manager of eSystem Support, Cardinal Health System.

The AO Foundation, a nonprofit organization dedicated to research and education in the area of trauma and corrective surgery, turned to IBM and IBM Business Partner WP Experts to create a portal that would let surgeons and faculty members communicate and access everything from surgical procedures and research data, to continuing education requirements.

"As a nonprofit organization we face an interesting challenge, we have a lot of information that needs to reach a wide variety of audiences world wide yet we have a limited amount of time and resources to make it happen," says Michael Redies, Head of Knowledge Services department for the AO Foundation. "With WebSphere Portal we are able to put the right information in the hands of our surgeons and faculty members easily and efficiently."

Citrix and VMWare Team with IBM

Citrix and VMWare are teaming with IBM to launch the first offering of the new IBM Virtualized Hosted Client Infrastructure that delivers full desktop functionality to any client from highly reliable and secure xSeries and BladeCenter servers.

IBM is announcing a new pre-integrated hosted client solution that is based on VMware virtual infrastructure software and leverages Citrix client access technologies to host multiple users of fully functional desktop environments on IBM BladeCenter, while increasing efficiencies such as utilization of computing resources and quicker deployment of new users. With the new virtualized solution, customers could see total cost savings of up to 60 percent and return on investment in as short as six months.

IBM's Virtualized Hosted Client Infrastructure helps overcome a key obstacle to adoption of hosted client architectures: resistance from the desktop user. Users will be able to enjoy all of the benefits and personal control of a stand alone desktop -- functionality including print capabilities, USB drive support, dual monitors and audio - while reducing many of the hassles of a "fat client," including limiting susceptibility to theft and viruses, extended downtime during a hard drive failure, or having to rebuild their preferences and settings after each client "refresh," according to IBM.

This companies says this new solution is ideal for all types of knowledge workers with constant computing requirements, and also for remote employees, branch office environments like automobile dealerships or retail stores, customer service call centers and software developers.

"Existing hosted client solutions gain some efficiencies by consolidating clients, but they fall short of delivering the full potential for client experience and IT efficiency," said Doug Balog, vice president and business line executive, BladeCenter, IBM Systems and Technology Group. "IBM's Virtualized Hosted Client Infrastructure with VMware virtual infrastructure software can provide high utilization, performance and reliability, allowing customers to reduce cost and complexity, while providing the level of desktop functionality and quality of support that end users have come to expect."

The first solution to be rolled out using the Virtualized Hosted Client Infrastructure is based on IBM BladeCenter, VMware's virtual infrastructureproduct line -- which has been proven in production at more than 10,000 customers worldwide -- and Citrix Presentation Server, the industry standard for providing access to client/server and desktop-based applications using application virtualization. VMware and Citrix are two key founding members of Blade.org, a collaborative organization and developer community focused on accelerating the expansion of blade solutions and the BladeCenter ecosystem.

BladeCenter

"VMware virtual infrastructure makes it possible to have one common infrastructure for both desktops and servers and has enormous benefits for customers implementing client hosting. It delivers availability, responsiveness, flexibility, scalability and cost benefits," said Brian Byun, vice president of strategic alliances at VMware. "

VMware virtual infrastructure boosts server utilization rates to up to 80 percent, driving more efficiencies than a solution that is merely dedicated hardware to support desktop features. We're excited to be working with IBM on continuing to bring these benefits to customers with IBM's new Virtualized Hosted Client Infrastructure offering."

"The new IBM Virtualized Hosted Client integrated solution including Citrix keeps security and management under IT control while enabling a more productive and personal work environment for end users," said David Jones, corporate vice president, business development and corporate affairs, for Citrix. "Citrix has added a new capability to Citrix Presentation Server that works with IBM BladeCenter and VMware virtualization technologies to enable dynamic provisioning of hosted clients running full Windows XP Desktop images in a secure data center. The resulting solution is used to provide the rich experience that PC users have come to expect and is now available from thin clients or legacy PC's."

Citrix Presentation Server allows customers to deliver full desktop capability on any client from anywhere with high levels of security and little to no downtime. In addition, the images of the application appear on a user's local, remote or mobile computer screen allowing them to work seamlessly regardless of their location, connection or device.

The IBM Virtualized Hosted Client Infrastructure brings new important benefits to the information technology (IT) department. In addition to delivering security and control, IBM says, as it uses VMware virtual infrastructure software to help improve IT department efficiencies by optimizing resource utilization, as well as increase end-user performance by dynamicaly provisioning resources to meet the performance needs of each virtual hosted client machine. This enables companies to support more users with fewer resources, while offering the best possible desktop experience from virtually anywhere and on virtually any device.

The IBM Virtualized Hosted Client Infrastructure will be delivered by IBM Global Services (IGS). IGS is currently enrolling customers for pilot implementations, and an IGS service offering is planned to be available in first quarter of 2006. In addition, availability is planned through select IBM Business Partners.

Subscribe Today!

– INCLUDES –
FREE DIGITAL EDITION!
(WITH PAID SUBSCRIPTION)
GET YOUR ACCESS CODE INSTANTLY!

*The major infosecurity issues of the day... **identity theft, cyber-terrorism, encryption, perimeter defense, and more** come to the forefront in* **ISSJ** *the storage and security magazine targeted at IT professionals, managers, and decision makers*

Editorial Mission

Greater Collaboration and Efficiency Through Education

✓ ***ISSJ*'s editorial mission is to showcase proven solutions** that will guide, motivate, and inspire senior IT and business management leaders in the planning, development, deployment, and management of successful enterprise-wide security and storage solutions.

✓ ***ISSJ*** brings together all key elements of data storage and protection, and presents compelling insight into the benefits, efficiencies, and effectiveness gained by focusing on these two critical areas of IT simultaneously.

✓ ***ISSJ*** is an objective, critical source of information that helps storage and security managers make informed management decisions about what is working today, and what they need to plan for tomorrow, and is the only publication that focuses exclusively on the needs of IT professionals who are driving the enterprise storage architecture/infrastructure while pursuing and incorporating the latest security technologies.

✓ ***ISSJ*** achieves our mission by delivering in-depth features, practical "how-to" solutions, authoritative commentary, hard-hitting product reviews, comprehensive real-world case studies, and successful business models, written by and for professional IT storage and security practitioners.

SAVE 50% OFF!

(REGULAR NEWSSTAND PRICE)

Only $39.99 ONE YEAR 12 ISSUES

www.ISSJournal.com

or

1-888-303-5282

SYS-CON MEDIA

The World's Leading i-Technology Publisher

WAS CE Takes Open-Source Approach

IBM has announced a new open source-based software and support initiative, called WebSphere Application Server Community Edition (WAS CE). To be available later this year, the software is designed to provide mid-sized businesses, departments in large enterprises and business partners with easy access to open source-based technologies with no upfront costs. WAS CE is customizable application server software that features a small footprint, making it easier to download and manage.

Based on core open source technology from the J2EE-certified Apache Geronimo application server, WAS CE and related subscription support provide a flexible and affordable alternative to traditional commercial software offerings and can help customers begin to deploy a Service Oriented Architecture (SOA).

WAS CE includes technology from Gluecode Software, which IBM acquired in May. With this new offering WAS CE, IBM broadens the range of customer choice for infrastructure software, while providing a transition to more feature-rich versions of WebSphere, the industry's leading standards-based middleware.

WAS CE helps Java developers reduce the complexity of application development by pre-integrating the most common services for building applications. It offers customers the choice of a blended commercial offering that provides the innovation of open source as well as greater stability and higher levels of support that comes from IBM. The new software supports Apache Tomcat, an industry standard Web server also from the Apache Software Foundation and WAS CE will also integrate IBM's Cloudscape database, which is based on the open source Apache Derby Project.

"Customers are looking for support for open source technologies from a trusted IT vendor," said Robert LeBlanc, general manager, WebSphere, IBM Software Group. "WAS CE allows customers and partners to tap the innovation of open source technology – backed by industry leading IBM support services – to quickly develop and deploy applications based on open source technologies."

There is no cost for customers to download and use WAS CE. For those customers that desire technical support, IBM is introducing a full line of WAS CE support services starting at $900 per server for an annual subscription. Available support includes a new developer-to-developer support service that allows customers' development teams to obtain support directly from IBM developers to resolve certain business problems using WAS CE.

IBM is an active contributor to the Apache Geronimo open source project and will continue to contribute and collaborate with the growing community of developers that advance this project and innovate on top of it. IBM is a long time supporter of the Apache Software Foundation and has donated software code to Apache Geronimo. Today IBM is also announcing it will extend technical support for Apache Geronimo to customers worldwide.

Flashback: OpenSource Ready For e-Business

January 31, 2001
2001 was remembered as the year business customers embraced OpenSource, declared IBM President and Chief Operating Officer Samuel J. Palmisano in the keynote address at the LinuxWorld Conference at the Javits Center in New York City on Wednesday, Jan. 31,2001. Palmisano said that the Linux operating system is rapidly moving from the world of academia and scientific computing to commercial businesses.

SOURCE: WWW.IBM.COM

Aperi Addresses Open-Source Storage Management

Brocade, Cisco, Computer Associates, Engenio, Fujitsu, IBM, McDATA, and Network Appliance announced their intent to form, a new open source community, initially working under the project name Aperi, to give customers more choices for deploying open-standards-based storage infrastructure software. The organization plans to develop a common storage software management platform that will give customers greater flexibility in the way they manage their storage environments.

Aperi – derived from the Latin word for "to open" -- will take an open approach to build a common platform for managing all brands of storage systems, with community members contributing code and taking advantage of a common platform for building storage software applications.

Previous storage industry group initiatives have focused on establishing standards but not collaborating to develop an open source-based platform to manage storage devices.

Members of Aperi, which will be managed by an independent, non-profit, multi-vendor organization, will work together to evolve the platform, which will be available free-of-charge. The consortium soon will announce details about the organization, including the multi-vendor board of directors.

At first, IBM plans to donate part of its storage infrastructure management technology to the open source community. Other members will have the option to also donate a portion of their intellectual property, so that collectively the group can advance the platform and encourage developers to write storage management software based on the platform.

Providing more flexibility for storage management provides a range of benefits to customers, who face storage complexity and staff shortages and lack a standard way to manage information through its lifecycle. By standardizing on a common software infrastructure platform, customers will be able to choose from a greater range of storage management products. In addition, doing so could help eliminate the need to "rip and replace" storage management software when purchasing new hardware or applications, and reduce training requirements for storage operations teams.

Similarly, the community will help customers by enabling storage vendors to efficiently develop a management platform within an open source-based consortium framework, while focusing on developing new capabilities that enhance their own storage products.

Aperi will build upon existing open storage standards, including the Storage Networking Industry Association's (SNIA) Storage Management Initiative Specification (SMI-S), which develops and standardizes interoperable storage management technologies for storage hardware interfaces.

IBM Says Aperi "Door is Open"

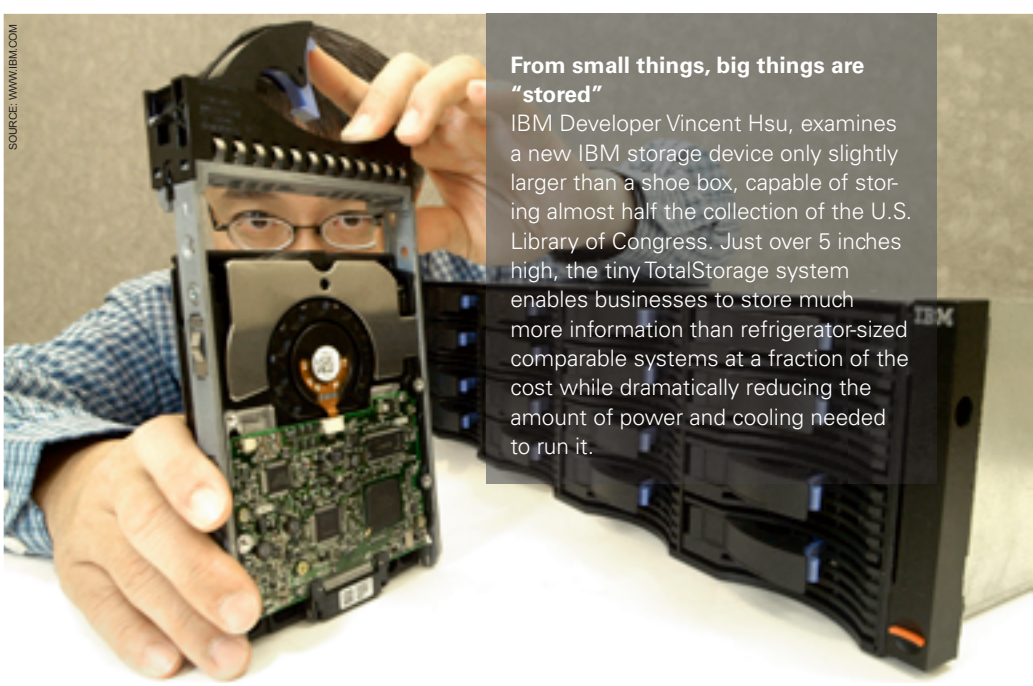
IBM led a conference call regarding the formation of a new open source storage platform initiative, to be known as Aperi. Founding members of this initiative also include CA, Brocade, NetApp, and others. Notably absent during the call were EMC, HP, and Sun Microsystems, although it was announced that Sun will be joining the initiative.

The new initiative, which apparently grew out of discussions within member companies of the Storage Network Industry Association (SNIA), but which will not be directly associated with the SNIA, still lacks a board of directors or a timeline. Company representatives said the first code will be available "in 2006," but could not be more specific.

Regarding the notable absence of EMC and HP, company representatives during the call said, "The Door is open, the door is open to any company that wants to join us."

There was also a lack of specificity about how this project will encompass both Storage Area Networks (SANs) and Network-Attached Storage (NAS), the two competing methods for deploying enterprise storage today. The SAN market, led by EMC, remains much larger, but NAS has a faster growth rate with solutions by market leader NetApp setting the standard. The major computing platform vendors--IBM, HP, and Sun--each try to be leaders in each category, and a new company, Pillar Data Systems, is promising a universal management solution.

Yet the worlds of SAN and NAS remain distinct, with specific management problems associated with each. Whether Aperi will offer something that resides above (or below) traditional storage management software, or does indeed produce some sort of universal storage management solution is a story that will play out as the Aperi board is formed, code starts to be written, and feedback is received from the vendor and user communities.



From small things, big things are "stored"
IBM Developer Vincent Hsu, examines a new IBM storage device only slightly larger than a shoe box, capable of storing almost half the collection of the U.S. Library of Congress. Just over 5 inches high, the tiny TotalStorage system enables businesses to store much more information than refrigerator-sized comparable systems at a fraction of the cost while dramatically reducing the amount of power and cooling needed to run it.

The wrong people are in the wrong jobs

Modern Management

BY DAVID LANGE

Many professionals believe the goals of management aren't related to achievement, but on perpetuating one's own position. The management process is a bewildering experience for freshman employees and a cause of anguish to experienced professionals. Common sense says a manager should be a well-balanced individual, unselfish, able to accept information, and adjudicate situations. A manager should be selected based on a proven track record that demonstrates responsibility toward everyone, common sense, lack of favoritism, attentiveness to detail, and a desire to achieve business objectives with a minimum of waste.

Some 20 years of shared experiences has proven these naïve notions to be counterintuitive and completely opposite to reality. Burnout among IT personnel is increasingly evident. Look at the number of publications reporting it. Evidently IT personnel have a cynical view of their profession and management. One is more likely to be confronted by a manager who lacks ability in technical and human terms than not.

Why has shoddy management become the norm rather than the exception?

A possible answer might be how managers get their jobs. Current business practice puts more value on degrees than experience. Business school advocates claim their curriculums can make a person management material regardless of the field. The fallacy of this logic is obvious to people who possess career experience over the ponderings of the learned and inexperienced. How can an executive manage people without a rudimentary knowledge of the task at hand? More likely you'll find yourself explaining something that's taken years of effort to grasp while the manager seems extremely confused and nursing a bruised ego. Not that this writer is against gaining a degree. But possessing a degree implies one can learn, not the right to manage. Unfortunately, business leaders have taken reality out of the equation and deem experience without merit. Industry reports indicate most IT projects wind up abysmal failures, though the problem is erroneously addressed by across-the-board cost cutting. Most professionals comparing experience say that management doesn't grasp reality or is swamped by ever more complex procedures that take on a life of their own. The problem is simply over-complication created by management ignorant of the realities of their own business.

Managers aren't trained properly by their companies, nor do they work their way up the ranks as common sense dictates. A manager should be a subject matter expert trained to manage versus simply having a management degree. We have created people trained to be self-centered, close-minded, and lacking any association with the people they manage not to mention the actual business at hand.

On the human side of the equation a common complaint centers on the information barriers between management and personnel. Professional consultants often complain about management's inability to listen, process information, and take reasonable action. The answer isn't because the manager is less intelligent, but because the goals of management are different than the staff's. By eliminating these differences you can actually form a team, though most executives don't see that.

The net effect of employing an illogical basis for choosing and promoting managers wastes resources in most IT departments. Directives to tighten up and limit IT costs have resulted in more administrative processes than before, though once again the processes are flawed by a lack of actual development or administration experience. The typical response to the failure of the first process is to just reinvent it in another form, albeit more complicated and wasteful than it was. The net result is that people become confused by directives and processes that have little to do with the daily business. When your staff seem exasperated there might be a reason, but most managers don't realized that they are the problem or seem concerned in the least.

Not that this situation hasn't always existed. The problem is practically biblical. Human nature is the root of the problem and our society has yet to address it. The answer isn't easy, but – be assured – it might be found by applying a few basic principles such as:

1. Minimize the myriad levels of management to a near-flat organization.
2. Allow groups to manage themselves with review by their peers.
3. Management promotion should be based on achievement.
4. Process design should originate from the bottom and not the top.
5. Managers should be reviewed by consulting those they manage.
6. Allow employees to be inventive and foster discussion.
7. Don't use ruses to dismiss employee frustration. Having a group lunch does nothing to resolve business problems.
8. Act constructively to resolve situation and earn respect.
9. Simplify all procedures and openly document them.
10. Treat your staff with respect.

ABOUT THE AUTHOR

David Lange is an independent consultant with some 15 years experience in development, architecture, and administration with regard to Unix-based systems. David has survived projects dealing with scientific research, manufacturing, banking, and trading systems, but would rather spend his time pursuing operatic study, sports cars, and travel.
dlange_scg@hotmail.com

YOUR TESTING SUCKS.

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

LISA from iTKO.

Because everyone should own quality.

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

We understand.

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

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

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

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

LISA is different.

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

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

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

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

IBM WEBSPHERE PRESENTS

YOU VS REWRITE, REVISE RE-EVERYTHING

INFLEXIBILITY: MEET SOA

PLUS: TANGIBLE BUSINESS BENEFITS ★ BEST PRACTICES FOR BEST RESULTS

OVER 10 YEARS OF WORLD-CLASS INTEGRATION EXPERTISE

FEATURING:

A FASTER, EASIER WAY TO IMPLEMENT TRUE SOA	SUPPORT FOR OVER 80 O.S. CONFIGURATIONS	UNPARALLELED INDUSTRY KNOWLEDGE & PROCESS SKILL

IBM MIDDLEWARE. POWERFUL. PROVEN.

FIGHT BACK AT WWW.IBM.COM/MIDDLEWARE/SOA. THIS IS A RIP-AND-REPLACE-FREE EVENT.